TEXT MINING BIOMEDICAL LITERATURE FOR GENOMIC KNOWLEDGE DISCOVERY

A Thesis
Presented to
The Academic Faculty

by

Ying Liu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Computer Science

Georgia Institute of Technology
August 2005

TEXT MINING BIOMEDICAL LITERATURE FOR GENOMIC KNOWLEDGE DISCOVERY

Approved by:

Shamkant B. Navathe, Chairman
College of Computing
Georgia Institute of Technology

Ray Dingledine
Department of Pharmacology
Emory University

Brian J. Ciliax
Department of Neurology
Emory University

Edward Omiecinski
College of Computing
Georgia Institute of Technology

Venu Dasigi
Department of Computer Science
Southern Polytechnic and State University

Ashwin Ram
College of Computing
Georgia Institute of Technology

Date Approved June 28th, 2005

This dissertation is dedicated to my parents.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the help and support of many people.

Firstly I would like to thank my advisor, Prof. Shamkant B. Navathe, for his excellent supervision, his knowledge, his belief and interest in the work and encouragement and motivation throughout. I would also like to thank Dr. Brian J. Ciliax for being my good friend, my mentor and someone to follow.

I am very grateful to Dr. Ray Dingledine for his hours of patient and detailed proofreading, and many conversations. My thanks also go to everyone who has provided support or advice in one way or another, including Dr. Venu Dasigi, Dr. Ashwin Ram, and Dr. Edward Omincinski. The researchers in Center of Disease of Control and Preventation, such as Dr. Muin Korey, Dr. Marta Gwinn, Mr. Bruce Lin, provided us the training sets and testing sets in Chapter 6. I also want to thank other graduate students in the lab, including Nalini Polavarapu, Saurav Sahay, Abhishek Dabral, Ramprasad Ramnarayanan for their support.

And finally I'd like to thank my family for their love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The last decade has been marked by unprecedented growth in both the production of biomedical data and the amount of published literature discussing it. Advances in computational and biological methods have remarkably changed the scale of biomedical research. Large-scale experimental methods, such as microarray, produce large quantities of *data.* When processed, the data can provide actual *information* about gene expression patterns, for instance, which genes are expressed in various tissues, and which ones are over/under expressed at the onset of a disease or during a specific phase of the cell development. Still, the ultimate goal of conducting large-scale biology is to translate this large amount of *information* into *knowledge* of the complex biological processes governing the human body and to utilize this knowledge to advance healthcare and medicine.

Almost every known or postulated piece of information pertaining to genes, proteins, and their role in biological processes is reported somewhere in the vast amount of published biomedical literature. However, the advancement of genome sequencing techniques is accompanied by an overwhelming increase in the literature discussing the discovered genes. This combined abundance of genes and literature produces a major bottleneck for interpreting and planning genome-wide experiments. Thus, we believe the ability to rapidly survey and analyze this literature and extract pertinent information constitutes a necessary step toward both the design and the interpretation of any large-scale experiment. Moreover, automated literature mining offers a yet untapped opportunity to integrate many fragments of information gathered by researchers from

multiple fields of expertise into a complete picture exposing the interrelated roles of various genes, proteins, and chemical reactions in cells and organisms.

In the present work it is our thesis that functional keywords in biomedical literature, particularly Medline, represent very valuable information and can be used to discover "new" knowledge about genes. To test this thesis and to validate our claim we conduct following studies:

1. We test sets of genes (26 genes compiled as a group by experts and 44 genes from the literature) to "discover" common functional keywords among them and use these keywords to cluster them into groups.

We cluster genes that share functionally relevant keywords in MEDLINE abstracts. The keywords that describe the most prominent common functions of the genes within each group are extracted to assist hypothesis generation. Words with no indexing values are filtered by stop list. Functionally less relevant words are filtered out based on the threshold of the weighting schemes. The resulting weights of the keywords are used as feature vectors for clustering algorithms. We develop an algorithm called BEA-PARTITION based on Bond Energy Algorithm (BEA). In order to explore whether this algorithm could be useful for clustering genes derived from microarray experiments, we compare the performance of BEA-PARTITION, hierarchical clustering algorithm, self-organizing map, and the *k*-means algorithm for clustering functionally-related genes. Genes are assigned into functionally relevant clusters based on shared keywords that suggest the principal biological functions of each cluster.

2. We show that it is possible to link genes to diseases by an expert human interpretation of the functional keywords for the genes- none of these diseases are as yet mentioned in public databases.

For the gene osteopontin, the keyword list generated by our keyword extraction methodology shows that our methodology is able to identify keywords associated with newly discovered functions of the gene in hypertension , tumor metastasis, or in autoimmune demyelinating disease. While this information is not represented in other resources, such as the Gene Ontology (GO) Consortium, SwissProt, GenBank, and GeneCards till today.

3. By clustering genes based on commonality of functional keywords it is possible to group genes into meaningful clusters that reveal more information about their functions, link to diseases and roles in metabolism pathways.

Keywords shared among genes within each cluster are ranked according to a metric based on both the degree of significance (the sum of weight for each keyword) and the breadth of distribution (the sum of the number of genes within the cluster for which the keyword has a z-score greater than a selected threshold). The respective keyword lists appeared to be highly informative about the general function of the original, pre-selected clusters. For the 44 yeast microarray gene set, the shared keyword list reveals the possible relationship between four genes (Exg1, Cwp1, Mnn1, and Och1) and polysaccharide metabolism pathway.

4. Using extracted functional keywords, we are able to demonstrate that for yeast genes, we can make a better functional grouping of genes in comparison to available public microarray and phylogenetic databases such as the one in Munich Information

Center for Protein Sequences Yeast Genome Database (MYGD) (*http://mips.gsf.de/genre/proj/yeast/index.jsp*).

Analysis of the yeast genome provides many challenges to existing computational techniques. Data is now available on a genome-wide scale from sources such as the results of microarray experiments, and sequence characteristics, accompanied by a number of publications discussing gene-related discoveries. All these data sources provide researchers valuable data sources for gene function prediction. We present a comparative study of yeast gene function prediction using different data sources, namely microarray data, phylogenetic data, and literature text data. The results show that text data outperforms microarray data and phylogenetic data in gene function classification. There is no significant difference between the results derived from microarray data and phylogenetic data.

5. We show an application of our approach to literature classification. Using functional keywords as features, we are able to extract epidemiological abstracts automatically from Medline with higher sensitivity and accuracy than a human expert.

PubMed (Medline) is a large repository of publicly available scientific literature. Searching PubMed database on a specific topic presents a big challenge to the users. Typically, even after formulating complex requests against PubMed, the Positive Predictive Value (PPV) (also called precision) of the search is at most 5-10%. The researcher typically ends up scanning the retrieved records for relevance, which is very time consuming and error- prone. We first analyze epidemiology relevant literature of interest to CDC and define a set of useful keywords that rank above a certain threshold. We then apply the Support Vector Machines (SVM) approach for automatic retrieval of

PubMed articles related to Human genome epidemiological research at CDC using these highly informative keywords as the features in the vectors. We discuss various investigations into biomedical literature categorization and analyze the effect of various issues related to the choice of keywords, training sets, and parameters for the SVM technique.

CHAPTER 1

MOTIVATION AND BACKGROUND

1.1 Motivation and contribution of this thesis

The last decade has been marked by unprecedented growth in both the production of biomedical data and the amount of published literature discussing it. Advances in computational and biological methods have remarkably changed the scale of biomedical research. Complete genomes can now be sequenced within months and even weeks (Myers, 1999; Venter *et al.*, 2001), computational methods expedite the identification of tens of thousands of genes within the sequenced DNA (Burge and Karlin, 1998; Bafna and Huson, 2000; Korf *et al.*, 2001), and automated tools are developed for analyzing properties of genes and proteins (Altschul *et al.*, 1997; Horton and Nakai, 1997; Sonnhammer *et al.*, 1998; Emanuelsson *et al.*, 2000; Jaakkola *et al.*, 2000). Modern techniques such as DNA microarrays allow simultaneous measurements for all genes/proteins expressed in a living system (Schena *et al.*, 1995; Lockhart *et al.*, 1996; DeRisi *et al.*, 1997; Spellman *et al.*, 1998). These large-scale experimental methods produce large quantities of *data.* When processed, the data can provide actual *information* about gene expression patterns, for instance, which genes are expressed in various tissues, and which ones are over/under expressed at the onset of a disease or during a specific phase of the cell development. Still, the ultimate goal of conducting large-scale biology (Bassett *et al.*, 1999) is to translate these large amounts of *information* into *knowledge* of the complex biological processes governing the human body and to utilize this knowledge to advance healthcare and medicine (Shatkay and Feldman, 2003).

Almost every known or postulated piece of information pertaining to genes, proteins, and their role in biological processes is reported somewhere in the vast amount of published biomedical literature. However, the advancement of genome sequencing techniques is accompanied by an overwhelming increase in the literature discussing the discovered genes. This combined abundance of genes and literature produces a major bottleneck for interpreting and planning genome-wide experiments. Thus, we believe the ability to rapidly survey this literature constitutes a necessary step toward both the design and the interpretation of any large-scale experiment. Moreover, automated literature mining offers a yet untapped opportunity to integrate many fragments of information gathered by researchers from multiple fields of expertise into a complete picture exposing the interrelated roles of various genes, proteins, and chemical reactions in cells and organisms.

In the present work it is our thesis that functional keywords in biomedical literature, particularly Medline, represent very valuable information and can be used to discover "new" knowledge about genes. To test this hypothesis and to validate our claim we conduct following studies:

1. We test sets of genes (26 genes compiled as a group by experts and 44 genes from the literature) to "discover" common functional keywords among them and use these keywords to cluster them into groups.

We cluster genes that share functionally relevant keywords in MEDLINE abstracts. The keywords that describe the most prominent common functions of the genes within each group are extracted to assist hypothesis generation. Words with no indexing values are filtered by stop list. Functionally less relevant words are filtered out based on

the threshold of the weighting schemes. The resulting weights of the keywords are used as feature vectors for clustering algorithms. We develop an algorithm called BEA-PARTITION based on Bond Energy Algorithm (BEA). In order to explore whether this algorithm could be useful for clustering genes derived from microarray experiments, we compare the performance of BEA-PARTITION, hierarchical clustering algorithm, self-organizing map, and the *k*-means algorithm for clustering functionally-related genes. Genes are assigned into functionally relevant clusters based on shared keywords that suggest the principal biological functions of each cluster.

2. We show that it is possible to link genes to diseases by an expert human interpretation of the functional keywords for the genes- none of these diseases are as yet mentioned in public databases.

For the gene osteopontin, the keyword list generated by our keyword extraction methodology shows that our methodology is able to identify keywords associated with newly discovered functions of the gene in hypertension , tumor metastasis, or in autoimmune demyelinating disease. While this information is not represented in other resources, such as the Gene Ontology (GO) Consortium, SwissProt, GenBank, and GeneCards till today.

3. By clustering genes based on commonality of functional keywords it is possible to group genes into meaningful clusters that reveal more information about their functions, link to diseases and roles in metabolism pathways.

Keywords shared among genes within each cluster are ranked according to a metric based on both the degree of significance (the sum of weight for each keyword) and the breadth of distribution (the sum of the number of genes within the cluster for which

the keyword has a z-score greater than a selected threshold). The respective keyword lists appeared to be highly informative about the general function of the original, pre-selected clusters. For the 44 yeast microarray gene set, the shared keyword list reveals the possible relationship between four genes (Exg1, Cwp1, Mnn1, and Och1) and polysaccharide metabolism pathway.

4. Using extracted functional keywords, we are able to demonstrate that for yeast genes, we can make a better functional grouping of genes in comparison to available public microarray and phylogenetic databases such as the one in Munich Information Center for Protein Sequences Yeast Genome Database (MYGD) (*http://mips.gsf.de/genre/proj/yeast/index.jsp*).

Analysis of the yeast genome provides many challenges to existing computational techniques. Data is now available on a genome-wide scale from sources such as the results of microarray experiments, and sequence characteristics, accompanied by a number of publications discussing gene-related discoveries. All these data sources provide researchers valuable data sources for gene function prediction. We present a comparative study of yeast gene function prediction using different data sources, namely microarray data, phylogenetic data, and literature text data. The results show that text data outperforms microarray data and phylogenetic data in gene function classification. There is no significant difference between the results derived from microarray data and phylogenetic data.

5. We show an application of our approach to literature classification. Using functional keywords as features, we are able to extract epidemiological abstracts automatically from Medline with higher sensitivity and accuracy than a human expert.

PubMed (Medline) is a large repository of publicly available scientific literature. Searching PubMed database on a specific topic presents a big challenge to the users. Typically, even after formulating complex requests against PubMed, the Positive Predictive Value (PPV) (also called precision) of the search is at most 5-10%. The researcher typically ends up scanning the retrieved records for relevance, which is very time consuming and error- prone. We first analyze epidemiology relevant literature of interest to CDC and define a set of useful keywords that rank above a certain threshold. We then apply the Support Vector Machines (SVM) approach for automatic retrieval of PubMed articles related to Human genome epidemiological research at CDC using these highly informative keywords as the features in the vectors. We discuss various investigations into biomedical literature categorization and analyze the effect of various issues related to the choice of keywords, training sets, and parameters for the SVM technique.

Therefore, the research in this thesis concentrates on the area of genomic knowledge discovery. When a genome is sequenced, and we have the predicted locations of the genes within the genome, the next stage is to work out the possible functions of these genes. In this thesis we test the hypothesis that extraction of functional keywords from Medline as a representative of the entire biomedical literature, will provide very valuable information to discover "new" knowledge about genes. . The outcomes of this thesis are in three distinct areas. For each area, we list the important issues that we considered and problems we addressed

1. Text mining biomedical literature to discover gene-to-gene relationships.

This is a challenging environment for computer science, where specific challenges include:

• Many new techniques in biology, such as microarray, are providing data on a genome wide scale. This data is noisy.

• Available clustering techniques typically provide little or no direct information about the nature of the functional links among genes within the derived clusters.

• By using text mining , our goal is to discover the functional link among genes.

2. Yeast (*Saccharomyces cerevisiae*) gene function prediction from different data sources.

   • The immense volume of data resulting from genomic sequencing and DNA microarray experiments, accompanied by the number of publications discussing gene-related discoveries provide researchers valuable data sources for gene function prediction.

   • Different data sources can be used to learn gene function.

   • There is no empirical comparison to determine the relative effectiveness or usefulness of different types of data in terms of gene function classification.

   • We wish to perform a comparative study of yeast gene function classification using different data sources.

3. Automated Classification of Biomedical literature .

   • PubMed (Medline) is a large repository of publicly available scientific literature. Currently, new data is being added to it at the rate of over 1500 abstracts per week. The ability to efficiently review the available literature is essential for rapid progress of research in scientific community.

• The traditional literature database search involves the use of simple Boolean queries, formulated using certain frequently used functionally important keywords the researcher is familiar with, followed by manual scanning of the retrieved records for relevance, which is time consuming, incomplete and error prone.

• There is a pressing need for the development of automated literature mining techniques that can help the researchers to effectively harvest the heap of the knowledge available in the scientific literature.

• Design a system based on support vector machine to categorize the biomedical articles automatically.

1.2 An overview of functional genomics (Clare, 2003)

The determination of gene function from genomic information is what is known as functional genomics. The central dogma of biology is that DNA is transcribed into RNA and RNA is translated into proteins. Figure 1-1 shows the relationship between the three. When we speak of gene function we usually mean the function of the products of genes after transcription and translation, which are proteins.

Proteins

Proteins are the molecules which do almost all the work in the cell. They are extremely important molecules, involved in everything from immunity to muscle structure, transportation, hormones, metabolism, respiration, repair, and control of genes. Understanding the roles of proteins is the key to understanding how the whole cell operates.

Proteins are polymers consisting of chains of amino acids. There are 20 different amino acids, so proteins can be represented by strings of characters for computational

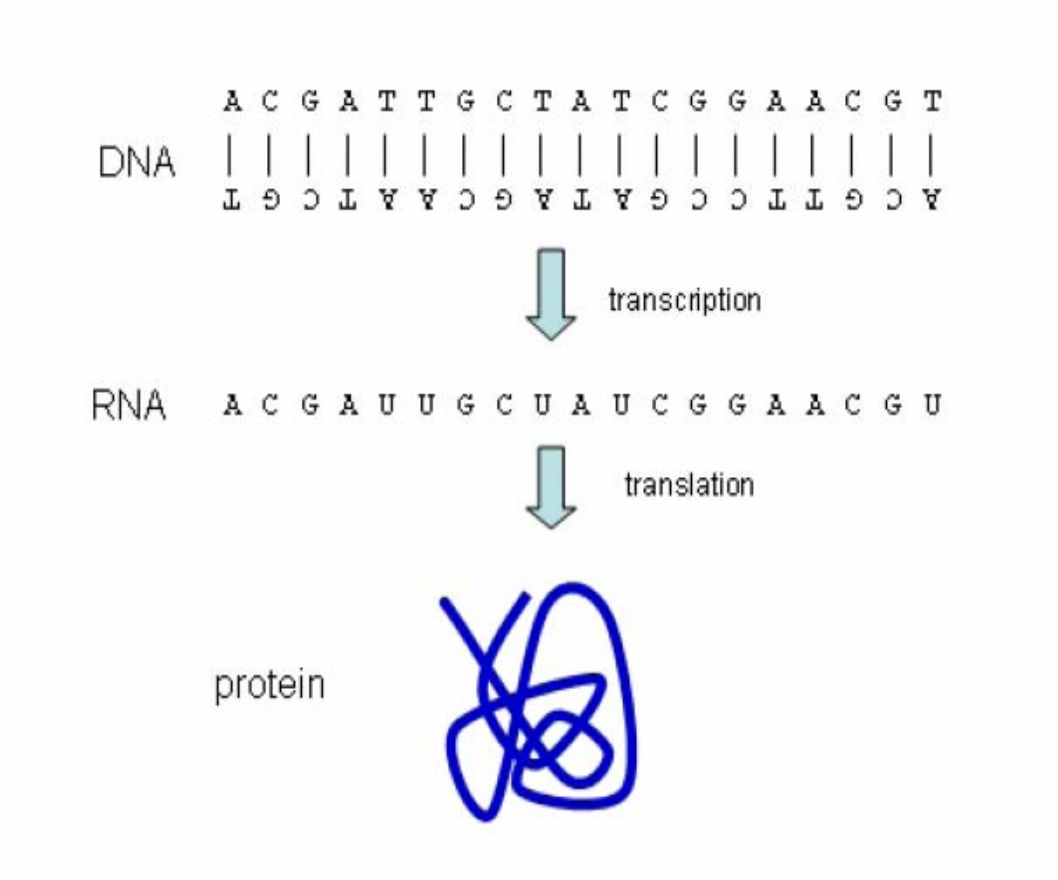


Figure 1-1: The central dogma of biology: information flows from DNA to RNA to proteins.

purposes. The structure and shape of the protein molecule (how the long chain of amino acids folds in 3-dimensional space) is relevant to the job the protein performs. Much work has been done on protein structure determination, as it gives clues to the protein's function.

Protein structure can be described at various levels. The primary structure is the amino acid sequence itself. The secondary structure and tertiary structure describe how the backbone of the protein is arranged in 3-dimensional space. The backbone of the protein makes hydrogen bonds with itself, causing it to fold up into arrangements known as alpha helices, beta sheets and random coils. Alpha helices are formed when the backbone twists into right-handed helices. Beta sheets are formed when the backbone folds back on itself to make pleats. Random coils are neither random, nor coils, but are connecting loops that join together the alpha and beta regions. The alpha, beta and coil components are what is known as secondary structure. The secondary structures then fold up to give a tertiary structure to the protein. This makes the protein compact and globular.

Other properties of proteins are also useful when determining function. Areas of hydrophobicity and polarity determine the shape of a protein and sites of interaction. The sequence length and molecular weight, and even just the ratios of the various amino acids have a bearing on the function of the protein. Sharing common patterns with other protein sequences, or common domains, can mean that the proteins have related function or evolved from a common ancestor. Evolutionary history or phylogeny of a protein can be used to understand why a protein was necessary and what its possible roles used to be.

Genes and ORFs

Genes are the units of heredity. They are sections of DNA which encode the information needed to make an organism and determine the attributes of that organism. Gene-finding programs are used to hypothesise where the genes lie in a DNA sequence. When an appropriate stretch of DNA (reasonable length, starting and ending with the right parts, etc.) is found, it is labeled as an Open Reading Frame or ORF - a putative gene. Most of the work in this thesis will use the word gene.

DNA

DNA is the molecular code of cells. It is a long chain molecule, consisting of a backbone of alternate sugar and phosphate groups, with a base attached to each sugar. There are 4 different bases which can be attached, and the sequence of the bases along the backbone makes the code. The bases are Adenine (A), Guanine (G), Cytosine (C), and Thymine (T). From a computer science /information processing perspective we would normally be dealing with DNA as a long string made up of the 4 letters A, G, C and T. The main purpose of DNA is to encode and replicate the information needed to make proteins.

The 4 bases of DNA are used in different combinations to code for the all the 20 amino acids that make proteins. A triplet of DNA bases is used to code for each amino acid. Figure 1-2 gives an example of this coding. As $4^3 = 64$, not 20, there is some redundancy in this coding, and there are several different ways to code for some amino acids (though when there are several ways they tend to be closely related). Each triple of DNA is known as a codon. Apart from the codons which are used for amino acids, three

of the triples are used to encode "stop" codons, which tell the cellular machinery where to stop reading the code.

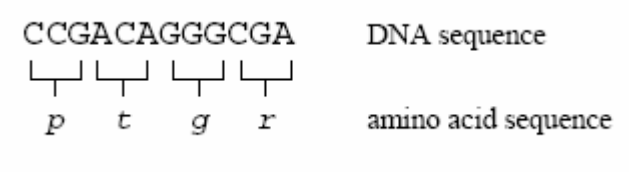


Figure 1-2: The DNA sequence is translated into a sequence of amino acids. Three DNA bases translate to one amino acid.

DNA is double stranded. It exists as two long chain molecules entwined together in the famous double helix. The two strands have complementary base pairing, so each C in one strand is paired with a G in the other and each A with a T. So when the size of DNA is quoted, it is usually in "base pairs". To give some idea of the size of the data: the DNA in the human genome is approximately $3 * 10^9$ base pairs (International human genome sequencing consortium, 2001), in the yeast genome *S. cerevisiae* it is approximately $13 * 10^6$ base pairs (Goffeau et al., 1996), and in the bacterium M. tuberculosis it is approximately $4 * 10^6$ base pairs (Cole et al., 1998).

Not all DNA codes for proteins. In mammals only about 5-10% does so. This percentage is much higher in bacteria (e.g. 90% coding in M. tuberculosis, 50-60% coding in M. leprae). The reason for the large amount of non-coding DNA is somewhat

unclear, but it includes promoter and regulatory elements, highly repetitive DNA, and so-called “junk” DNA. There are theories which suggest some “junk” is for padding, so that the DNA is folded up in the correct position, and others which say it is the remnants of DNA which used to be coding, but has now become defunct or miscopied.

RNA

DNA is translated to proteins via RNA. RNA is a nucleic acid, very similar to DNA but single stranded, and the 4 bases of RNA are A, G, C and U (Uracil replaces Thymine). RNA is used for several roles in the cell. Its primary role is to take a copy of one of the strands of DNA. This piece of RNA (known as messenger RNA) might then undergo splicing to remove introns, pieces of sequence which are non-coding, which interrupt the coding regions (exons) of a gene. Finally, the sequence of bases of RNA are then translated into amino acids to make the protein. Measurement of the RNA being produced (“expressed”) in a cell can be used to infer which proteins are being produced. The process of transcribing the gene’s DNA sequence into mRNA that severs as a template for protein production is known as gene expression. Thus, the tem “gene expression” is of particular importance. In a given tissue sample, certain genes may be over expressed, meaning the corresponding RNA production is higher than normal giving a higher probability of synthesis of the corresponding proteins. Similarly, certain genes may be underexpressed in a tissue sample.

Gene function

Even after a genome is fully sequenced, and the ORFs (or putative genes) have been located, we typically still do not know what many of them do. At the current time, approximately 40% of yeast ORFs have unknown function, and this figure is normal - in

fact yeast is one of the best studied organisms. The functions of genes are usually determined either by sequence similarity to already known sequences, or by "wet" biology.

Functional genomics by biology

Previously biologists would work on discovering the function of just a few genes of interest, but recently there has been an increase in work on a genome-wide scale. For example, now there are genome wide knockout experiments where the genes are disrupted or "knocked out" and the organism grown under different conditions to see what effect the gene has if it is missing (Ross-Macdonald, 1999). And there are experiments to look at the genome wide "expression" of cells, that is, analysis of which RNA is currently being produced in the cell. Expression data can then be used to infer which genes are switched on under different environmental conditions, and hence the biological role of the genes. Ways to measure the expression of genes in a cell include Northern blot analysis and SAGE. More recently, experiments are being done on a genome-wide scale with microarrays, a technique which can take a sample of the production of RNA in the cell at a point in time (DeRisi et al., 1997; Eisen et al., 1998; Zhang, 1999). Microarray technology has grown extremely popular and standard microarrays are being mass produced and widely used. Winzeler and Davis (1997) describes various of the biological methods for functional genomics that have been applied to yeast. This includes expression analysis, proteomics and large-scale deletion and mutational analysis. Oliver et al. (1998) surveys a similar collection of techniques, with the added inclusion of metabolomic analysis. Functional genomics is currently a major area of research, as can be seen for example by the special supplement to Nature

magazine, an “Insight” section devoted to functional genomics (Nature, 15th June 2000, 405(6788)).

Microarray expression analysis is one of the most popular methods of functional genomics. Analysis of expression data can be used to infer similar functions for genes which show similar expression patterns. Most expression analysis uses unsupervised clustering, but other methods have also been tried. Supervised learning by support vector machines has been used (Brown et al., 2000) to predict gene function. Rough sets have also been used to predict gene function from human expression data using the GeneOntology classification (Hvidsten et al., 2001) and the Rosetta toolkit. Rosetta generates if-then rules using rough set theory, and has been used in several medical applications (Komorowski & Øhrn, 1999). In the next two sections, we will discuss the microarray technology.

1.3 Introduction to Microarrays

All living cells contain chromosomes, large pieces of DNA containing hundreds or thousands of genes, each of which specifies the composition and structure of a single protein. Proteins are the workhorse molecules of the cell, responsible, for example, for cellular structure, producing energy, and for reproducing the human chromosomes. Every cell in an organism has the same set of chromosomes, but they can have very distinct properties. This is due to differences in the abundance, state, and distribution of cell proteins. The changes in protein abundance are in turn partly determined by changes in the levels of messenger RNAs (mRNAs), which are nucleic acid polymers that shuttle information from chromosomes to the cellular machines that synthesize new proteins.

The recently developed DNA microarray technology (1990) makes it possible to quickly, efficiently and accurately measure the relative representation of each mRNA and related gene expression data in the total cellular mRNA population. A DNA experiment consists of measurements of the relative representation of a large number of mRNA species (typically thousands or tens of thousands) in a set of related biological samples, e.g. time-points taken during a biological process or clinical samples taken from different patients. Each experimental sample is compared to a common reference sample and the result for each gene is the ratio of the relative abundance of the gene in the experimental sample compared to the reference. The results of such experiments are represented in a table, with each row representing a gene, each column a sample, and each cell the log(base 2)-transformed expression ratio of the appropriate gene in the appropriate sample.

The whole microarray process is shown in Figure 1-2. The DNA samples (which may be several thousands) are fixed to a glass slide in a tiny well, each at a known position in the array. A target sample and a reference sample are labeled with red and green dyes, respectively, and each is hybridized on the slide. Using a fluorescent microscope and image analysis, the log(green/red) intensities of mRNA hybridizing at each site is measured. The result is a few thousand numbers, typically ranging from say -4 to 4, one per (x,y) position on the glass slide, measuring the expression level of each gene in the experimental sample relative to the reference sample in that position. Positive values indicate higher expression in the target versus the reference, and vice versa for negative values.

The data from a series of *M* such experiments may be represented as an *N X M* gene expression matrix, in which each of the *N* rows consists of an *M*-element expression vector for a single gene. Such a data set is usually very large.

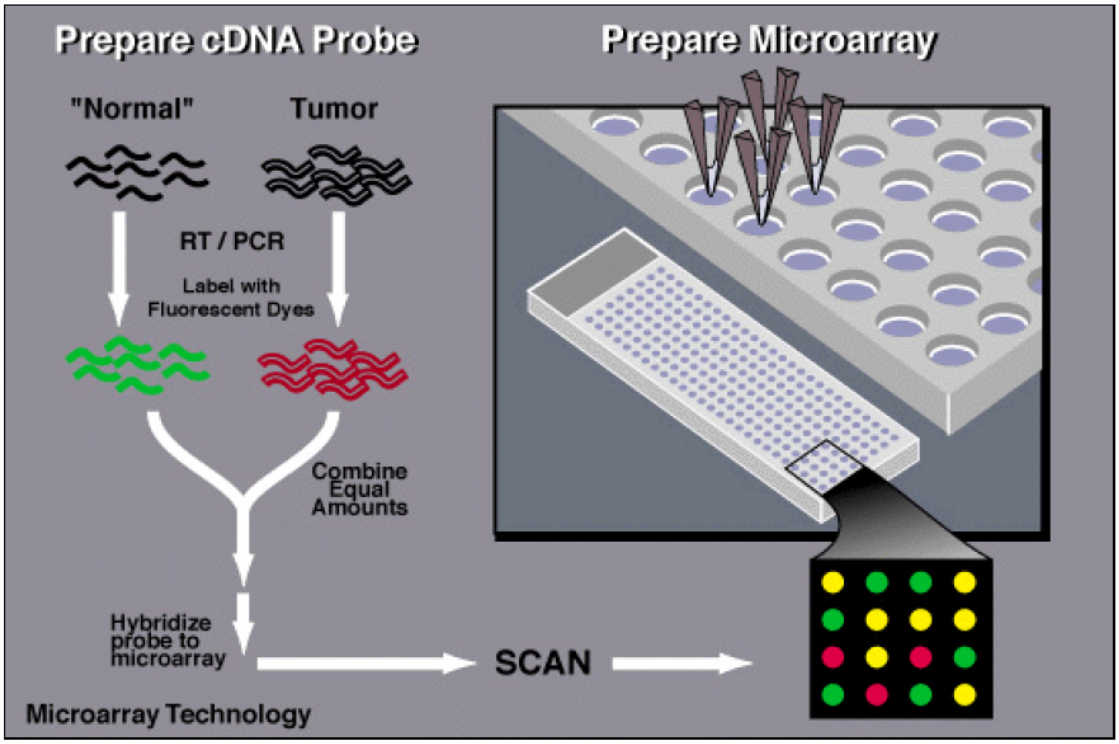


Figure 1-3. Microarray process.

1.4 Two types of Microarray

These microarrays can be divided into two main types which differ in their construction: spotted microarrays and high density oligionucleotide arrays.

1.4.1 Spotted microarray

Spotted microarrays generally use cDNA (Complimentary DNA) probes but they can also be oligonucleotides and other DNA components (van Hal et al., 2000). cDNA is a single-stranded DNA molecule synthesised in the laboratory using mRNA as a template by using reverse transcriptase. Oligonucleotides are any polynucleotide whose molecules are made up of a relatively small number of nucleotides. In other words, these are synthesized DNA sequences of small length. These cDNA products are the product of the purified polymerase chain reaction (PCR) generated from cDNA libraries or clone collections (Schulze and Downward, 2001). They are generally gene fragments greater than several hundred base pairs long (Harrington et al., 2000). These probes are deposited onto a solid surface in defined locations by an xyz robot (Macgregor and Squire, 2002). There are two main methods to deposit the spots (van Hal et al., 2000):

1. Active dispensers. This method is based around inkjet technology and uses either piezoelectric or solenoid valve delivery to drop the spot onto the solid surface.

2. Passive dispensers. This method applies DNA solution with a pin that touches the solid surface. A higher density of spots can be achieved using this approach.

The solid surface is normally a specially coated glass microscope slide but different surfaces such as nylon membranes, gold coated slides and other materials which form a 3 dimensional matrix have also been used (van Hal et al., 2000) . For a glass slide

different methods can be used to attach the cDNA including giving the glass slide a positively charged layer to bind the negatively charged DNA fragments or coating the glass surface in reactive groups and modifying the cDNA so that it can be covalently bonded to the glass surface (van Hal et al., 2000). Spots contain a minimum volume of about 50 pl of DNA solution (van Hal et al., 2000) and typical spot sizes range from 80-150 μm in diameter (Macgregor and Squire, 2002). A maximum of 80 000 spots can be fitted onto a single glass slide (Macgregor and Squire, 2002).

In spotted microarrays it is standard practice to compare the gene expression of two biological samples on one chip. The mRNA is prepared in such a way that the two different expression levels can be measured. This is mainly because of a lack of consistency in the construction of spotted microarrays that makes it unwise to compare data between them.

1.4.2 Synthesised high density oligonucleotide microarrays

These microarrays are constructed commercially to an extremely high density and accuracy using short oligonucleotides with a length of between 20 and 25 nucleotides as the probes (Schulze and Downward, 2001). Two firms construct these high density arrays, Affymetrix and Agilent Technologies (who have licensed the ink-jet technology to construct the microarrays from Rosetta Inpharmatics) (Schulze and Downward, 2001).

These GeneChips^{TM} are produced by synthesising tens of thousands of short oligonucleotides in situ onto glass wafers, one nucleotide at a time, using a modification of semiconductor photolithography technology (Macgregor and Squire, 2002). Initially the solid surface is embedded with linker molecules that have a photo-labile protective group. A mask is placed over the slide and illuminated with light thus selectively

removing the exposed protective group. The microarray is then incubated with a solution containing a particular photo-protected nucleotide which will only couple with the exposed linker molecules. The excess molecules are then removed and another mask and exposure to light is used to de-protect other areas of the microarray. A solution containing a different photo-protected nucleotide is then exposed to the microarray and hence is coupled to the newly exposed areas. This process is repeated for all four nucleotides. The photo-protected groups are replaced with photo-sensitive groups and the process continues onto the next layer. In this way oligonucleotides of any code can be constructed (van Hal et al., 2000). This technique produces an extremely high density microarray.

The GeneChips^{TM} have a number of strategies in their design to help minimise crosstalk. Crosstalk occurs when RNA from one gene binds with an oligonucleotide that represents another gene and so misrepresenting the amount of expression from both genes. The first strategy is that next to each oligonucleotide is a mismatched oligonucleotide which represents an identical copy of the oligonucleotide except that its central nucleotide is changed to a different nucleotide (e.g., GATTCG and GATGCG). The amount of gene expression associated with the mismatch provides a measure of the background crosstalk that can occur for that particular oligonucleotide and hence this can be used to more accurately produce the exact expression signal for the gene. The second strategy is to have between 15-20 different oligonucleotides/mismatch pairs for each gene on each chip (Harrington et al., 2000). These oligonucleotides are specially designed to uniquely represent the gene. When the gene expression from all these oligonucleotides is combined, an accurate measure of the expression is obtained. Additionally, as the large

number of oligonucleotides are scattered across the microarray, it means that if a part of the microarray is damaged (through for example dust or a scratch) then an estimate for the gene expression can still be made from the remaining oligonucleotides that represent that gene.

Affymetrix produces a number of different microarrays each having a different composition of genes represented on the microarray. The most important ones for this project are the Human U133A and B GeneChips^{TM} each of which represents 19,000 different genes, in combination they represent a total of 33,000 genes, which covers the vast majority of the human genome. There is also a more basic microarray that contains the 8,700 best annotated genes and these are called Human Focus arrays. Other species represented with microarrays include mouse, rabbit, rat, drosophila, *Arabidopsis*, *C. elegins*, yeast and *E. coli*.

1.5 Roadmap of chapters in the thesis

The organization of this thesis will be as follows:

- Chapter 1 presents our thesis, the motivation behind it, the methodology we followed and the issues we considered in testing our nhypethesis and introduces the functional genomic background which is essential to understand the impact of our present work. .
- Chapter 2 surveys the data mining and machine learning methods that are applied to computational biology and bioinformatics.
- In Chapter 3 we describe the issues of biomedical literature text analysis. Detailed keyword extraction process is performed and an optimum conditions for

keyword identification are presented. The keywords that describe the most prominent common functions of the genes are extracted to assist hypothesis generation. Functionally irrelevant words are filtered by a stop list.

• Chapter 4 deals with using the results of the functional keyword extraction process and performing gene clustering by functional key word association. The keywords are used as feature vectors for clustering algorithms (Bond Energy Algorithm). Genes are assigned into functionally relevant clusters based on shared keywords that suggest the principal biological functions of each cluster.

• Chapter 5 tests the claim that biomedical literature and the functional keyword extraction can yield a superior functional grouping of genes compared to other publicly available sources such as microarray and phylogenetic databases. We investigate yeast (*Saccharomyces cerevisiae*) gene function prediction from different data sources. An empirical comparative study of yeast gene function classification is performed using different data sources, namely microarray data, and phylogenetic data, from Munich Information Center for Protein Sequences Yeast Genome Database (MYGD) (*http://mips.gsf.de/genre/proj/yeast/index.jsp*). Comparative analysis of above databases with literature text data, as well as results of combining of these three data sources are reported.

• In Chapter 6 we present an application to further support our original thesis. We design a biomedical text categorization system based on support vector machines . This system is applied to categorize Human Genome Epidemiology (HuGE) relevant articles from PubMed database into the Center for Disease Control and Prevention's (CDC) Human Genome Epidemiology Network, or HuGENet™

(http://www.cdc.gov/genomics/hugenet/) published literature database. This work has been performed with another Ph.D. student in Biology, Ms. Nalini Polavarapu.

• Finally Chapter 7 presents ideas for future work, possible future experiments, and summarizes the original contributions to knowledge that this thesis has made.

1.6 Summary

This chapter described the motivation behind the thesis, which stems from providing a systematic approach that will be generically useful in interpreting the result of microarray experiments. We discussed our basic thesis, the motivation behind it and the detailed plan to test and validate our hypothesis as well as to investigate its applications. We then presented the basic concepts in biology which are important for an understanding of the contribution of this thesis. We discussed the microarray techniques in detail so that the reader understands the nature and the high volume of data typically provided by a single microarray experiment. In the next chapter, we describe the previous related work on microarray data cluster analysis and text mining application in bioinformatics.

CHAPTER 2

RELATED PREVIOUS WORK

2.1 Microarray data clustering analysis approaches

DNA microarray technology has made it possible to simultaneously monitor the expression levels of thousands of genes during important biological processes and across collections of related samples. Elucidating the patterns hidden in gene expression data offers a tremendous opportunity for an enhanced understanding of functional genomics. However, the large number of genes and the complexity of biological networks greatly increase the challenges of comprehending and interpreting the resulting mass of data, which often consists of millions of measurements. A first step toward addressing this challenge is the use of clustering techniques, which is essential in the data mining process to reveal structures within the data and identify interesting patterns in the underlying data.

Cluster analysis seeks to partition a given data set into groups based on specified features so that the data points within a group are more similar to each other than the points in different groups. A very rich literature on cluster analysis has developed over the past three decades. Many conventional clustering algorithms have been adapted or directly applied to gene expression data, and also new algorithms have recently been proposed specifically aiming at gene expression data. These clustering algorithms have been proven useful for identifying biologically relevant groups of genes and samples.

In this chapter, we present specific challenges pertinent to clustering techniques and introduce several representative approaches.

2.1.1 K-means

The K-means algorithm (McQueen et al., 1967) is a typical partition-based clustering method. Given a pre-specified number *K*, the algorithm partitions the data set into *K* disjoint subsets (clusters) which optimize the following objective function:

$$E = \min(\sum_{i=1}^{K} \sum_{O \in C_i} | O - \mu_i |^2) \qquad (2\text{-}1)$$

Here, O is a data object in cluster C_i and μ_I is the centroid (mean of objects) of C_i. Thus, the objective function E tries to minimize the sum of the squared distances of objects from their cluster centers.

The K-means algorithm is simple and fast. The time complexity of K-means is $O(l * k * n)$, where l is the number of iterations and k is the number of clusters. However, the K-means algorithm has several drawbacks as a gene-based clustering algorithm. First, the number of gene clusters in a gene expression data set is usually unknown in advance. To detect the optimal number of clusters, users usually run the algorithms repeatedly with different values of k and compare the clustering results. For a large gene expression data set which contains thousands of genes, this extensive parameter fine-tuning process may not be practical. Second, gene expression data typically contain a large amount of noise; however, the K-means algorithm forces each gene into a cluster, which may cause the algorithm to be more sensitive to noise than other clustering algorithms described below (Sherlock, 2000; Smet et al., 2002).

2.1.2 Hierarchical clustering

In contrast to partition-based clustering, which attempts to directly decompose the data set into a set of disjoint clusters, hierarchical clustering generates a hierarchical series of nested clusters which can be graphically represented by a tree, called *dendrogram*. The branches of a dendrogram not only record the formation of the clusters but also indicate the similarity between the clusters. By cutting the dendrogram at some arbitrary level, we can obtain a specified number of clusters. By reordering the objects such that the branches of the corresponding dendrogram do not cross, the data set can be arranged with similar objects placed together.

Hierarchical clustering algorithms can be further divided into *agglomerative* approaches and *divisive* approaches based on how the hierarchical dendrogram is formed. Agglomerative algorithms (bottom-up approach) initially regard each data object as an individual cluster, and at each step, merge the closest pair of clusters until all the groups are merged into one cluster. Divisive algorithms (top-down approach) starts with one cluster containing all the data objects, and at each step split a cluster until only singleton clusters of individual objects remain. For agglomerative approaches, different measures of *cluster proximity*, such as single link, complete link and minimum-variance (Kaufman and Rousseeuw, 1990; Dubes and Jain, 1988), are used to derive various merge strategies. For divisive approaches, the essential problem is to decide how to split clusters at each step. Some are based on heuristic methods such as the deterministic annealing algorithm, while many others are based on the graph theoretical methods (Alon et al., 1999).

Eisen et al. (1998) applied an agglomerative algorithm called UPGMA (Unweighted Pair Group Method with Arithmetic Mean) and adopted a method to graphically represent the clustered data set. In this method, each cell of the gene expression matrix is colored on the basis of the measured fluorescence ratio, and the rows of the matrix are re-ordered based on the hierarchical dendrogram structure and a consistent node-ordering rule. After clustering, the original gene expression matrix is represented by a colored table (a *cluster image*) where large contiguous patches of color represent groups of genes that share similar expression patterns over multiple conditions (Figure 2-1).

Alon et al. (1999) split the genes through a divisive approach, called the *deterministic-annealing algorithm (DAA)* (Rose et al., 1990; Rose, 1998). First, two initial cluster centroids C_j, j = 1, 2, were randomly defined. The expression pattern of gene k was represented by a vector $\vec{g}_k$, and the probability of gene k belonging to cluster j was assigned according to a two-component Gaussian model:

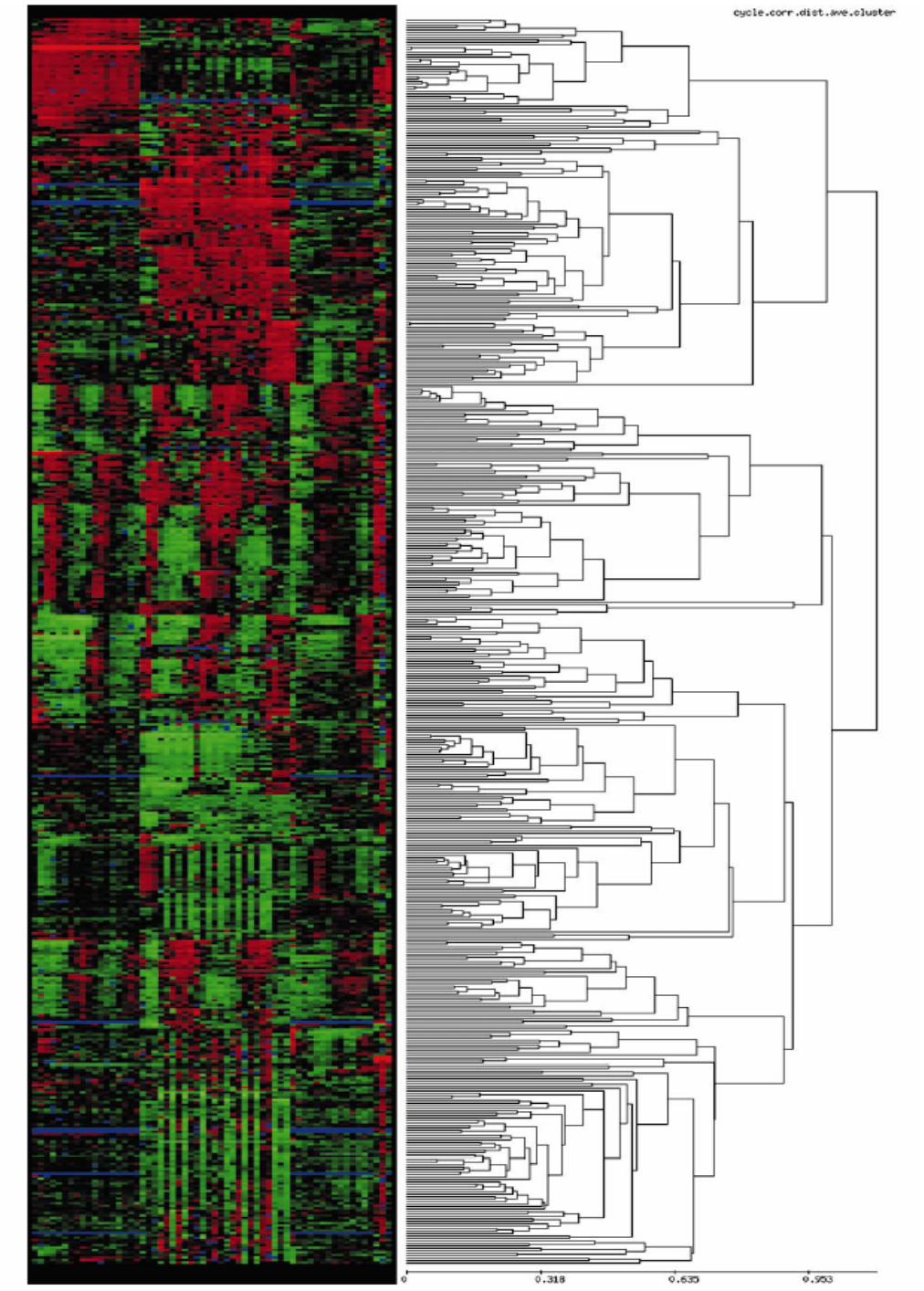


Figure 2-1. Hierarchical clustering of gene expression matrices.

$$P_j(\vec{g}_k) = \exp(-\beta |\vec{g}_k - C_j|^2) / \sum_j \exp(-\beta |\vec{g}_k - C_j|^2) \quad (2\text{-}2)$$

The cluster centroids were recalculated by

$$C_j = \sum_k \vec{g}_k P_j(\vec{g}_k) / \sum_k P_j(\vec{g}_k) \quad (2\text{-}3)$$

An iterative process (the *EM algorithm*) was then applied to solve P_j and C_j. For $\beta = 0$, there was only one cluster, $C_1 = C_2$. When β was increased in small steps until a threshold was reached, two distinct, converged centroids emerged. The whole data set was recursively split until each cluster contained only one gene.

Hierarchical clustering not only groups together genes with similar expression pattern but also provides a natural way to graphically represent the data set. The graphic representation allows users a thorough inspection of the whole data set and affords an initial impression of the distribution of data. Eisen's method is much favored by many biologists and has become the most widely-used tool in gene expression data analysis (Eisen et al., 1998; Allon et al., 1999; Iyer et al., 1999; Perou et al., 1999; Alizadeh et al., 2000). However, the conventional agglomerative approach suffers from a lack of robustness (Tamayo et al., 1999), i.e., a small perturbation of the data set may greatly change the structure of the hierarchical dendrogram. Another drawback of the hierarchical approach is its high computational complexity. To construct a "complete" dendrogam (where each leaf node corresponds to one data object, and the root node

corresponds to the whole data set), the clustering process should take $\frac{n^2 - n}{2}$ merging (or splitting) steps. The time complexity for a typical agglomerative hierarchical algorithm is $O(n^2 \log n)$ (Jain et al., 1999). Furthermore, for both agglomerative and divisive approaches, the hierarchical clustering prevents the refinement of the previous clustering. If a "bad" decision is made in the initial steps, it can never be corrected in the following steps.

2.1.3 Other clustering techniques in microarray data analysis (Valafar, 2002)

Sasik et al. (2001) have presented the Percolation Clustering approach to clustering of gene expression patterns based on the mutual connectivity of the patterns. Unlike SOM or *k*-means which force gene expression data into a fixed number of predetermined clustering structures, this approach is to reveal the natural tendency of the data to cluster, in analogy to the physical phenomenon of percolation.

GA/KNN is another algorithm described by Li, et al. (2001) "This approach combines a Genetic Algorithm (GA) and the *k*-Nearest Neighbor (KNN) method to identify genes that can jointly discriminate between different classes of samples. The GA/KNN is a supervised stochastic pattern recognition method. It is capable of selecting a subset of predictive genes from a set of large noisy data for sample classification" (Li et al., 2001). We will discuss genetic algorithm next.

***Genetic Algorithm*:** Holland (1975) invented the genetic algorithm (GA) in 1975. GA is essentially an optimization technique that was inspired by mutation (in nature) that gives rise to biological evolution. In GA, the coordinates of points in the problem space are organized as a sequence, much like sequences of genes. The process of searching for a maximum or a minimum is accomplished by mutating the sequence, and hence arriving

at a new coordinate. At each new coordinate, the function is evaluated, and if the new point is determined to be more optimal than those previously observed, the new point is stored as the new extrema (minimum or maximum). GA has been used in a variety of applications in sequencing. For instance, in DNA fragment assembly, the work of Parsons *et al* (1995), Cedeno and Vemuri (1993), Fickett and Cinkosky (1993) can be mentioned. Zhang and Wong (1997) applied GA to multiple molecular sequence alignment. Most varieties of GA differ in the way the sequences are mutated, and hence search the problem space in different patterns. As an example of a variety, Valafar's *distributed global optimization* (*DGO*) algorithm can be mentioned (Valafar et al., 1996). Hybrid systems also exist in which, for instance, a neural network is built using a GA algorithm as the learning algorithm. For example, Valafar (1996) used the DGO as the learning algorithm of a multilayer, feed-forward neural network to develop a system that could automatically identify the chemical structure of a group of complex carbohydrates and some glycoproteins from their ^{1}HNMR spectra (Valafar et al., 1996; Valafar et al., 1998)

***Artificial Neural Network*:** Artificial neural networks (ANNs) belong to the adaptive class of techniques in *machine learning*. ANNs have been used as solutions to various types of problems (e.g. pattern recognition, prediction, estimation, etc.). However, ANNs' success as an intelligent pattern recognition methodology has been advertised most prominently. ANNs were inspired by the brain (a biological neural network). Most models of ANNs are organized in the form of a number of processing units (also called artificial neurons, or simply neurons (McCulloch and Pitts, 1943)), and a number of weighted connections (artificial synapses) between the neurons. The process of building

an ANN (similar to its biological inspiration) involves a learning episode (also called training). During the learning episode, the network observes a sequence of recorded data, and adjusts the strength of its synapses according to a learning algorithm and based on the observed data. The process of adjusting the synaptic strengths in order to be able to accomplish a certain task (much like the brain) is called "learning". Learning algorithms are generally divided into two types, supervised and unsupervised. Supervised algorithms require labeled training data. In other words, they require more a priori knowledge about the training set. The most important, and attractive, feature of ANNs is their capability of learning (generalizing) from example (extracting knowledge from data). ANNs can do this without any prespecified rules that define intelligence or represent an expert's knowledge. This feature makes the ANN an attractive choice for gene expression analysis and sequencing. ANNs were the first group of machine learning algorithms to be used on a biological pattern recognition problem (Selaru et al., 2002).

Due to their power and flexibility, ANNs have even been used as tools for selection of relevant variable, which can in turn greatly increase the expert's knowledge and understanding of the problem. For instance, Selaru et al. (2002) used ANNs to distinguish among subtypes of neoplastic colorectal lesions. They then used the trained ANN to identify the relevant genes that are used to make this distinction. Specifically, the authors evaluated the ability of ANNs in distinguishing between *complementary DNA* (cDNA) microarray data (8064 clones) of two types of colorectal lesions (sporadic colorectal adenomas and cancers or SAC, and inflammatory bowel disease-associated or IBD-associated dysplasias and cancers). Salura and colleagues (2002) report the failure of hierarchical clustering to make the above distinction, even when all 8064 clones were

used. ANNs not only correctly identified all twelve samples of the test set (3 IBDNs and 9 SACs), but also helped identify the subset of genes that were important to make this diagnostic distinction: "Using an iterative process based on the computer programs GeneFinder, Cluster, and MATLAB, we reduced the number of clones used for diagnosis from 8064 to 97." Using the 97 clones, even the cluster analysis was then able to make the correct distinction between the two types of lesions. The authors conclude: "Our results suggest that ANNs have the potential to discriminate among subtly different clinical entities, such as IBDNs and SACs, as well as to identify gene subsets having the power to make these diagnostic distinctions."

There is a very large body of research that has resulted in a large number of ANN designs. For a more comprehensive review of the various ANN types, see (Rumelhart and McClelland, 1988; Rojas, 1996). In this chapter, we discuss only two types that has been used in sequencing.

Layered, feed-forward neural networks: This is a class of ANNs whose neurons are organized in layers. The layers are normally fully connected, meaning that each element (neuron) of a layer is connected to each element of the next layer. However, self-organizing varieties also exist in which a network starts either with a minimal number of synaptic connections between the layers and adds to the number as training progresses (*constructive*), or starts as a fully connected network and prunes connections based on the data observed in training (*destructive*) (Rumelhart and McClelland, 1988; Rojas, 1996).

Backpropagation (Rumelhart and McClelland, 1988; Rojas, 1996) is a learning algorithm that, in its original version, belongs to the gradient descent optimization

methods (Hassoun, 1995). It is the most popular learning algorithm that has been used to train layered ANNs. A large number of varieties of the algorithm have been developed that use a number of various optimization techniques (Rojas, 1996). The combination of backpropagation learning algorithm and the feed-forward, layered networks provides the most popular type of ANNs. These ANNs have been applied to virtually all pattern recognition problems, and are typically the first networks tried on a new problem. The reason for this is the simplicity of the algorithm, and the vast body of research that has studied these networks. As such, in sequencing many researchers have also used this type of network as a first line of attack. Examples can be mentioned in (Wu, 1995; Wu et al., 1996). Wu (1995) developed a system called gene classification artificial neural system (GenCANS), which is based on a three layered, feed-forward backpropagation network. GenCANS was designed to "classify new (unknown) sequences into predefined (known) classes. It involves two steps, sequence encoding and neural network classification, to map molecular sequences (input) into gene families (output)". The same type of network has been used to perform rapid searches for sequences of proteins (Wu et al., 1996)

Other examples can be mentioned in Snyder and Stormo's work in designing a system called *GeneParser* (Snyder and Stormo, 1995). Here authors experimented with two variations of a single layer network (one fully connected, and one partially connected with an activation bios added to some inputs), as well as a partially connected two-layer network. The authors use dynamic programming as the learning algorithm in order to train the system for protein sequencing.

As mentioned before, the advantage of these networks is in their simplicity of implementation and understanding of the underlying mathematics. Because of the large

body of research conducted on these networks, there are a large number of public domain software packages available that implement virtually all the different varieties of the network.

Although these networks are theoretically capable of separating a problem space into appropriate classes irrespective of the complexity of the separation boundaries, one of the "classical" disadvantages of these networks is that a certain amount of a priori knowledge is required in order to build a useful network. A crucial factor in training a useful network is its size (number of layers, size of layers, and number of synaptic connections). In many cases, it takes a large number of simulations before a close-to-optimum size of the network is found. If the network is designed to be larger than this optimum size, it will memorize (also called over-fit) the data rather than generalizing and extracting knowledge. If the network is chosen to be smaller than the optimum size, the network will never learn the entire task at hand. An attractive alternative to these networks are self-organizing networks which automatically, or semi-automatically, determine the optimal size from the data at hand.

***Self-organizing map*:** The Self-Organizing Map (SOM) was developed by Kohonen (1984), on the basis of a single layered neural network. SOM is an unsupervised artificial neural network. It maps high-dimensional data into a two-dimensional representation space, and similar data may be found in neighboring regions The data objects are presented at the input, and the output nods are organized with a simple neighborhood structure such as a two dimensional $p * q$ grid (Figure 2-2). Each node of the neural network (typically called a "cell" of the p by q grid) is associated with a reference vector, and each data point (input data) is "mapped" to the node with the "closest" reference

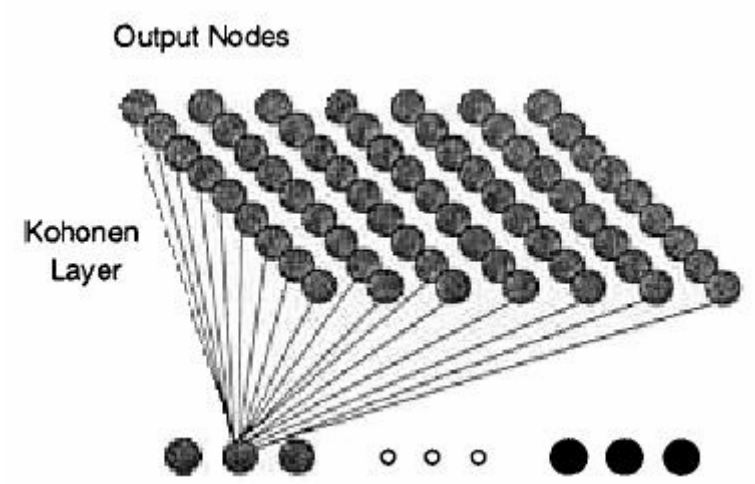


Figure 2-2. Schematic representation of a self-organizing map method

vector. In the process of running the algorithm, each data object acts as a training sample which directs the movement of the reference vectors towards the denser areas of the input vector space, so that those reference vectors are trained to fit the distributions of the input data set. When the training is complete, clusters are identified by mapping all data points to the output neurons.

The Self-Organizing Map Algorithm may be described by the following steps:

I. The set of input data is described by a real vector x(t) where t is the index of the data. Each node *i* in the map contains a model vector $m_i(t)$, which has the same number of elements as the input vector x(t).

II. The SOM algorithm performs a regression process. The initial values of the components of the model vector, $m_i(t)$, may be selected at random.

III. Any input data is thought to be mapped into the location, the $m_i(t)$ of which matches best with x(t) in some metric (e.g. Euclidean distance). The self-organizing algorithm creates the ordered mapping as a repetition of the following basic tasks:

1. An input vector x(t) is compared with all the model vectors $m_i(t)$. The best-matching unit (node) on the map, i.e., the node where the model vector is most similar to the input vector in some metric (e.g. Euclidean) is identified. This best matching unit is often called the winner.

2. The model vectors of the winner and a number of its neighboring nodes in the array are changed towards the input vector according to the learning principle. The basic ideal in the SOM learning process is that, for each sample input vector x(t), the winner and the nodes in its neighborhood are changed closer to x(t) in the input data space. During the learning process, individual changes may be contradictory, but the net outcome in the process is that ordered values for the $m_i(t)$ emerge over the array. If the number of available input samples is restricted, the samples must be presented repeatedly to the SOM algorithm. Adaptation of model vectors in the learning process may take place according to the following equations:

$$m_i(t+1) = m_i(t) + \alpha(t)[x(t) - m_i(t)] \text{for each } i \in N_c(t),$$

$$m_i(t+1) = m_i(t) \text{ otherwise,}$$

where, t is the discrete-time index of the variables, the factor $\alpha(t) \in [0,1]$ is a scalar that defines the relative size of the learning step, and $N_c(t)$ specifies the neighborhood around the winner in the map array. At the beginning of the

learning process the radius of the neighborhood is fairly large, but it is made to shrink during learning. This ensures that the global order is obtained already at the beginning, whereas towards the end, as the radius gets smaller, the local corrections of the model vectors in the map will be more specific. The factor $\alpha(t)$ also decreases during learning.

One of the remarkable features of SOM is that it generates an intuitively-appealing map of a high-dimensional data set in 2*D* or 3*D* space and places similar clusters near each other, while *K*-means clustering captures local features of the data but fails to provide an organization scheme. The node training process of SOM provides a relatively more robust approach than K-means to the clustering of highly noisy data (Tamayo et al., 1999; Herrero et al., 2001). However, SOM requires users to input the number of clusters and the grid structure of the node map. These two parameters are preserved through the training process; hence, improperly-specified parameters will prevent the recovering of the natural cluster structure. Furthermore, if the data set is abundant with irrelevant data points, such as genes with invariant patterns, SOM will produce an output in which this type of data will populate the vast majority of clusters (Herrero et al., 2001). In this case, SOM is not effective because most of the interesting patterns may be merged into only one or two clusters and cannot be identified.

2.2 Limitation of microarray data clustering analysis approaches

DNA microarray data cluster analysis approaches described in section 2.1 applied clustering methods directly to the expression data, in order to find clusters of genes demonstrating similar expression patterns. The assumption motivating such search for co-expressed genes is that simultaneously expressed genes often share a common function. However, there are several reasons that cluster analysis of DNA expression data alone cannot fully address this core issue:

1. Genes that are functionally related may demonstrate strong anti-correlation in their expression levels, (a gene may be strongly suppressed to allow another to be expressed), thus clustered into separate groups, blurring the relationship between them;
2. Simultaneously expressed genes do not always share a function;
3. Genes that are expressed at different times may serve complementing roles of one unifying function;
4. Even when similar expression levels correspond to similar functions, the function and the relationships between genes in the same cluster data cannot be determined from the cluster data alone.

2.3 Text mining biomedical literature

The past decade has seen a tremendous growth in the amount of experimental and computational biomedical data, specifically in the areas of genomics and proteomics. This growth is accompanied by an accelerated increase in the number of biomedical publications discussing the findings. In the last few years, there has been a lot of interest within the scientific community in literature-mining tools to help sort through this abundance of literature and find the nuggets of information most relevant and useful for specific analysis tasks.

2.3.1 Text mining biomedical literature for discovering gene-to-gene relationships

During the last few years, there was a surge of interest in using the biomedical literature, (e.g., Andrade and Valencia, 19980; Leek, 1997; Fukuda *et al.*, 1998; Craven and Kumlien, 1999; Rindflesch *et al.*, 2000; Shatkay *et al.*, 2000; Friedman *et al.*, 2001; Jenssen *et al.*, 2001; Yandell and Majoros, 2002; Hanisch *et al.*, 2003), ranging from relatively modest tasks such as finding reported gene location on chromosomes (Leek, 1997) to more ambitious attempts to construct putative gene networks based on gene-name co-occurrence within articles (Jenssen *et al.*, 2001). Since the literature covers all aspects of biology, chemistry, and medicine, there is almost no limit to the types of information that may be recovered through careful and exhaustive mining. Some possible applications for such efforts include the reconstruction and prediction of pathways, establishing connections between genes and disease, finding the relationships between genes and specific biological functions, and much more. It is important to note that a single mining strategy is unlikely to address this wide spectrum of goals and needs (Shatkay et al., 2000).

The automated handling of text is an active research area, spanning several disciplines. These include the following: *information retrieval*, which mostly deals with finding documents that satisfy a particular information need within a large database of documents (Sahami, 1998; Salton, 1989, Witten *et al.,* 1999); *natural language processing (NLP)*, a broad discipline concerned with all aspects of automatically processing both written and spoken language (Charniak, 1993; Allen, 1995; Russell and Norvig, 1995); *information extraction (IE)*, a subfield of NLP, centered around finding explicit entities and facts in unstructured text (Cowie and Lehnert, 1996; Cardie, 1997). For instance, identifying all the positions in the text that mention a protein or a kinase (entity extraction), or finding all phosphorylation relationships to populate a table of phospohrylated proteins along with the responsible kinase (relationship extraction) are both IE tasks. Finally, *text mining* (Hearst, 1999), the combined, automated process of analyzing unstructured, natural language text in order to discover information and knowledge that are typically difficult to retrieve.

A number of groups are developing algorithms that link information from medical literature with gene names. Andrade and Valencia (1998) introduced a statistical profiling strategy that accepts user-supplied abstracts related to a protein of interest and returns an ordered set of keywords that occur in those abstracts more often than would be expected by chance. We have extended the approach in terms of investigation into background datasets, stop lists and use of other ranking measures like TFIDF instead of the Z-score used in the original work (Chapter 3). With the goal of automating the functional annotation of new proteins, Andrade et al. (1999) presented an interactive suite of programs called “Genequiz”, which searches and organizes information from many

sequence and text databases. Andrade and Bork (2000) and Perez-Iratxeta et al. (2002) developed a program that links the OMIM database of human inherited diseases to keywords derived from MEDLINE. A variety of nonstatistical approaches have also been used to organize genes. The web tool PubGene finds links between pairs of genes based on their co-occurrence in MEDLINE abstracts (Jenssen and Vinterbo 2000; Jenssen et al., 2001). Another approach (Masys *et al.* 2001), the basis of the HAPI web tool, organizes gene names according to predefined hierarchical classification systems of enzymes and diseases, and includes hyperlinks to specific MEDLINE citations responsible for the individual classifications. Still another approach (Tanabe *et al.* 1999), used by the MedMiner system, automatically retrieves functional information (both keywords and gene names related to a user-defined function) from the GeneCards database, and configures it for a PubMed search. The algorithm presents the results by the specific sentence containing the information rather than by the title, speeding review of the results if the user prefers to extract relevant sentences rather than scan through the whole abstract text. A similar method of presenting the statistically most significant sentences was used by Andrade and Valencia (1998).

The above approaches provide useful information that organizes or relates genes, but a major shortcoming is they either do not address specific functions of the genes or are constrained by functions predefined in other databases, which can be biased, incomplete, or out-of-date.

We believe that MEDLINE abstracts contain much additional, valuable information, which is comprehensive, up-to-date and unbiased in the sense that many authors contribute the information rather than one or several database administrators and

curators. Much functional information describing the genes' corresponding proteins, their cellular location, elemental functions, binding partners, biochemical pathways, etc., is encoded in keywords or phrases in the titles, subheadings, and abstract text. We propose to use several different data mining techniques to retrieve and organize this text-based information and present keywords as well as proper visual displays to the user that will reveal functional connections among various gene products. This approach will be more likely to discover novel relationships between genes since it links them by shared functional keywords rather than just reporting known interactions based on published reports; thus, genes that never co-occur in the same publication could still be linked by their shared keywords. On the contrary, unrelated genes will not be considered to be related just because they happen to be mentioned in the same article. Furthermore, instead of just indicating that there is a link between genes, our approach - clusters the genes together and describe the specific functions they share, which should enable the user to comprehend more efficiently the role(s) of these genes in the context of the known experimental conditions and subsequently allow them to form more meaningful hypotheses for investigation.

2.3.2 Classification of biomedical literature

Text Classification, or the task of automatically assigning semantic categories to natural language text, has become one of the key methods for organizing online information, such as PubMed, the large repository of publicly available scientific literature. Currently, new data is being added to it at the rate of over 1500 abstracts per week. Most biomedical researchers want to access PubMed with specific goals based on

the areas of interest. The ability to efficiently review the available literature is essential for rapid progress of research in scientific community.

There are two main approaches to text classification. One is the *knowledge engineering* approach (Hayes and Weinstein, 1990; Hayes, 1992) where the user manually defines a set of rules to encode expert knowledge regarding the correct classification of documents into given categories. The other approach is based on *machine learning* (Lehnert, 1994; Lewis and Hayes, 1994; Lewis and Ringuette, 1994; Yang and Chute, 1994; Lewis, 1995; Vapnik, 1995; Larkey and Croft, 1996; Lewis *et al.*, 1996; Dumais *et al.*, 1998; Joachims, 1998; Cohen and Singer, 1999; Yang and Liu, 1999; Riloff and Sebastiani, 2002) where a general inductive process automatically builds a text classifier by training over a set of pre-classified documents.

An example of the knowledge engineering approach is the CONSTRUE system (Hayes and Weinstein, 1990; Hayes, 1992) built by the Carnegie Group for Reuters. A typical rule in the CONSTRUE system consists of a condition defined as a disjunction of conjunctive clauses (a *DNF* formula) followed by the resulting category. For example, the following rule in CONSTRUE identifies articles that should be categorized as relevant to *wheat*:

```
If ((wheat & farm) or
    (wheat & commodity) or
    (bushels & export) or
    (wheat & tones) or
    (wheat & winter & soft))
then Wheat
else ~Wheat.
```

The main drawback of this approach is known as the *knowledge acquisition bottleneck.* The rules must be manually defined by a knowledge engineer interviewing a domain expert. If the set of categories is modified, these two professionals must intervene again. Hayes *et al.* (1992, 1990) reported a 90% breakeven between precision and recall on a small subset of the Reuters test collection (723 documents). However, it took a tremendous effort (several man years) to develop a system, and the test set was not significant to validate the results. It is not clear that these superb results scale up in a larger system.

The machine learning (ML) approach is based on the existence of a training set of documents, already classified into a predefined set of categories. A diagram of a typical ML-based categorization system is shown in Figure 2-3.

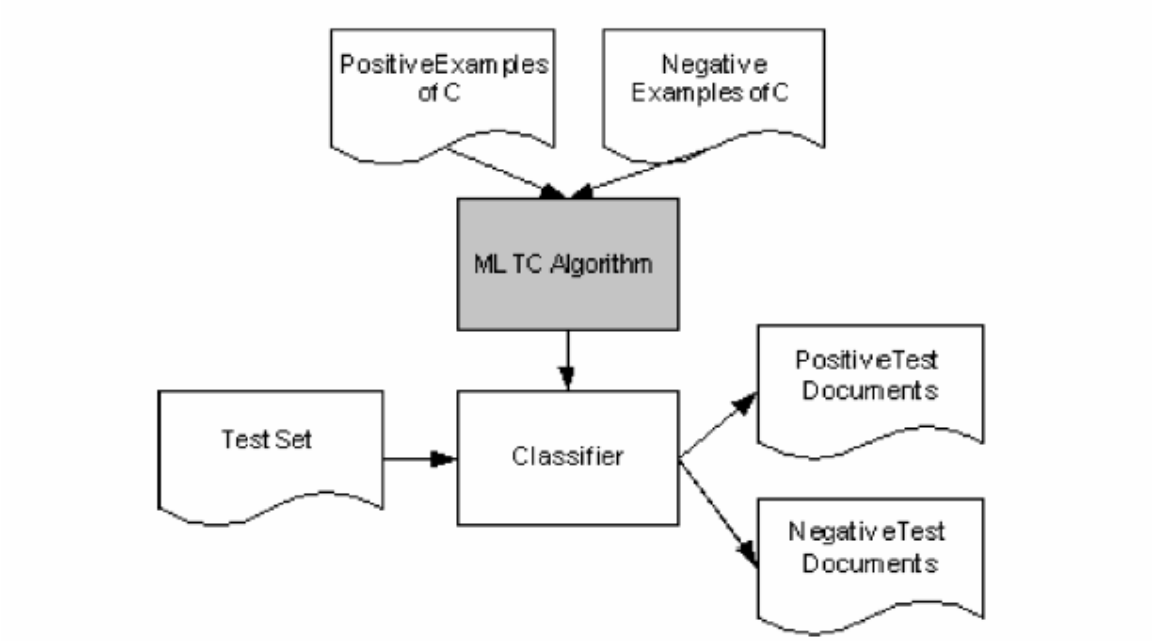


Figure 2-3 Diagram of a typical ML-based categorization system.

2.3.3 Support Vector Machine

One supervised machine learning approach, Support Vector Machine (SVM), has been widely used in text classification.

The classification process involves training and testing data which consists of some data instances. Each instance in the training set consists one "target value" (class label) and several "attributes" (features). SVM produces a model from the training set that predicts the target value of data instances in the testing set. SVM operates by finding a hyperplane in the space of possible inputs. This hyperplane will attempt to split the positive examples from the negative examples. The split will be chosen to have the largest distance from the hyperplane to the nearest of the positive and negative examples (Figure 2-4). The data vectors, which lie on the boundary of the hyperplane, are called support vectors. Intuitively, this makes the classification correct for testing data that is near, but not identical to the training data. SVM achieves the classification results by mapping non-linearly separable training vectors in input space to linearly separable higher dimensional feature space. The SVM finds a separating hyperplane with maximal margin in that higher dimensional space (Figure 2-5)ϕIn Figure 2-4, is a mapping function.

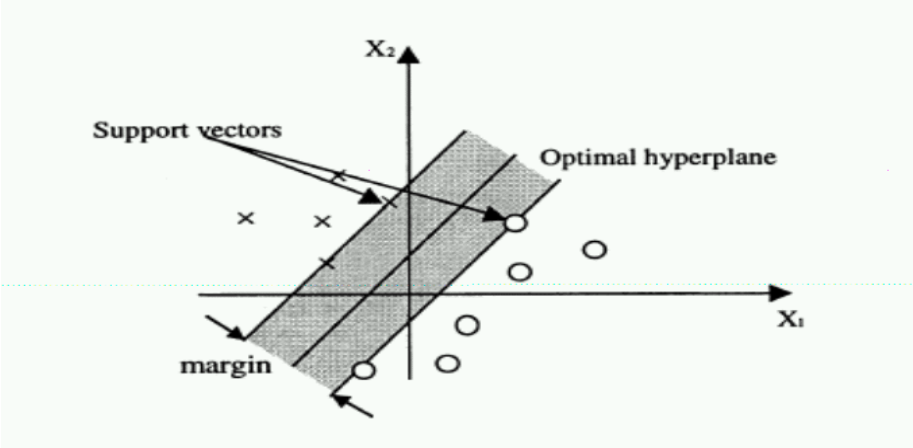


Figure 2-4 Hyperplane and support vectors (Burges, 1998)

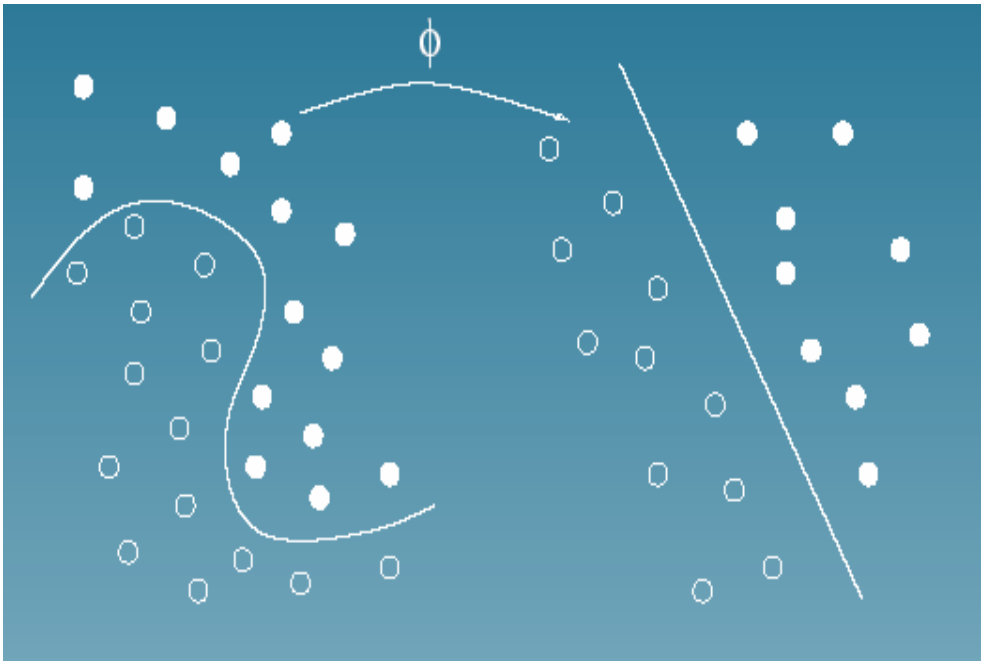


Figure 2-5 The mapping of non-linearly separable training vectors in input space to linearly separable higher dimensional feature space (Burges, 1998)

SVM is a kernel-based learning approach. The kernel-based methods define the class of possible patterns by introducing a notion of similarity between data. Kernel function is the way to define the similarity between data items. Some widely used kernels are:

Linear kernel:

$$K(x_i, x_j) = x_i^T x_j. \tag{2-4}$$

Polynomial kernel:

$$K(x_i, x_j) = (\gamma {x_i}^T x_j + r)^d, \gamma > 0. \tag{2-5}$$

Radial basis function

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0. \tag{2-6}$$

Sigmoid:

$$K(x_i, x_j) = \tanh(\gamma {x_i}^T x_j + r). \tag{2-7}$$

Here, γ, r, and d are kernel parameters.

In this thesis, we will use SVM for incorporation of Human Genome Epidemiology (HuGE) relevant articles from PubMed database into the Center for Disease Control and Prevention's (CDC) Human Genome Epidemiology Network, or HuGENet™ (http://www.cdc.gov/genomics/hugenet/) published literature database (Chapter 6).

2.4 Mining text for association rules

The goal of the association rule techniques is to detect relationships or associations between specific values of categorical variables in large data sets. This is a common task in data mining as well as in text mining. These powerful exploratory techniques have a wide range of applications in many areas of business practice and also research - from the analysis of consumer preferences or human resource management, to the history of language. These techniques enable analysts and researchers to uncover hidden patterns in large data sets, such as "customers who order product *A* often also order product *B* or *C*" or "employees who said positive things about initiative *X* also frequently complain about issue *Y* but are happy with issue *Z*." The implementation of the so-called a-priori algorithm (Agrawal and Swami, 1993; Agrawal and Srikant, 1994; Han and Lakshmanan, 2001) allows you to process rapidly huge data sets for such associations, based on predefined "threshold" values for detection.

2.4.1 How association rules work.

The usefulness of this technique to address unique data mining problems is best illustrated by a simple example. Suppose you are collecting data at the check-out cash registers at a large book store. Each customer transaction is logged in a database, and consists of the titles of the books purchased by the respective customer, perhaps additional magazine titles and other gift items that were purchased, and so on. Hence, each record in the database will represent one customer (transaction), and may consist of a single book purchased by that customer, or it may consist of many (perhaps hundreds of) different items that were purchased, arranged in an arbitrary order depending on the order in which the different items (books, magazines, and so on) came down the

conveyor belt at the cash register. The purpose of the analysis is to find associations between the items that were purchased, i.e., to derive association rules that identify the items and co-occurrences of different items that appear with the greatest (co-)frequencies. For example, you want to learn which books are likely to be purchased by a customer who you know already purchased (or is about to purchase) a particular book. This type of information could then quickly be used to suggest to the customer those additional titles. You may already be "familiar" with the results of these types of analyses, if you are a customer of various on-line (Web-based) retail businesses; many times when making a purchase on-line, the vendor will suggest similar items (to the ones purchased by you) at the time of "check-out", based on some rules such as "customers who buy book title *A* are also likely to purchase book title *B*," and so on.

2.4.2 Mining text for association

Experiments of association extraction have been carried out by Feldman et al. (1996), and Rajman and Besancon (1997) with the KDT (Knowledge Discovery in Texts) system on the Reuter corpus. The Reuter corpus is a set of 22,173 documents that appeared on the Reuter newswire in 1987. The documents were assembled and manually indexed by Reuters Ltd. and Carnegie Group Inc. in 1987. The documents were indexed with 135 categories in the Economics domain. The mining was performed on the indexed documents only.

All known algorithms for generating association rules operate in two phases. Give a set of keywords $A = \{w_1, w_2, \ldots, w_m\}$ and a collection of indexed documents $T = \{t_1, t_2, \ldots, t_n\}$, the extraction of associations satisfying given support and confidence constraints σ and γ is performed:

- by first generating all the keyword sets with support at least equal to σ. The generated keyword sets are called the frequent sets (or σ-covers);

- then by generating all the association rules that can be derived from the produced frequent sets and that satisfy the confidence constraint γ.

Generating the frequent sets: The set of candidate σ-covers (frequent sets) is built incrementally, by starting from singleton σ-covers and progressively adding elements to a σ-cover as long as it satisfies the confidence constraint. The frequent set generation is the most computationally expensive step (exponential in the worse case). Heuristic and incremental approaches are currently investigated. A basic algorithm fro generating frequent sets is indicated in Figure 2-6.

$i = 1, Cand_i = \{\{w\}, |[\{w\}]| \geq \sigma\}$, *where w are key-words*;
while $(Cand_i \neq \emptyset)$ **do**
  $Cand_{i+1} = \{S_1 \cup S_2 |\ S_1, S_2 \in Cand_i,$
    and $|S_1 \cup S_2| = i + 1$
    and $\forall S \subseteq S_1 \cup S_2, (|S_1 \cup S_2| = i) \Rightarrow (S \in Cand_i)$
    and $|[S_1 \cup S_2]| \geq \sigma\}$
  $i = i + 1$;
endw

Figure 2-6 Generating the frequent sets

Generating the association. Once the maximal frequent sets have been produced, the generation of the association is quite easy. A basic algorithm is presented in Figure 2-7.

foreach W *maximal frequent set* **do**
  generate all the rules $W \backslash \{w\} \Rightarrow \{w\}$, where $w \in W$, such that $\frac{|[W \backslash \{w\}]|}{|[W]|} \geq \sigma$;
endfch

Figure 2-7 Generating the associations

To apply association rule to biomedical literature, we will start off by treating the keywords for each gene as "items". We will then apply rules in association rule mining in text to discover rules where the l.h.s. of the rule may be a gene name and the r.h.s. will be a list of functional keywords with support at least equal to the support σ. The association rules can be derived from the produced frequent sets that satisfy the confidence constraint γ. The end result will be to generate a list of "highly associated" keywords for each gene, without bringing in the Z-score or TFIDF based ranking scheme. This will constitute an alternative way of constructing functional keyword lists for genes based on the parameters σ and γ. Then by linking the keywords across the genes, we may potentially find the "new" relationship between genes.

2.5 Summary

In this chapter, we survey the main data clustering techniques that have been applied to microarray data analysis. We point out the limitations of these available approaches. Our aim is to help scientists switch from some random, un-guided search to a more guided, intelligent search by applying machine learning and statistical techniques to improve the relative effectiveness of the search. Furthermore, automated biomedical literature classification using supervised machine learning approaches, such as support vector machines, can assist researchers to quickly separate the articles they are interested in from the huge literature databases. In the next chapter, we will discuss some issues related to biomedical text analysis.

CHAPTER 3

ISSUES FOR ANALYSIS OF BIOMEDICAL TEXT

One of the rich resources of on-line information is the scientific literature. The MEDLINE database, for example, provides bibliographic information and abstracts for more than 12 million articles that have been published in biomedical journals (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed). However, all the information contained in the database is stored as text. The rapid growth of these collections makes it increasingly difficult for humans to access the required information in a convenient and effective manner (Andrade and Bork, 2000). Clearly, there is a necessity of developing methods for automatic extraction of relevant information (such as keywords associated with genes) from the literature, which is written in natural language.

A number of groups are developing algorithms that link information from medical literature with gene names. Andrade and Bork (2000) developed a program that links the OMIM database of human inherited diseases to keywords derived from MEDLINE with their statistical profiling algorithm. A variety of nonstatistical approaches have also been used to organize genes. The web tool, PubGene, finds links between pairs of genes based on their co-occurrence in MEDLINE abstracts (Jenssen et al., 2001). Another approach (Masys et al., 2001), the basis of the HAPI web tool, organizes gene names according to predefined hierarchical classification systems of enzymes and diseases, and includes hyperlinks to specific MEDLINE citations responsible for the individual classifications. Still another approach (Tanabe et al., 1999), used by the MedMiner system, automatically

retrieves functional information (both keywords and gene names related to a user-defined function) from the GeneCards database, and configures it for a PubMed search. The algorithm presents the results by the specific sentence containing the information rather than by the title, speeding review of the results if the user prefers to extract relevant sentences rather than scan through the whole abstract text

Keyword extraction is an important step to link genes with biomedical literature. Ideally, high quality keyword lists for gene identification should be able to distinguish certain individual genes from others. Various weighting schemes have been developed to determine the importance of a word to a document. Andrade and Valencia (1998) introduced a statistical profiling approach (the "z-score" method), which accepts user-supplied abstracts related to a protein of interest and returns an ordered set of keywords that occur in those abstracts more often than would be expected by chance, and it has been used by Blaschke et al. (2001) and us (Chapter 3 and Chapter 4). Term frequency–inverse document frequency (TFIDF) (Salton and Buckley, 1988), one of the most widely used weighting schemes in the information retrieval research area, has also been applied to analyze biomedical literature to identify functionally coherent gene groups (Raychaudhuri et al., 2002). Term frequency (TF) is used as part of TFIDF weighting scheme to measure the frequency of occurrence of the words in a document. In our case, the collection of abstracts related to a single gene is a "document". However, term frequency factors alone cannot ensure high quality keyword lists. Specifically, when the high frequency words are not concentrated in a few particular genes, but instead are prevalent in all the genes, the keyword lists cannot be used to identify the specific genes. Inverse Document Frequency (IDF) is introduced as a multiplier to favor words

concentrated in a few genes rather than all the genes. IDF varies inversely with the number of genes n with which a word is associated in a collection of N genes.

In this chapter, we first expand, extend, and optimize the z-score method by testing new background sets, a new stemming algorithm, and a new, extensive stop list customized for use with the biological literature. This extended method was used to create a repository of functional keywords from MEDLINE abstracts for genes. We also compare our results with information found in public databases. We then compare the performance of the z-score method with TFIDF for the purpose of extracting the functional keywords for each tested gene set by evaluating the precision and recall values.

3.1 Creating a Relational Database of Medline abstracts

The entire PubMed from 1965-2000 is obtained from the National Library of Medicine (http://www.nlm.nih.gov). The original abstracts are in XML format. These abstracts are processed and stored in an ORACLE database in order to use SQL to query the database.

There are four tables in the databases. The create table statements are:

```
create table JOURNAL(
Journal_ID      NUMBER(10,0) primary key,
ISSN            VARCHAR2(25),
Volume          VARCHAR2(25),
Issue           VARCHAR2(25),
Year            NUMBER(4,0),
Month           VARCHAR2(5)
);

create table AUTHOR(
Author_ID       NUMBER(10,0) primary key,
Last_Name       VARCHAR2(30),
First_Name      VARCHAR2(25),
Middle_Name     VARCHAR2(25)
);

create table CITATIONS(
Medline_ID      NUMBER(10,0) primary key,
PMID            NUMBER(12,0),
Title           VARCHAR2(500),
Journal_ID      NUMBER(10,0),
Abstract        VARCHAR2(4000),
foreign key (Journal_ID) references JOURNAL(Journal_ID)
);


create table CIT_AUTHORS(
Medline_ID      NUMBER(10,0) NOT NULL,
Author_ID       NUMBER(10,0)  NOT NULL,
constraint CIT_AUTHORS_PK PRIMARY KEY (Medline_ID, Author_ID),
foreign key (Medline_ID) references CITATIONS(Medline_ID),
foreign key (Author_ID) references AUTHOR(Author_ID)
);
```

Using these four tables, a parser is developed to parse XML abstracts into appropriate tokens which represent values for individual attributes in the above tables. All subsequent analysis has been performed against this populated database of over 5.5 million abstracts.

3.2 Keyword extraction from biomedical literature

We use the z-score method and TFIDF to extract keywords from MEDLINE. These methods estimate the significance of words by comparing the frequency of words in a test (gene-related) set of abstracts with their frequency in a background set of abstracts.

3.2.1 Background Sets

The first step is building the dictionary of all the words used in the abstracts based on the background distribution of the same words in the documents/families/pseudo-families. The goal is to identify keywords that "stand out" in comparison to their average occurrence in the background set of documents. The background sets of documents are used to build a hash table of words and their respective statistics for comparison with the corresponding words in the test sets.

The background set used by Andrade and Valencia (1998) consists of abstracts associated with 71 protein families in the 1993 release of the PDBSELECT database. By the year 2000 this database had grown to 1155 protein families, 760 of which have >4 members. We use abstracts associated with the PDB-1155 and PDB-760 protein families, which have an average of 41 and 57 abstracts per family, respectively. A third background set is created consisting of 50,000 randomly selected MEDLINE abstracts sorted into 1000 pseudo-families of 50 abstracts each. Finally, we build a large random background set (approximately 112,000 pseudo-families of 50 abstracts each), which incorporates abstracts in the entire MEDLINE collection up to year 2000.

3.2.2 Test Sets

For each gene analyzed, word frequencies are calculated from a group of MEDLINE abstracts retrieved by an SQL search against the relational database version of Medline, in the TITLE field, for the specific gene name or any known aliases. The resulting set of abstracts iss processed to generate a specific keyword list.

We use three test sets in our comparisons.

3.2.2.1 The first group of genes (test set #1) are calcyclin (C), cathepsin H (H), cathepsin S (S), glutamine-oxaloacetate transaminase (GOT), nexin-1 (N), osteopontin (OPN), and uridine kinase (UK).

3.2.2.2 To test if our system can extract new information from the medical literature, we design a second query set for OPN using abstracts only from the year 2001 (test set #2), with the hope of extracting relevant keywords for several novel functional links between OPN and diseases, such as hypertension (Hartner et al., 2001), autoimmune demyelinating diseases (Chabas et al., 2001) and tumor metastasis (Furger et al., 2001).

3.2.2.3 The third group of genes is used to evaluate the keyword identification algorithms by precision-recall and error-minimization tests as described below. We evaluate the accuracy of the keyword-selection algorithms by comparing their output with the set of keywords selected by three knowledgeable investigators from the same set of abstracts. For each of 10 genes with diverse biological functions (adenylate cyclase, androgen receptor, calmodulin, caspase-3, dopamine D2 receptor, GluR2 AMPA receptor subunit, glutamic acid decarboxylase-65, histone H4, L-type calcium channel, and tyrosine hydroxylase), we retrieve a set of abstracts by a simple search for the gene name in the citation TITLE field (limited to the 10 most recent citations for each set). These 10

sets of 10 abstracts each are processed for keyword selection by the two weighting schemes. These abstracts are also hand-processed by three medical researchers (Karen Borges, Brian J. Ciliax, and Ray Dingledine), who select keywords that are reflective of the biological functions described in each abstract.

3.2.3 Stemming

Word stemming is used to truncate suffixes and trailing numerals so that words having the same root (e.g., activate, activates, activation, and active, all have the same root of "activ") are collapsed to the same word for frequency counting. Two stemming algorithms are compared, one used by Andrade and Valencia (1998), and one devised by Porter (1980). A third condition, in which the words are not stemmed, is used as a control.

3.2.4 Stop-word Lists

Stop-word lists are typically used to filter out non-scientific English words that carry low domain-specific information content. We test two stop-word lists initially: a simple list of 319 common English words (http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words), and an online dictionary of 22,205 words (http://ftp.std.com/obi/Dictionary/dict). Our initial tests lead us to add methodological words that are unrelated to gene or protein function to the online dictionary, and to remove selected words. This results in a stop-word list customized for biological applications. This stop-list, abbreviated PD+ (pocket dictionary plus), is evolving as we delete more biological or functional words and add methodology words. We also analyze keywords without applying a stop-word list, which served as the control condition.

3.3 Keyword Assessment

3.3.1 Z-score method

Statistical formulae from Andrade and Valencia (1998) for word frequencies and z-scores are used without modification. The weight of word a for gene g is represented by the z-score, and is defined as

$$Z_g^a = \frac{F_g^a - \overline{F^a}}{\sigma^a} \tag{3-1}$$

where F_g^a equals the document frequency of word a in test set (gene) g and, as defined by Andrade and Valencia (1998), $\overline{F}^a$ and σ^a are the average frequency (frequency per document) and standard deviation, respectively, of word a in the background set. For the random background set, the document frequencies of word a across pseudo-families of 50 randomly-selected abstracts each are used to calculate these latter metrics instead of the proportions of proteins in individual families for which word *a* appears in at least one representative abstract (Andrade and Valencia, 1998). In other words, the original work of Andrade and Valencia (1998) treats abstracts related to a protein family as one document – they had 955 families. In our random background set, 50 abstracts make up a pseudo-family and are treated as one document for computing z-score values.

3.3.2 TFIDF method

The standard TFIDF function is used (Salton and Buckley, 1988). TFIDF combines term frequency (TF), which measures the number of times a word occurs in the gene's set of abstracts (reflecting the importance of the word to the gene), and inverse document

frequency (IDF), which measures the information content of a word – its rarity across all the documents/families in the background set. The inverse document frequency (IDF) is calculated as:

$$idf^a = \log \frac{N}{df^a} \tag{3-2}$$

where idf^a denotes the inverse document frequency of word a in the background set; df^a denotes the number of families (or pseudo-families) in which word a occurs; and N is the total number of documents/families/ pseudo-families in the background set.

TFIDF is defined as:

$$tfidf_g^a = tf_g^a \times idf^a \tag{3-3}$$

$tfidf_g^a$ denotes the weight of the word a to the gene g; tf_g^a the number of times word a occurs in gene g.

To distribute the word weights over the [0, 1] interval, the weights resulting from TFIDF are often normalized by cosine normalization, given by

$$weight_g^a = \frac{tfidf_g^a}{\sqrt{\sum_{s=1}^{|W|} (tfidf_g^s)^2}} \tag{3-4}$$

where |W| denotes the number of words in the abstracts of gene g.

3.3.3 Normalized z-score method

In order to compare with TFIDF, the z-scores of the words are also normalized (normalized z-score method) as:

$$weight_g^a = \frac{Z_g^a}{\sqrt{\sum_{s=1}^{|W|} (Z_g^s)^2}} \tag{3-5}$$

The weight of a word is assigned the value "New", if the word occurs in the test set but not in the background set, since no background statistics are available from which to calculate the z-score or tfidf values.

3.4 Precision-Recall and Error-Minimization

Using the keyword lists generated from the first test set, investigator-derived lists are used as the standard against which the algorithm-derived lists are evaluated by Precision and Recall measurements. Precision (P) and Recall (R) are the standard metrics for retrieval effectiveness in information retrieval. They are calculated as follows:

$$\mathbf{P = tp / (tp + fp)}$$

$$\mathbf{R = tp / (tp + fn),}$$

Where:

tp = words in the algorithm-derived list also found in the investigator-derived list;

fp = words in the algorithm-derived list not found in the investigator-derived list;

fn = words in the investigator-derived list not found in the algorithm-derived list.

The optimum combination of the parameters (different background sets,

stemming algorithms, and stop lists) plus the z-score threshold for accepting a word is determined by minimizing the joint error minimization function: $E= V * (1 – P) + (1 – R)$ (Hvidsten et al., 2001), which combines the role of precision and recall simultaneously. If $V > 1$, the cost of false positives is weighted more heavily than the cost of false negatives. We select $V = 4$ empirically to limit the number of irrelevant words when classifying gene function.

3.5 Keyword lists

An example of a keyword list (test set #1) for the gene name "OPN" is shown in Table 3-1 (Only the top 100 words were shown). To find out the relevance of the keywords for a gene, one expert (Brian J. Ciliax) inspected a word list for osteopontin (OPN, test Set #1) to select keywords with z-scores above 2.0 and filter out general or methodological words (essentially all non-functional words related to methodology, e.g. cDNA, polyclonal, chromatography, escherichia, coli, histology, lysates, Sepharose, clone, biotinylated, recombinant, nmr, hybridization, densitometric, luciferase, polyacrylamide, immunogold, immunostaining, immunohistochemistry). The relevance of keywords for OPN function is determined by searching the query set of abstracts for their occurrence and reading the abstracts. Virtually every keyword is found to have at least one highly relevant meaning in the context of the OPN literature (Table 3-2).

Table 3-1. Keyword list for gene Osteopontin

Word	z-score	Word	z-score
integrin	**48.3**	polyelectrolytes	8.7
transwell	22.5	**postovulatory**	**8.7**
ctgf	**19.5**	reaggregate	8.7
fluorimetry	19.5	**glomeruli**	**8.6**
histotroph	**19.5**	upregulation	8.5
tcf	**19.5**	**normoxia**	**8.3**
trophectoderm	**19.5**	**proteinaceous**	**8.1**
vsm	19.5	**metastasis**	**8.0**
vibrissa	**15.9**	**lithogenic**	**7.9**
adpkd	**13.8**	**losartan**	**7.9**
atn	**13.8**	ranets	7.9
catagen	**13.8**	**uremia**	**7.9**
chitosan	**13.8**	**hyperglycemic**	**7.7**
diphenylene	13.8	**mapk**	**7.6**
enterokinase	**13.8**	**kappab**	**7.6**
iodonium	**13.8**	**normoxic**	**7.6**
ishikawa	13.8	**antineutrophil**	**7.3**
mdh	**13.8**	**crevicular**	**7.3**
nexin	**13.8**	**doca**	**7.3**
nondialysis	13.8	**mek**	**7.3**
ptf	**13.8**	**neurotomy**	**7.3**
pulpitis	**13.8**	**periglomerular**	**7.3**
renoprotective	**13.8**	**mcp**	**7.3**
vth	**13.8**	**fibrotic**	**7.2**
interstitium	**13.2**	autoregulatory	6.8
lsab	11.2	**gcf**	**6.8**
matrigel	11.2	**gonadotropes**	**6.8**
postovulation	**11.2**	**jnk**	**6.8**
pulmonal	**11.2**	**muc**	**6.8**
telogen	**11.2**	spatio	6.8
upar	**11.2**	**hgf**	**6.8**
upa	**10.8**	**migratory**	**6.7**
stone	**10.6**	**thrombin**	**6.7**
atheromatous	**10.4**	**chemokine**	**6.5**
urolithiasis	**10.4**	antagonises	6.4
erk	**10.3**	deoxynucleotidyl	6.4
tartrate	10.3	**hypercholesterolaemia**	**6.4**
hoxa	**9.7**	**hyperphosphatemia**	**6.4**
igan	**9.7**	**oro**	**6.4**
lucigenin	9.7	**kidneys**	**6.4**
lymphoproliferation	**9.7**	ethylene	6.3
nephritic	**9.7**	**autoimmunity**	**6.2**
osteoclastogenesis	**9.7**	**henle**	**6.1**
talin	**9.7**	morphogenic	6.1
tympanosclerosis	**9.7**	propidium	6.1
mesangial	**9.2**	**smad**	**6.1**
mmp	**8.9**	**spongiosa**	**6.1**

Table 3-2. OPN facts extracted by manual inspection of 100 MEDLINE abstracts

1. 2ar) is described whose abundance is greatly increased by the tumor promoter 12-O-tetradecanoylphorbol 13-acetate both in JB6 epidermal cells in vitro and in epidermis in vivo.
2. 2ar, a tumor promoter-inducible protein secreted by mouse JB6 epidermal cells, is the murine homolog of rat osteopontin,
3. several clonal lines of preneoplastic JB6 cells derived from Balb/c mouse epidermal cultures upon treatment with 12-O-tetradecanoyl phorbol-13-acetate (TPA), become irreversibly oncogenic and concomitantly synthesize OPN at elevated levels
4. TGF beta promotes the production of osteopontin in the osteoblastic osteosarcoma cells
5. induction of 2ar in epidermal or fibroblast cell lines by tumor promoters, growth factors, and transformation with H-ras
6. Osteopontin mRNA is regulated by the osteotropic hormones dexamethasone and 1,25(OH)2D3.
7. hPTH(1-34) suppresses the production of the novel extracellular matrix protein, OP, in osteoblasts
8. stretch-induced upregulation of osteopontin mRNA expression is mediated, in part, via production of ANG II
9. Studies with several fibroblast and epithelial-derived cell lines in culture indicate that secretion of osteopontin can be dramatically increased when these cells are treated with phorbol esters, growth factors and hormones. However, osteopontin does not appear to be expressed by mesenchymal cells, fibroblasts, epidermal cells or by most epithelial cells in vivo.
10. The expression level of osteopontin (OPN) mRNA was found to be increased in a macrophage cell line in the presence of recombinant tumor necrosis factor-alpha (TNF-alpha)
11. the hormonal form of vitamin D regulates the biosynthesis of osteopontin
12. a potential calcium binding loop and two potential heparin binding sites
13. A thrombin-cleaved NH(2)-terminal fragment of osteopontin containing the RGD sequence has recently been shown to also be a ligand for alpha(9)beta(1).
14. among the alphav integrins, only the alphavbeta3 is able to support cellular adhesion to osteopontin.
15. The results show that the Arg-Gly-Asp sequence also confers cell-binding properties on bone-specific sialoprotein.
16. Elevated expression of osteopontin (OPN), a secreted adhesive phosphoglycoprotein, is frequently associated with many transformed cell lines of epithelial and stromal origin. Moreover,
17. oncogenically transformed tsB77 cells may exploit the lack of OPN-receptor interactions for their invasive behavior
18. OPN and alphavbeta3 integrin, were also predominantly observed in the microvasculature of glioblastomas associated with VEGF expression.
19. OPN has been associated with malignant transformation as well as being ligand to the CD44 receptor.
only the 69-kDa OPN, not its 62-kDa form, undergoes receptor-

Table 3-2 (continued)

mediated localization on the cell surface, although tsB77 cells synthesize OPN receptors (alpha(v)beta3 integrins) at both permissive and nonpermissive temperatures.
20. OPN stimulates pp60c-src kinase activity associated with the alpha v beta 3 integrin and that the association requires the cytoplasmic tail of the alpha v chain
21. osteopontin (OP), a matrix protein that mainly interacts with the alphav integrin family, increased time-dependently by TNF-alpha stimulation at gene and protein levels
22. Osteopontin (OPN) is a negatively charged, highly acidic glycosylated phosphoprotein that contains an GRGDS amino acid sequence, characteristic of proteins that bind to integrin receptors
23. Osteopontin (OPN) is a secreted glycoprotein with mineral- and cell-binding properties that can regulate cell activities through integrin receptors.
24. Osteopontin (OPN) is a soluble secreted phosphoprotein that binds with high affinity to several different integrins.
25. Osteopontin (OPN) is an acidic 70-kDa glycoprotein that is cleaved by proteases to yield 45-kDa and 24-kDa fragments. The 70-kDa and 45-kDa proteins contain a Gly-Arg-Gly-Asp-Ser (GRGDS) sequence that binds to cell surface integrins (primarily alpha(v)beta(3) heterodimer) to promote cell-cell attachment and cell spreading.
26. phosphorylated and nonphosphorylated forms of osteopontin have different physiological properties, which may regulate the functional roles of this extracellular matrix protein
27. Bone sialoprotein (BSP) and osteopontin (OPN) are secreted glycoproteins with a conserved Arg-Gly-Asp (RGD) integrin-binding motif and are expressed predominantly in bone.
28. Osteopontin is a predominant integrin binding protein of bone and its expression has been shown to be induced by mechanical stimuli within osteoblasts
29. the fragmentation of SPPI is important in bone formation and remodeling
30. Rat bone cells in culture produce several forms of SPPI that differ in post-translational modifications such as phosphorylation and sulphation.
31. Secreted OPN is then available as ligand for alpha(v)beta(3) integrin heterodimer on trophectoderm and uterus to 1) stimulate changes in morphology of conceptus trophectoderm and 2) induce adhesion between luminal epithelium and trophectoderm essential for implantation and placentation.
32. The expression of BSP and OPN in tumor cells provides a selective advantage for survival via initial binding to alpha(V)beta(3) integrin (both) or CD44 (OPN) on the cell surface, followed by sequestration of Factor H to the cell surface and inhibition of complement-mediated cell lysis.
33. the 44-kDa bone phosphoprotein (44K BPP, also called sialoprotein I or osteopontin)
34. The cDNA sequence indicated the presence of a Gly-Arg-Gly-Asp-Ser-(GRGDS) amino acid sequence identical to a cell binding sequence in fibronectin,
the difference between the 69-kDa and 62-kDa isoforms of OPN resides in

Table 3-2 (continued)

their sialic acid content, and sialylation of OPN is crucial for its receptor-mediated binding on tsB77 cells
35. The extracellular matrix protein osteopontin (OPN) interacts with a number of integrins, namely alphavbeta1, alphavbeta3, alphavbeta5, alpha9beta1, alpha8beta1, and alpha4beta1.
36. the integrin attachment sequence (RDG)
37. the linear sequence SVVYGLR directly binds to alpha(9)beta(1) and is responsible for alpha(9)beta(1)-mediated cell adhesion to the NH(2)-terminal fragment of osteopontin.
38. thrombin cleavage regulates the adhesive properties of OPN and that alpha5beta1 integrin can interact with thrombin-cleaved osteopontin when in a high activation state.
39. vitronectin receptor, which has known specificity for osteopontin,
40. attachment is inhibited by RGD-containing peptides.
41. Sppl amino acid sequence contains the GRGDS cell-binding sequence which is known to be important for cell attachment to several adhesive proteins found in extracellular matrices.
42. Because of the presence of the GRGDS cell-binding sequence in Sppl, it is probable that abnormally high expression of this soluble protein by tumor cells has important consequences for interactions between tumor cells and the host tissue matrix.
43. osteoclasts when resorbing bone are anchored by osteopontin bound both to the mineral of bone matrix and to a vitronectin receptor on the osteoclast plasma membrane.
44. Osteopontin (OP) is a recently discovered bone matrix protein which was shown to promote the attachment of osteoblastic rat osteosarcoma ROS 17/2.8 cells to their substrate.
45. Osteopontin is a macrophage adhesive protein that is expressed by renal tubules in tubulointerstitial disease.
46. the hippocampus and the striatum following global forebrain ischemia upregulate OPN mRNA
47. The transient induction of OPN mRNA after global ischemia occurred earlier in the striatum than in the hippocampus. It was pronounced in the dorsomedial striatum close to the lateral ventricle and in the CA1 subfield and the subiculum of the hippocampus before microglial cells became more reactive.
48. accumulation of the non-collagenous matrix bone-associated proteins, osteopontin, osteocalcin, and osteonectin, has been demonstrated in atheromatous plaques.
49. calcification is associated with increased expression of osteopontin by smooth muscle cells
50. circle development of diabetic atherosclerosis associated with arterial wall hypoxia
51. osteopontin (OPN) has been shown to participate in the pathological calcification
52. osteopontin has recently been implicated in the development of atherosclerosis
53. Osteopontin is a good marker for the injury-induced SMC phenotypic state in vivo and in vitro.
54. Bone sialoprotein (BSP) and osteopontin (OPN, ETA-1) are expressed by trophoblasts and are strongly up-regulated by many tumors.
55. enhanced secretion of 2ar/pp69/osteopontin by transformation of a wide variety of mammalian fibroblasts and epithelial cells is often correlated with tumorigenicity.

Table 3-2 (continued)

56. Increased levels of OPN exist in blood from the lungs, breasts, and gastrointestinal tracts of cancer patients with metastases.
57. oncogenically transformed cells secrete different molecular forms of osteopontin (OPN), a sialic acid-rich, adhesive, phosphoglycoprotein, than OPNs secreted by their nontransformed counterparts
58. OPN are non-collageneous bone matrix proteins expressed by some epithelial tumor cells in exceptional cases.
59. Osteopontin (OPN) has been associated with enhanced malignancy in breast cancer
60. Osteopontin (OPN) is a secreted, adhesive protein that is highly expressed in JB6 cells with TPA treatment, and its expression persists for at least 4 days, which is the time required for subsequent expression of transformed phenotype.
61. osteopontin is identical to a transformation- associated phosphoprotein whose level of expression by cultured cells and abundance in human sera has been correlated with tumorigenicity
62. OPN was involved in the stromal formation of myxoid or hyaline tissues in pleomorphic adenomas. In summary, pleomorphic adenomas expressed the bone matrix proteins OSN and OPN.
63. suggest a role for osteopontin in carcinogenesis.
64. OPN exists as an integral component of a hyaluronan-CD44-ERM attachment complex that is involved in the migration of embryonic fibroblasts, activated macrophages, and metastatic cells.
65. Osteopontin (OPN) induces endothelial cell migration and upregulates endothelial cell migration induced by VEGF.
66. Osteopontin induces cellular chemotaxis but not homotypic aggregation
67. the chemotactic activity of osteopontin (OPN) on the precursor of osteoclasts.
68. osteopontin is involved in the accumulation of macrophages within the peritubular and periglomerular interstitium in the obstructed renal cortex
69. expression of OPN was identified in the retina and OPN-like immunoreactivity was present in a number of ganglion cells.
70. osteopontin gene is turned on relatively late in calvarial development
71. Alkaline phosphatase (AP), osteopontin (OP), and osteocalcin (OC) are expressed during osteoblastic differentiation
72. Osteopontin (OPN) is one of the major non-collagenous proteins in root cementum and other mineralized tissues.
73. synthesized by some odontoblasts and secreted into predentin, prior to the onset of mineralization.
74. Osteoclasts express the alphavbeta3 integrin, which is one of the receptors for osteopontin.
75. bone phosphoprotein (44K BPP, also called sialoprotein I or oestopontin
76. Bone sialoprotein (BSP) and osteopontin (OPN) are prominent, mineral-associated proteins in the extracellular matrix of bone that have been implicated in the metastatic activity of cancer cells.

Table 3-2 (continued)

OPN is an important factor triggering bone remodeling caused by mechanical stress
77. OPN, rather than HUA, is the major ligand for CD44 on bone cells in the remodelling phase of healing of fractures.
78. Osteopontin is one of the major noncollagenous bone matrix proteins produced by osteoblasts and osteoclasts, bone cells that are uniquely responsible for the remodeling of mineralized tissues.
79. Secreted phosphoprotein 1 (Spp-1; osteopontin) is one of the abundant noncollagenous proteins in bone matrix and is produced by osteoblasts.
80. synthesized by osteoblasts and osteocytes
81. the sulphation of SPPI is closely associated with mineralization and that it can be used as a sensitive and specific marker for the osteoblastic phenotype.
82. expression of myocardial osteopontin, an extracellular matrix protein, coincides with the development of heart failure and is inhibited by captopril, suggesting a role for angiotensin II
83. increased OPN expression plays an important role in regulating post-MI LV remodeling, at least in part, by promoting collagen synthesis and accumulation
84. Osteopontin (OPN), an extracellular matrix protein, is expressed in the myocardium with hypertrophy and failure.
85. These results suggest that p42/44 MAPK is a critical component of the ROS-sensitive signaling pathways activated by ANG II in CMEC and plays a key role in the regulation of osteopontin gene expression.
86. The cytokine osteopontin (Eta-1), was found to be a protein ligand of CD44.
87. Expression of the cytokine osteopontin (OPN) is elevated in granulomas caused by Mycobacterium tuberculosis.
88. OPN secreted by exudate macrophages might be an important regulator in the calcification of tympanosclerosis
89. Osteopontin (OPN) is a glycosylated phosphoprotein found in all body fluids and in the proteinaceous matrix of mineralized tissues.
90. osteopontin (OPN) is a protein involved in normal and pathological calcifications
91. osteopontin augments the host response against a mycobacterial infection
92. Osteopontin (OPN) is a sialic acid-rich, adhesive, extracellular matrix (ECM) protein with Arg-Gly-Asp cell-binding sequence that interacts with several integrins, including alpha(v)beta(3).
93. These findings identify Eta-1 as a key cytokine that sets the stage for efficient type-1 immune responses through differential regulation of macrophage IL-12 and IL-10 cytokine expression.
94.
95. a possible role in renal injury and regeneration

Table 3-2 (continued)

96. Macrophages present in the human glomerular crescent express osteopontin protein and mRNA at a high level.
97. OPN expressed by tubular epithelium played a pivotal role in mediating peritubular monocyte infiltration consequent to glomerular disease
98. OPN gene and protein expression is induced in both proximal and distal tubular cells during rat toxic acute renal failure.
99. OPN mediates early interstitial macrophage influx and interstitial fibrosis in unilateral ureteral obstruction.
100. osteopontin (OPN) and calprotectin (CPT) are present in the matrix of urinary calcium stones, and that OPN mRNA is expressed in the renal distal tubular cells.
101. Osteopontin (OPN) is a secreted phosphoprotein that is constitutively expressed in the normal kidney and is induced by various experimental and pathologic conditions.
102. Osteopontin (OPN) is one of the most important components in calcium stone matrix,
103. osteopontin expression in glomerular crescents in a rat model of anti-glomerular basement membrane glomerulonephritis.
104. Osteopontin mRNA is most abundant in bone but is also found in considerable amounts in kidney.
105. the 69-kDa major phosphoprotein, secreted by normal rat kidney (NRK) cells, is osteopontin,
106. The stones showed staining in two distinct zones: a core area was stained with randomly aggregated OPN and CPT, and peripheral layers were stained in concentric circles.
107. Urinary concentrations of OPN assessed using the enzyme-linked immunosorbant assay were significantly lower for stone-formers
108. A distinct co-localization of perimembranous OPN and cell-surface CD44 was observed in fetal fibroblasts, periodontal ligament cells, activated macrophages, and metastatic breast cancer cells.
109. mammary epithelial cells express OPN at elevated levels,
110. 2ar (OPN) codes for mouse osteopontin, an RGDS-containing, phosphorylated, sialic acid-rich Ca++-binding protein originally isolated from bone
111. an essential role of OPN in mammary gland differentiation and that the molecular mechanism(s) of its action, at least in part, involves down-regulation of MMP-2.
112. expression only in the brain stem with higher level in the pons and the medulla than in the midbrain.
113. mouse secreted phosphoprotein 1 (Spp-1, also known as 2ar, osteopontin, bone sialoprotein 1, 44-kDa bone phosphotein, tumor-secreted protein)
114. OPN mRNA was restricted to likely neurons in the olfactory bulb and the brain stem; it was not detected in the telencephalon and the diencephalon.
115. Osteopontin (OPN), a noncollagenous bone extracellular matrix, is a secreted adhesive glycoprotein with a functional RGD cell-binding domain that interacts with the alpha(v)beta3 cell surface integrin heterodimer.
116. This places Spp-1 on mouse chromosome 5
117. we identified an intracellular form of osteopontin with a perimembranous distribution in migrating fetal fibroblasts

3.6 Extraction of new information

The keyword list results using test set #2 showed that our system was able to identify keywords associated with newly discovered functions of OPN (Table 3-1). For example, our algorithms can identify the keywords and their associated z- scores captopril 2.3 (not shown), losartan 7.9, and atherosclerosis 4.4 (not shown) after the possible role of OPN in hypertension (Hartner et al., 2001). Similarly, a functional link between OPN and autoimmune demyelinating diseases (Chabas et al., 2001) is suggested by the keywords demyelinating 2.1 (not shown), encephalomyelitis 2.9 (not shown), autoantigen 2.6 (not shown), and autoimmune 6.2, whereas a link to tumor metastasis (Furger et al., 2001 is pointed to by the keywords catenin 5.3 (not shown), cadherin 3.3 (not shown), and tumorigenic 4.4 (not shown).

Besides MEDLINE (PubMed), there are several other resources which are available over the Internet that contain useful information regarding the specific functions of genes, for example, the Gene Ontology (GO) Consortium, SwissProt, GenBank, and GeneCards. These databases necessarily reduce the vast literature into a few functional concepts, whereas the algorithm-derived keywords often convey a much broader sense of the functions of genes. Using the osteopontin (OPN) gene as an example, we manually extracted all available functional information on OPN from these resources in January 2002. The extracted results are presented in Table 3-3. For OPN, the three GO keywords represent functional concepts, whereas the 19 words in GenBank are mostly aliases or biochemical descriptions for OPN. The GeneCards and SwissProt information are essentially the same and contain aliases, general characteristics and functional information. Taken together, a number of biological concepts regarding OPN are

represented in these various databases; however, individually, there are certain gaps in discrete topics for each database. For example, as of April 30, 2003, none of these databases identified the possible role of OPN in hypertension (Hartner et al., 2001), tumor metastasis (Furger et al., 2001), or in autoimmune demyelinating disease (Chabas et al., 2001). Finally, we searched Gene Ontology for the other gene names that we used to generate our preliminary data and found: 9 keywords for nexin, 0 keywords for cathepsin H and cathepsin S, 3 keywords for calcyclin**,** and no entries for glutamate-oxaloacetate transaminase or uridine monophosphate kinase. Therefore, we conclude that these popular public databases are useful, but individually and collectively incomplete when it comes to computing all relevant functional information about genes available publicly in Medline. This example indicates that our statistical algorithm can extract many relevant keywords, a number of which point to biological concepts not found in the existing public gene databases.

Table 3-3 – Information on OPN Extracted from Various Internet Gene Resources

Resource		Information
Gene Ontology:	cell adhesion cell adhesion molecule ossification	
GeneCards:	alternate/related names:	osteopontin precursor bone sialoprotein 1 urinary stone protein secreted phosphoprotein 1 SPP-1 nephropontin uropontin
	gene:	SPP1 or OPN
	composition:	314 amino acids
	molecular weight:	35 kD
	function:	binds tightly to hydroxyapatite; appears to form an integral part of the mineralized matrix; probably important to cell-matrix interaction.
	subunit:	ligand for integrin alpha-v/beta-3.
	alternative products:	3 isoforms are produced by alternative splicing: a/opn-a/op1b, b/opn-b/op1a, and c/opn-c.
	posttrans. modifications:	extensively phosphorylated on serine residues. N- and O-glycosylated.
	similarity:	belongs to the osteopontin family.
SwissProt:	[All of the above info from GeneCards is available in the human osteopontin entry for SwissProt, which also included the following fact:] Disease: this protein plays a principal role in urinary stone formation as the stone matrix.	

GenBank (Keywords field)

bone phosphoprotein; bone sialoprotein; calcium binding protein; cell adhesion phosphoprotein; extracellular matrix Protein; hydroxyapatite-binding protein; integrin-binding protein; matrix protein; mOP; osteopontin; phosphoprotein; secreted phosphoprotein; sialoprotein; sialoprotein I; SPP1 gene; SPPI protein; structural protein; tumor-associated phosphoprotein; hOP.

3.7 Optimization of the keyword selection algorithm.

The performance of the keyword-selection weighting schemes was evaluated initially by comparing their output with the set of keywords selected by human investigators from an identical set of 100 abstracts. The statistical algorithms used 1008 (3 * 3 * 4 * 28) combinations of three background sets: PDB-1155, PDB-760, and random families; three stemming rules: none, weak (Andrade and Valencia, 1998), and strong (Porter, 1980); four stop lists: none, simple stop list of 319 words, a 22,205 word online pocket dictionary (PD), and the supplemented pocket dictionary named PD+; and 28 z-score thresholds for accepting a keyword as being associated with the gene. A word was deemed to be associated with a gene by the algorithm only if the weight was above a user-set threshold. The investigator-derived lists were then used as the standard for evaluation of the algorithm-derived lists. For each combination of parameters we used the typical metrics of Precision (P) and Recall (R) to evaluate algorithm performance.

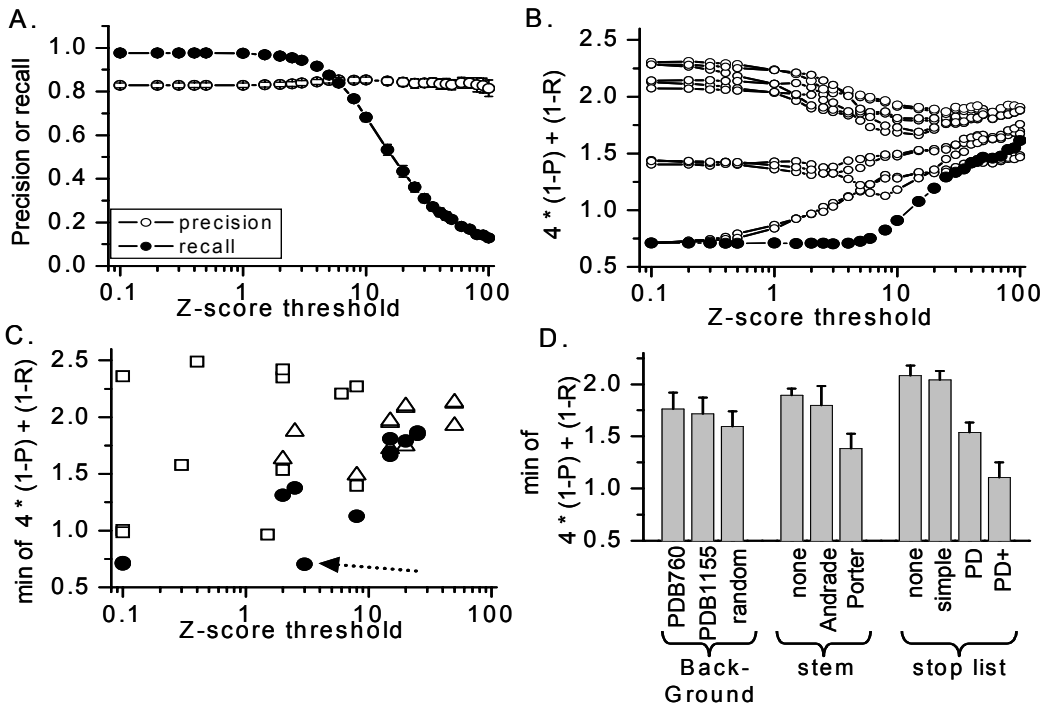


Figure 3-1. Evaluation and optimization of the keyword selection algorithm. A) Precision and recall as a function of the z-score threshold for accepting a word. B) The optimization function is plotted for each parameter set that includes the Porter strong stemmer. Solid circles represent data from the random background set and the PD+ stop list. C) The minima of the optimization function were determined from plots in B) and are plotted against the corresponding z-score for all parameter combinations. Solid circles = Porter stemmer, open boxes = the stemmer described in Andrade and Valencia (1998), and open triangles = no stemming. The arrow points to the optimum combination of parameters, which involve a z-score threshold > 3 and the combination of Porter stemmer, random background set and PD+ stop list. D) The sensitivity of the algorithm performance to changes in each parameter was systematically evaluated by calculating the mean ($\pm$SEM) of all optimization function minima in a data set, holding each parameter constant in turn. Performance was most affected by the stop list.

For the case in which 1000 families in the random background list were stemmed by the Porter algorithm and filtered by the PD+ stop list, as word selection became more stringent (increasing z-scores), recall fell but precision was nearly unaffected (Figure 3-1A). Examination of all P-R plots indicated that the extensive stop list was primarily responsible for the relatively flat precision because less extensive stop lists caused low precision at low z-score values. Figure 2B plots the error minimization function with V=4 for all 12 parameter sets that included the Porter strong stemming algorithm, and Figure 2C plots the minimum of this function against the z-score threshold for each parameter combination. Overall the best performance was achieved with the random background set, Porter strong stemming, the PD+ stop list and a z-score acceptance threshold of 3-8 for V ranging between 2 and 4.

Examination of Figure 3-1C shows that the stronger stemming algorithm (Porter, solid circles) often outperformed the weaker stemming algorithm (open squares) or no stemming (open triangles). To determine which parameter (background set, stemming algorithm or stop list) exerts the most influence on the performance of the algorithm, we calculated the mean value of the optimization function with each parameter being fixed in turn. Figure 3-1D shows that a stringent stop list (PD+) is most important for optimizing the algorithms, followed by a strong stemming algorithm. Selection of the background set had relatively little effect on the performance of the keyword selection algorithm, which indicates that as long as the weight of a word is reduced if it occurs commonly in a fairly large set of MEDLINE abstracts, it may be less important as to how that set is chosen.

Therefore, the best keyword selection performance for the z-score scheme utilizes a random background set, the PD+ stop list and Porter's stemming algorithm. To

preclude the occurrence of the "New" words, which occur in the test but are missing in all background sets, we created a large random background set (about 112,000 pseudo-families of 50 abstracts each), which included all MEDLINE abstracts up to the year 2000. For the remaining study, we used the combination of this large random background set, PD+ stop list and Porter's stemming algorithm to extract keywords for each gene.

The use of keywords selected from gene-related literature to cluster functionally-related genes has two fundamental limitations. First, with the keyword selection algorithms described above, some words with high *z*-scores have low predictive potential for biological function or are erroneously associated with the gene in question (per observation of experts). Such results could occur more often when the gene name is referenced in the abstracts, but is not the actual topic of discussion, when the topic switches from the gene name to some other issue, or when the word "not" reverses the meaning of the sentence. Enhancements to the basic schemes could involve using natural language processing to exploit the added information in compound phrases, syntax, and grammatical structures such as negative sentences, and improving our stop list. The sensitivity analysis (Figure 3-1D) indicates that the quality of the stop list is the most important element in algorithm performance. Second, inconsistency among human investigators in the task of agreeing upon keywords from a document places a fundamental limit on our ability to evaluate the performance of computer algorithms against human opinion. Keyword selection by an investigator is ultimately subjective and leads to ambiguities in document classification (Funk and Reid, 1983; Blair and Maron, 1985; Swanson, 1960; Saracevic, 1991) with the consequence that performance better than ~75-80% precision may not be achievable.

For the reasons described above, the use of investigator-selected keywords as the "gold standard" for evaluating the performance of keyword-selection algorithms is imperfect. However, even in the face of these challenges, the keyword selection algorithms used here appear sufficiently robust to serve as the basis for functional gene clustering.

We also tested some other random background sets, such as one which contains all the Medline abstracts up to the year 2000. The keyword ranking is the same as the 50,000-abstract background set. The only difference is that the z-score value of each keyword is increased.

3.8 Comparison of TFIDF and Normalized Z-score Method for Keyword Extraction.

The performance of keyword-selection by TFIDF and normalized z-score methods were also evaluated with precision and recall metrics (Figure 3-2) by comparing the TFIDF and normalized z-score method outputs with the set of keywords selected by human investigators from an identical set of 100 abstracts (the first test set). Figure 3-2 shows that TFIDF outperforms the normalized z-score method with higher precision and recall values. Due to cosine normalization, the thresholds are much smaller in Figure 3-2 than those in Figure 3-1.

Word weighting is an important step in information retrieval, text mining, and text categorization for indexing documents. The main function of a word-weighting scheme is to enhance retrieval effectiveness (Salton and Buckley, 1988). In gene clustering by functional keyword associations, the weighting scheme is used to extract high quality keyword lists. Despite the variations in weighting schemes, the essential

ideas on which they are based can be grouped into a few categories (Kageura and Umino, 1996): (1) A "word" which appears once in a document is likely to be a keyword for that document; (2) a "word" which appears frequently in a document is likely to be a keyword for that document; (3) a "word" which appears only in a limited number of documents is likely to be a keyword for any document in which it appears; (4) a "word" which appears relatively more frequently in a document than in the other documents is likely to be a keyword for that document; (5) a "word" which shows a specific distribution in a collection of documents is likely to be a keyword for that collection of documents.

Categories (1) and (2) emphasize the "representation" aspect of keywords, and categories (3) and (4) emphasize the "discrimination" aspect. While categories (1) to (4) focus on individual documents, category (5) takes into account the relationships among documents as seen from the overall distribution of words. Therefore, category (5) has the advantage of considering topics as represented by a group of documents, while categories (1) to (4) only treat each document as a basic topic unit. Accordingly, the weighting schemes based on category (5) vary considerably, both in theoretical viewpoints and in the resultant weights given to words (Kageura and Umino, 1996). TFIDF is based on categories (1) to (4) because it considers the representation and discrimination aspects of keywords by combining the term frequency and inverse-document frequency. On the other hand, the word distribution in the background set is also taken into account in the z-score method because the word's average frequency and standard deviation in the background set are used to calculate the z-scores. Andrade and Valencia (1998) used a δ measure to present the distribution of the words in the background set. In their original z-

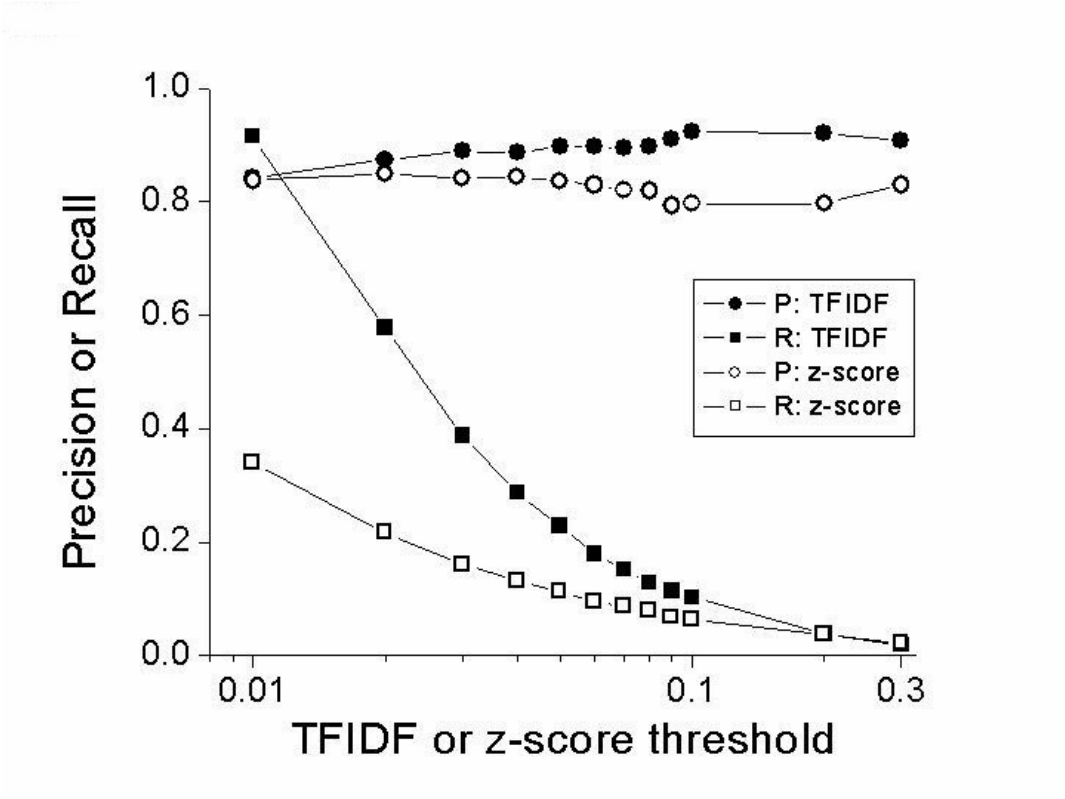


Figure 3-2. Keyword extraction by two weighting schemes (TFDIF and normalized z-score). Precision and recall is plotted as a function of the weight threshold for accepting a word.

score method, the abstracts in the background set were grouped by protein families, indicating that the abstracts inside a family were closely related. Therefore, it is reasonable to consider the relationship among families as seen from the overall word distribution. However, in the random background sets, the abstracts inside the pseudo-families were randomly chosen. Therefore, the word distribution among pseudo-families is meaningless. Our results show that TFIDF outperforms the normalized z-score method, indicating that the word distribution does not add any information to the metric.

A particular word is more likely to be repeated in a larger test set than in a shorter test set, and as a result, the term frequency of that word will be higher, which causes a higher TFIDF value since the IDF is the same. In our case, a larger test set means the gene has more abstracts and/or longer abstracts. Cosine normalization is applied in TFIDF so that the words in the longer documents are not unfairly given more weight. In order to compare with TFIDF, the z-score values were also normalized. In direct comparisons of cluster quality with keywords selected by the two schemes, TFIDF outperformed the normalized z-score for both test sets of genes.

3.9 Future experiments

As we mentioned in section 3.6, the text mining method we develop can extract new gene-to-disease information that we cannot find from publically available databases. One future experiment we can do is to test the gene osteopontin with the abstracts before 2001 to see if the method can predict the gene-disease relationship which was discovered after 2001.

3.10 Summary

In this chapter, we expand, extend, and optimize the z-score method by testing new background sets, a new stemming algorithm, and a new, extensive stop list customized for use with the biological literature. This extended method was used to create a repository of functional keywords from MEDLINE abstracts for genes. We also compare our results with information found in public databases. We then compare the performance of the z-score method with TFIDF for the purpose of extracting the functional keywords for each tested gene set by evaluating the precision and recall values. In the next chapter, we will use the keyword extraction strategy to extract functional keywords for each gene and cluster the gene based on the shared keywords. A new algorithm is proposed.

CHAPTER 4

Clustering Genes based on Keyword Feature Vectors

Partitioning genes into closely related groups has become an element of practically all analyses of microarray data (Cherepinsky et al., 2003).

A number of computer algorithms have been applied to gene clustering. Based on the assumption that genes with the same function or in the same biological pathway usually show similar expression patterns, the functions of unknown genes can be inferred from those of the known genes with similar expression profile patterns. Therefore, expression profile gene clustering by all the algorithms mentioned above has received much attention, however, the task of finding functional relationships between specific genes is left to the investigator. Manual scanning of the biological literature (for example, via MEDLINE) for clues regarding potential functional relationships among a set of genes is not feasible when the number of genes to be explored rises above approximately ten. Restricting the scan (manual or automatic) to annotation fields of GenBank, SwissProt or LocusLink is quicker but can suffer from the ad hoc relationship of keywords to the research interests of whoever submitted the entry. Moreover, keeping annotation fields current as new information appears in the literature is a major challenge that is rarely met adequately.

If, instead of organizing by expression pattern similarity, genes were grouped according to shared function, investigators might more quickly discover patterns or themes of biological processes that were revealed by their microarray experiments and focus on a select group of functionally related genes. A number of clustering strategies

based on shared functions rather than similar expression patterns have been devised. Chaussabel and Sher (2002) analyzed literature profiles generated by extracting the frequencies of certain terms from the abstracts in MEDLINE and then clustered the genes based on these terms, essentially applying the same algorithm used for expression pattern clustering. Jenssen et al. (2001) used co-occurrence of gene names in abstracts to create networks of related genes automatically. Text analysis of biomedical literature has also been applied successfully to incorporate functional information about the genes in the analysis of gene expression data (Blaschke et al., 2001; Raychaudhuri et al., 2002; Raychaudhuri et al., 2003; Masys et al., 2003) without generating clusters de novo. For example, Blaschke et al. (2001) extracted information about the common biological characteristics of gene clusters from MEDLINE using Andrade and Valencia's statistical text mining approach, which accepts user-supplied abstracts related to a protein of interest and returns an ordered set of keywords that occur in those abstracts more often than would be expected by chance (Andrade and Valencia, 1998).

In this chapter, we describe an approach that applies an algorithm called the Bond Energy Algorithm (BEA) (McCormick et al., 1972; Navathe et al., 1984) for functional gene clustering based on keyword association. We modify it so that the "affinity" among attributes (in our case genes) is defined based on the sharing of keywords between them and we develop a scheme for partitioning the clustered affinity matrix to produce clusters of genes. We call the resulting algorithm as BEA-PARTITION. BEA was originally conceived as a technique to cluster questions in psychological instruments (McCormick et al., 1972), has been used in operations research, production engineering, marketing, and various other fields (Arabie and Hubert, 1990), and is a popular clustering algorithm

in distributed database system (DDBS) design. The fundamental task of BEA in DDBS design is to group attributes based on their affinity, which indicates how closely related the attributes are, as determined by the inclusion of these attributes by the same database transactions. In our case, each gene is considered as an attribute. Hence, the basic premise is that two genes would have higher affinity, thus higher bond energy, if abstracts mentioning these genes shared many informative keywords. BEA has several useful properties (Navathe et al., 1984; Ozsu and Valduriez, 1999). First, it groups attributes with larger affinity values together, and the ones with smaller values together (i.e., during the permutation of columns and rows, it shuffles the attributes towards those with which they have higher affinity and away from those with which they have lower affinity). Second, the composition and order of the final groups are insensitive to the order in which items are presented to the algorithm. Finally, it seeks to uncover and display the association and interrelationships of the clustered groups with one another.

In this chapter, we develop a methodology to cluster the genes by shared functional keywords. Our gene clustering strategy is similar to the document clustering in information retrieval. Document clustering, defined as grouping documents into clusters according to their topics or main contents in an unsupervised manner, organizes large amounts of information into a small number of meaningful clusters and improves the information retrieval performance either via cluster-driven dimensionality reduction, term-weighting, or query expansion (Aslam et al., 1982; Willett, 1998; Jain et al., 1999; Baeza-Yates and Ribeiro-Neto, 1999; Sebastiani, 1999). In order to explore whether this algorithm could be useful for clustering genes derived from microarray experiments, we compared the performance of BEA-PARTITION, hierarchical clustering algorithm, self-

organizing map, and the *k*-means algorithm for clustering functionally-related genes based on shared keywords, using purity, entropy, and mutual information as metrics for evaluating cluster quality.

4.1 Keyword Extraction from biomedical literature

We use statistical methods to extract keywords from MEDLINE citations, based on the work of Andrade and Valencia (1998). This method estimates the significance of words by comparing the frequency of words in a given gene-related set (Test Set) of abstracts with their frequency in a background set of abstracts. We modify the original method by using a (i) different background set, (ii) a different stemming algorithm (Porter's stemmer), and (iii) a customized stop list as mentioned in Chapter 3.

For each gene analyzed, word frequencies were calculated from a group of abstracts retrieved by an SQL (structured query language) search of MEDLINE for the specific gene name, gene symbol or any known aliases (see LocusLink, ftp://ftp.ncbi.nih.gov/refseq/LocusLink/LL_tmpl.gz for gene aliases) in the TITLE field. The resulting set of abstracts (the Test Set) was processed to generate a specific keyword list.

4.1.1 Test sets of genes

We compared BEA-PARTITION and other clustering algorithms (*k*-means, hierarchical, and SOM) on two test sets.

1. 26 genes in four well-defined functional groups consisting of ten glutamate receptor subunits, seven enzymes in catecholamine metabolism, five cytoskeletal proteins and four enzymes in tyrosine and phenylalanine synthesis. The gene names and aliases

are listed in Table 4-1. This experiment was performed to determine whether keyword associations can be used to group genes appropriately and whether the four gene families or clusters that were known *a priori* would also be predicted by a clustering algorithm simply using the affinity metric based on keywords.

Table 4-1. 26 Genes manually clustered based on functional similarity

Group	Genes	Functions
1	*GluR1,GluR2,GluR3,GluR4,GluR6, KA1,KA2,NMDA-R1,NMDA-R2A, NMDA-R2B*	Glutamate receptor channels
2	*Tyrosine hydroxylase, DOPA decarboxylase, Dopamine beta-hydroxylase, Phenethanolamine N-methyltransferase, Monoamine oxidase A, Monoamine oxidase B, Catechol-O-methyltransferase*	Catecholamine synthetic enzymes
3	*Actin, Alpha-tubulin, Beta-tubulin, Alpha-spectrin, Dynein*	Cytoskeletal proteins
4	*Chorismate mutase, Prephenate dehydratase, Prephenate dehydrogenase, Tyrosine transaminase*	Enzymes in tyrosine and phenylalanine synthesis

2. 44 yeast genes involved in the cell cycle of budding yeast (*Saccharomyces cerevisiae*) that had altered expression patterns on spotted DNA microarrays (Eisen et al., 1998) were analyzed by Cherepinsky et al. (2003) to demonstrate their Shrinkage algorithm for gene clustering. A master list of member genes for each cluster was assembled according to a combination of 1) common cell-cycle functions and regulatory systems and 2) the corresponding transcriptional activators for each gene (Cherepinsky et al., 2003) (Table 4-2).

Table 4-2. 44 Yeast Genes grouped by transcriptional activators and cell cycle functions

Group	Activators	Genes	Functions
1	Swi4, Swi6	*Cln1, Cln2, Gic1, Gic2, Msb2, Rsr1, Bud9, Mnn1, Och1, Exg1, Kre6, Cwp1*	Budding
2	Swi6, Mbp1	*Clb5, Clb6, Rnr1, Rad27, Cdc21, Dun1, Rad51, Cdc45, Mcm2*	DNA replication and repair
3	Swi4, Swi6	*Htb1, Htb2, Hta1, Hta2, Hta3, Hho1*	Chromatin
4	Fkh1	*Hhf1, Hht1, Tel2, Apr7*	Chromatin
5	Fkh1	*Tem1*	Mitosis control
6	Ndd1, Fkh2, Mcm1	*Clb2, Ace2, Swi5, Cdc20*	Mitosis control
7	Ace2, Swi5	*Cts1, Egt2*	Cytokinesis
8	Mcm1	*Mcm3, Mcm6, Cdc6, Cdc46*	Prereplication complex formation
9	Mcm1	*Ste2, Far1*	Mating

4.1.2 Keyword Assessment

Statistical formulae from Andrade and Valencia (1998) for word frequencies were used without modification. These calculations were repeated for all gene names in the test set, a process that generated a database of keywords associated with specific genes, the strength of the association being reflected by a z-score.

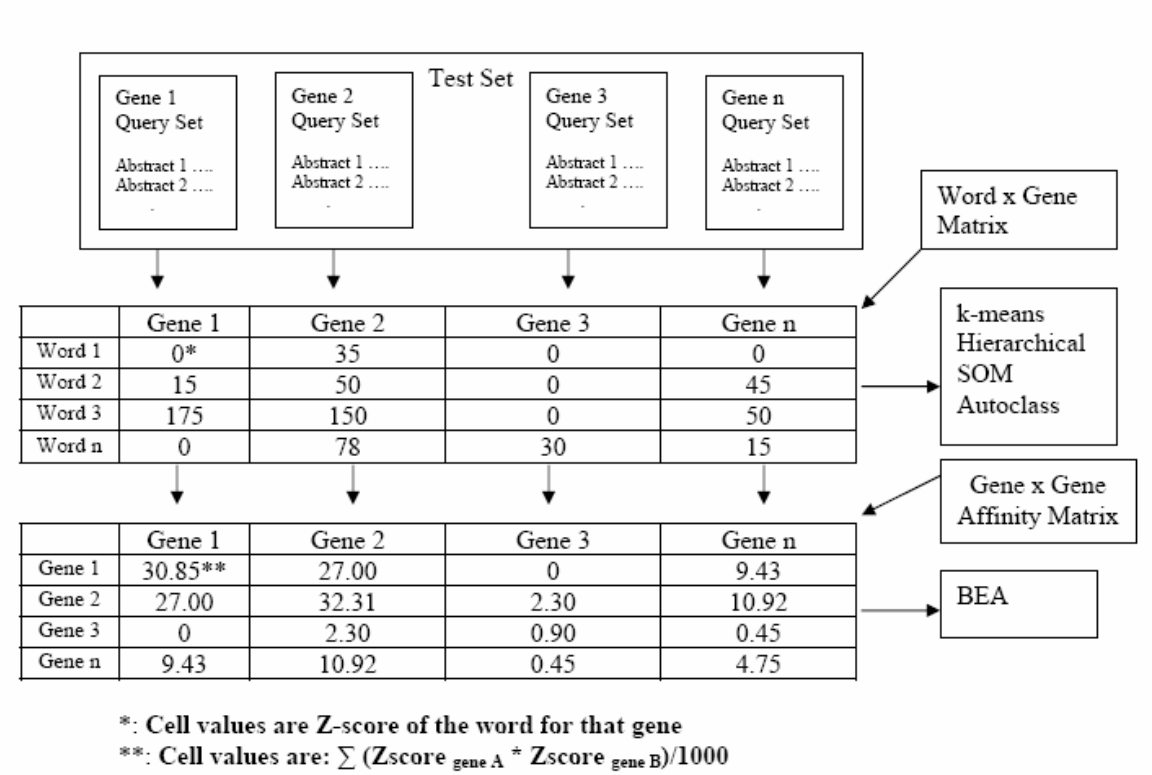
4.1.3 Keyword selection for gene clustering

We used z-score thresholds to select the keywords used for gene clustering. Those keywords with z-scores less than the threshold were discarded. The z-score thresholds we tested were 0, 5, 8, 10, 15, 20, 30, 50, and 100. The output of the keyword selection for all genes in each Test Set is represented as a sparse keyword (rows) x gene (columns) matrix with cells containing z-scores.

4.2 List of keywords and keyword x gene matrix generation

A list of keywords was generated for each gene to build the keyword x gene matrix. Keywords were sorted according to their z-scores. The keyword selection experiment (see below) showed that a z-score threshold of 10 generally produced better results, which suggests that keywords with z-scores lower than 10 have less information content, e.g. “cell”, “express”. The relative values of z-scores depend on the size of the background set (data not shown). Since we used 5.6 million abstracts as the background set, the z-scores of most of the informative keywords were well above 10 (based on smaller values of standard deviation in the definition of z-score). The keyword x gene matrices were used as inputs to *k*-means, hierarchical clustering algorithm, self-

organizing map, while as required by the BEA approach, they were first converted to a gene X gene matrix based on common shared keywords and these gene x gene matrices were used as inputs to BEA-PARTITION. An overview of the gene clustering by shared keyword process is provided in Figure 4-1.



	Gene 1	Gene 2	Gene 3	Gene n
Word 1	0*	35	0	0
Word 2	15	50	0	45
Word 3	175	150	0	50
Word n	0	78	30	15

	Gene 1	Gene 2	Gene 3	Gene n
Gene 1	30.85**	27.00	0	9.43
Gene 2	27.00	32.31	2.30	10.92
Gene 3	0	2.30	0.90	0.45
Gene n	9.43	10.92	0.45	4.75

Figure 4-1. Procedure for clustering genes by the strength of their associated keywords

4.3 BEA-PARTITION: a new algorithm with application to gene clustering

The BEA-PARTITION takes a symmetric matrix as input, permutes its rows and columns, and generates a sorted matrix, which is then partitioned to form a clustered matrix.

4.3.1 Constructing the symmetric gene x gene matrix

The sparse word X gene matrix, with the cells containing the z-scores of each word-gene pair, was converted to a gene X gene matrix with the cells containing the sum of products of z-scores for shared keywords. The z-score value was set to zero if the value was less than the threshold. Larger values reflect stronger and more extensive keyword associations between gene-gene pairs. For each gene pair (*Gi*, *Gj*) and every word *a* they share in the sparse word x gene matrix, the *Gi* x Gj cell value (*aff(Gi,Gj)*) in the gene X gene matrix represents the affinity of the two genes for each other and is calculated as:

$$aff(G_i, G_j) = \frac{\sum_{a=1}^{N} (Z_{G_i}^{a} \times Z_{G_j}^{a})}{1000} \qquad (4\text{-}1)$$

Dividing the sum of the z-score product by 1000 was done to reduce the typically large numbers to a more readable format in the output matrix.

This calculation is called cosine similarity calculation in text mining, which is a popular way to find out how close two documents are.

4.3.2 Sorting the matrix (Ozsu and Valduriez, 1999)

The sorted matrix is generated as follows:

1. *Initialization*. Place and fix one of the columns of symmetric matrix arbitrarily into the clustered matrix;

2. *Iteration*. Pick one of the remaining *n-i* columns (where *i* is the number of columns already in the sorted matrix). Choose the placement in the sorted matrix that maximizes the change in bond energy as described below (equation 3). Repeat this step until no more columns remain;

3. *Row ordering*. Once the column ordering is determined, the placement of the rows should also be changed correspondingly so that their relative positions match the relative position of the columns. This restores the symmetry to the sorted matrix.

To calculate the change in bond energy for each possible placement of the next ($i+1$) column, the bonds between that column (k) and each of two newly adjacent columns (i, j) are added and the bond that would be broken between the latter two columns is subtracted. Thus, the "bond energy" between these three columns i, j, and k (representing gene i (Gi); gene j (Gj); gene k (Gk))) is calculated by the following interaction measure:

$$energy(G_i, G_j, G_k) = 2 \times [bond(G_i, G_k) + bond(G_k, G_j) - bond(G_i, G_j)] \quad (4\text{-}2)$$

where bond(*Gi,Gj*) is the bond energy between gene *Gi* and gene *Gj* and

$$bond(G_i, G_j) = \sum_{r=1}^{N} aff(G_r, G_i) \times aff(G_r, G_j) \quad (4\text{-}3)$$

$$aff(G_0, G_i) = aff(G_i, G_0) = aff(G_{(n+1)}, G_i) = aff(G_i, G_{(n+1)}) = 0 \quad (4\text{-}4)$$

The last set of conditions (equation 4-5) takes care of cases where a gene is being placed in the sorted matrix to the left of the leftmost gene or to the right of the rightmost gene during column permutations, and prior to the topmost row and following the last row during row permutations.

4.3.3 Partitioning the sorted matrix

The original BEA algorithm (McCormick et al., 1972) did not propose how to partition the sorted matrix. The partitioning heuristic was added by Navathe et al. (1984) for the problems in the distributed database design. These heuristics were constructed using the goals of design: to minimize access time and storage costs. We do not have the luxury of such a clear cut objective function in our case. Hence, to partition the sorted matrix into submatrices, each representing a gene cluster, we experimented with different heuristics, and finally derived a heuristic that identifies the boundaries between clusters by sequentially finding the maximum sum of the quotients for corresponding cells in adjacent columns across the matrix. With each successive split, only those rows corresponding to the remaining columns were processed, i.e. only the remaining

symmetrical portion of the submatrix was used for further iterations of the splitting algorithm. The number of clusters into which the gene affinity matrix was partitioned was determined by AUTOCLASS (described below), however, other heuristics might be useful for this determination. The boundary metric (*B*) for columns *Gi* and *Gj* was defined as:

$$B(G_i, G_j) = \max_{index(p-1) \le q \le index(p)} \sum_{k=index(p-1)}^{index(p)} \frac{\max(aff(k,q), aff(k,q+1))}{\min(aff(k,q), aff(k,q+1))} \quad (4\text{-}5)$$

where *q* is the new splitting point (for simplicity, we use the number of the leftmost column in the new submatrix that is to the right of the splitting point), which will split the submatrix defined between two previous splitting points, index(*p)* and index(*p*-1) (which do not necessarily represent contiguous columns). To partition the entire sorted matrix, the following initial conditions are set, index(*p)* = N, index(*p*-1) =0.

4.4 Other Clustering Algorithm

K-means and hierarchical clustering analysis were performed using Cluster/Treeview programs available online (http://bonsai.ims.u-tokyo.ac.jp/~mdehoon/software/cluster/software.htm).

Self-organizing map

Self-organizing map was performed using GeneCluster 2.0 (http://www.broad.mit.edu/cancer/software/software.html).

Euclidean distance measure was used for gene X keyword matrix to determine similarity among genes. When gene X gene matrix was used as input, the gene similarity was calculated by equation 4-1.

4.4.1 Determination of number of clusters

In order to apply BEA-PARTITION and *k*-means clustering algorithms, the investigator needs to have *a priori* knowledge about the number of clusters in the test set. We determined the number of clusters by applying AUTOCLASS, an unsupervised Bayesian classification system developed by (Cheeseman and Stutz, 1996). AUTOCLASS, which seeks a maximum posterior probability classification, determines the optimal number of classes in large data sets. Among a variety of applications, AUTOCLASS has been used for the discovery of new classes of infra-red stars in the IRAS Low Resolution Spectral catalogue, new classes of airports in a database of all USA airports, and discovery of classes of proteins, introns and other patterns in DNA/protein sequence data (Cheeseman and Stutz, 1996). We applied an open source implementation of AUTOCLASS (http://ic.arc.nasa.gov/ic/projects/bayes-group/autoclass/autoclass-c-program.html). The resulting number of clusters was then

used as the endpoint for the partitioning step of the BEA-PARTITION algorithm. To determine whether AUTOCLASS could discover the number of clusters in the test sets correctly, we also tested different number of clusters other than the ones AUTOCLASS predicted.

4.4.2 Determination of z-score threshold

The effect of using different z-score thresholds for keyword selection on the quality of resulting clusters is shown in Figure 4-2A1 and 4-2B1. For both test sets, BEA-PARTITION produced clusters with higher mutual information when z-score thresholds were within a range of 10 to 20. For the 44-gene set, *K*-means produced clusters with the highest mutual information when the z-score threshold was 8, while, for the 26-gene set, mutual information was highest when z-score threshold was 15. For the remaining studies, we chose to use a z-score threshold of 10 to keep as many functional keywords as possible.

Once the keyword X gene matrix was created, we used AUTOCLASS to decide the number of clusters in the test sets. AUTOCLASS took the keyword X gene matrix as input and predicted that there were 5 clusters in the set of 26 genes and 9 clusters in the set of 44 yeast genes. The effect of the numbers of clusters on the algorithm performance was shown in Figures 4-2A2 and 4-B2. BEA-PARTITION again produced a better result regardless of the number of clusters used. BEA-PARTITION had the highest mutual information when the numbers of clusters were 5 (26-gene set) and 9 (44-gene set), whereas *k*-means worked marginally better when the numbers of clusters were 8 (26-gene set) and 10 (44-gene set). Based on these results we chose to use 5 and 9 clusters,

respectively, for the 26-gene and 44-gene data sets, because the probabilities were higher than the other choices.

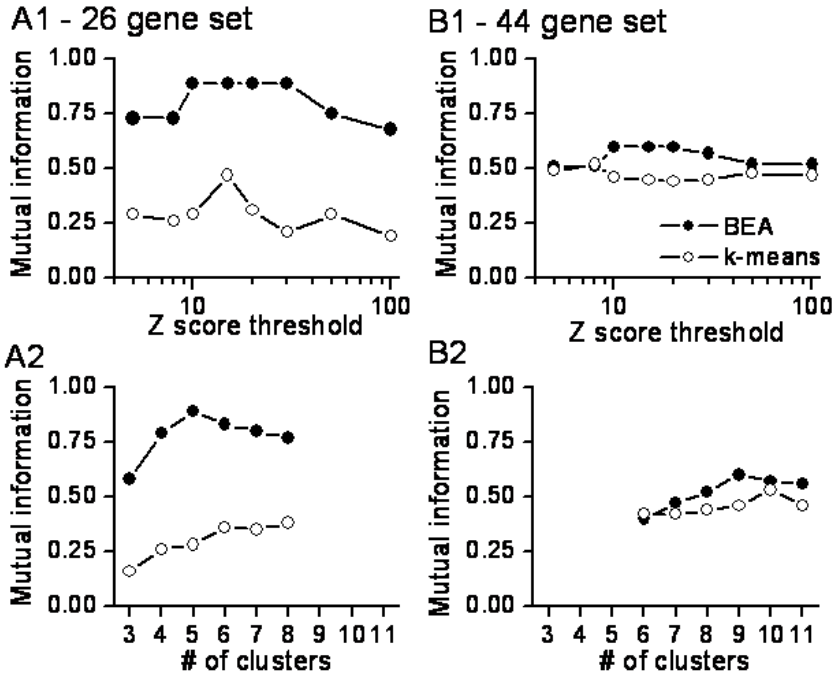


Figure 4-2. Effect of keyword selection by z-score thresholds (A1 and B1) and different number of clusters (A2 and B2) on the cluster quality. Z-score thresholds were used to select the keywords for gene clustering. Those keywords with z-scores less than the threshold were discarded. To determine the effect of keyword selection by z-score thresholds on cluster quality, we tested z-score thresholds 0, 5, 8, 10, 15, 20, 30, 50, and 100. To determine whether AUTOCLASS could be used to discover the number of clusters in the test sets correctly, we tested different number of clusters other than the ones AUTOCLASS predicted (4 for the 26-gene set and 9 for the 44-gene set).

4.5 Evaluating the clustering results

To evaluate the quality of our resultant clusters, we used the established metrics of Purity, Entropy and Mutual Information, which are briefly described below (Strehl, 2002). Let us assume that we have C classes (i.e. C expert clusters, as shown in Tables 1 and 2), while our clustering algorithms produce K clusters, $\pi_1, \pi_2, \ldots, \pi_k$.

4.5.1 Purity

Purity can be interpreted as classification accuracy under the assumption that all objects of a cluster are classified to be members of the dominant class for that cluster. If the majority of genes in cluster A are in class B, then class B is the dominant class. Purity is defined as the ratio between the number of items in cluster π_i from dominant class *j* and the size of cluster π_i, that is:

$$P(\pi_i) = \frac{1}{n_i} \max_j (n_i^j), i = 1,2,...,k \qquad (4\text{-}6)$$

where $n_i = |\pi_i|$, that is, the size of cluster i and n_i^j is the number of genes in π_i that belong to class j, j = 1, 2, . . . ,C. The closer to 1 the purity value is, the more similar this cluster is to its dominant class. Purity is measured for each cluster and the average purity of each test gene set cluster result was calculated.

4.5.2 Entropy

Entropy denotes how uniform the cluster is. If a cluster is composed of genes coming from different classes, then the value of entropy will be close to 1. If a cluster only contains one class, the value of entropy will be close to 0. The ideal value for entropy would be zero. Lower values of entropy would indicate better clustering. Entropy is also measured for each cluster and is defined as:

$$E(\pi_i) = -\frac{1}{\log C}\sum_{j=1}^{C}\frac{n_i^j}{n_i}\log(\frac{n_i^j}{n_i}) \qquad (4\text{-}7)$$

The average entropy of each test gene set cluster result was also calculated.

4.5.3 Mutual Information

One problem with purity and entropy is that they are inherently biased to favor small clusters. For example, if we had one object for each cluster, then the value of purity would be 1 and entropy would be zero, no matter what the distribution of objects in the expert classes is.

Mutual information is a symmetric measure for the degree of dependency between clusters and classes. Unlike correlation, mutual information also takes higher order dependencies into account. We use mutual information because it captures how related clusters are to classes without bias towards small clusters. Mutual information is a measure of the discordance between the algorithm-derived clusters and the actual clusters. It is the measure of how much information the algorithm-derived clusters can tell us to infer the actual clusters. Random clustering has mutual information of 0 in the limit. Higher mutual information indicates higher similarity between the algorithm-derived clusters and the actual clusters. Mutual information is defined as:

$$M(\pi) = \frac{2}{N} \sum_{i=1}^{K} \sum_{j=1}^{C} n_i^j \frac{\log \frac{n_i^j \times N}{\sum_{t=1}^{K} n_i^t \sum_{t=1}^{C} n_i^t}}{\log(K \times C)} \quad (4\text{-}8)$$

where N is the total number of genes being clustered and K is the number of clusters the algorithm produced, and C is the number of expert classes.

4.6 A comparative study of BEA-PARTITION with other Clustering algorithms: Using 26-gene set.

To determine whether keyword associations could be used to group genes appropriately, we cluster the 26-gene set with either BEA-PARTITION or *k*-means. Keyword lists are generated for each of these 26 genes, which belong to one of four well-defined functional groups (Table 4-1). The resulting word x gene matrix has 26 columns (genes) and approximately 8,540 rows (words with z-scores >= 10 appearing in any of the query sets). The BEA-PARTITION, with z-score threshold = 10, correctly assigns 25 of 26 genes to the appropriate cluster based on the strength of keyword associations (Figure 4-3). Tyrosine transaminase is the only outlier. As expected from the BEA-PARTITION, cells inside clusters tend to have much higher values than those outside. Hierarchical clustering algorithm, with the gene X keyword matrix as the input, generate similar result as BEA-PARTITION (5 clusters and TT is the outlier) (Figure 4-4A).

While BEA-PARTITION and hierarchical clustering algorithm produce clusters very similar to the original functional classes, those produced by *k*-means (Table 4-4), self-organizing map (Table 4-5), and AUTOCLASS (Table 4-6), with gene X keyword

matrix as input, are heterogeneous and thus more difficult to explain. The average purity, average entropy, and mutual information of the BEA-PARTITION and hierarchical algorithm result are 1, 0, and 0.88, while those of *k*-means result are 0.53, 0.65 and 0.28, respectively, those of SOM result are 0.76, 0.35, and 0.18, respectively, and those of AUTOCLASS result are 0.82, 0.28, and 0.56 (Table 4-3) (gene X keyword matrix as input). When gene X gene matrix is used as input to hierarchical algorithm, k-means, and SOM, the results are even worse as measured by purity, entropy, and mutual information (Table 4-3).

The results, with gene X gene matrix as the input, are shown in Tables 4-7, 4-8, 4-9, and 4-10.

	MOA	MOB	COM	DOPA	TH	PNMT	DBH	CM	PD2	PD1	Beta-tubulin	Alpha-Tubulin	Dynein	Actin	Alpha-Spectrin	GluR1	GluR3	GluR4	GluR2	GluR6	KA2	KA1	NMDA-R1	NMDA-R2A	NMDA-R2B	TT
MOA	9057	1937	250	154	32	83	43	3	5	1	0	2	0	0	2	6	9	15	11	5	8	0	6	3	12	0
MOB	1937	11465	321	162	34	45	39	2	3	2	1	1	0	0	0	7	17	12	11	6	7	5	5	3	23	0
COM	250	321	10021	600	40	185	107	3	4	6	2	0	0	0	2	2	4	6	5	5	3	0	3	3	15	1
DOPA	154	162	600	10104	195	41	118	42	26	44	1	1	0	0	4	5	36	5	12	16	9	0	11	10	40	8
TH	32	34	40	195	1023	150	206	7	8	7	1	2	0	0	0	15	8	15	6	8	14	0	8	10	34	8
PNMT	83	45	185	41	150	23736	612	4	2	4	3	0	0	0	0	9	24	8	26	25	19	0	17	9	39	13
DBH	43	39	107	118	206	612	7537	37	160	80	2	0	1	0	40	1	11	0	6	2	1	0	3	1	8	9
CM	3	2	3	42	7	4	37	110194	43746	24460	1	1	8	1	6	4	1	4	9	2	1	3	5	6	6	12
PD2	5	3	4	26	8	2	160	43746	792347	47172	1	1	1	1	9	4	6	50	4	2	4	10	2	57	54	139
PD1	1	2	6	44	7	4	80	24460	47172	773747	596	1	0	1	24	1	14	2	2	0	1	0	0	0	2	202
Beta-tubulin	0	1	2	1	1	3	2	1	1	596	8995	1579	148	35	10	2	2	3	8	48	151	1	7	3	3	6
Alpha-Tubulin	2	1	0	1	2	0	0	1	1	1	1579	10320	363	39	8	11	22	14	5	5	10	6	7	2	3	2
Dynein	0	0	0	0	0	0	1	8	1	0	148	363	7362	18	13	2	1	0	2	7	14	1	10	73	48	0
Actin	0	0	0	0	0	0	0	1	1	1	35	39	18	1605	46	2	0	3	1	0	1	0	1	0	1	0
Alpha-Spectrin	2	0	2	4	0	0	40	6	9	24	10	8	13	46	48696	45	17	4	14	42	9	2	9	3	5	4
GluR1	6	7	2	5	15	9	1	4	4	1	2	11	2	2	45	33849	6855	7625	6591	2978	8538	5918	1799	604	378	0
GluR3	9	17	4	36	8	24	11	1	6	14	2	22	1	0	17	6855	362444	9033	5465	2724	3261	1022	505	69	291	0
GluR4	15	12	6	5	15	8	0	4	50	2	3	14	0	3	4	7625	9033	426204	9046	3155	6120	1306	911	173	179	1
GluR2	11	11	5	12	6	26	6	9	4	2	8	5	2	1	14	6591	5465	9046	37311	10097	6815	1097	1001	470	245	1
GluR6	5	6	5	16	8	25	2	2	2	0	48	5	7	0	42	2978	2724	3155	10097	134750	85155	6183	864	159	198	0
KA2	8	7	3	9	14	19	1	1	4	1	151	10	14	1	9	8538	3261	6120	6815	85155	793545	27719	969	171	244	0
KA1	0	5	0	0	0	0	0	3	10	0	1	6	1	0	2	5918	1022	1306	1097	6183	27719	930731	1977	314	315	0
NMDA-R1	6	5	3	11	8	17	3	5	2	0	7	7	10	1	9	1799	505	911	1001	864	969	1977	22376	8914	5412	0
NMDA-R2A	3	3	3	10	10	9	1	6	57	0	3	2	73	0	3	604	69	173	470	159	171	314	8914	71927	16837	0
NMDA-R2B	12	23	15	40	34	39	8	6	54	2	3	3	48	1	5	378	291	179	245	198	244	315	5412	16837	37712	0
TT	0	0	1	8	8	13	9	12	139	202	6	2	0	0	4	0	0	1	1	0	0	0	0	0	0	4998

Figure 4-3. Gene clusters by keyword associations using BEA-PARTITION. Keywords with z-scores >=10 were extracted from MEDLINE abstracts for 26 genes in 4 functional classes. The resulting word x gene sparse matrix was converted to a gene x gene matrix. The cell values are the sum of z-score products for all keywords shared by the gene pair. This value is divided by 1000 for purpose of display.

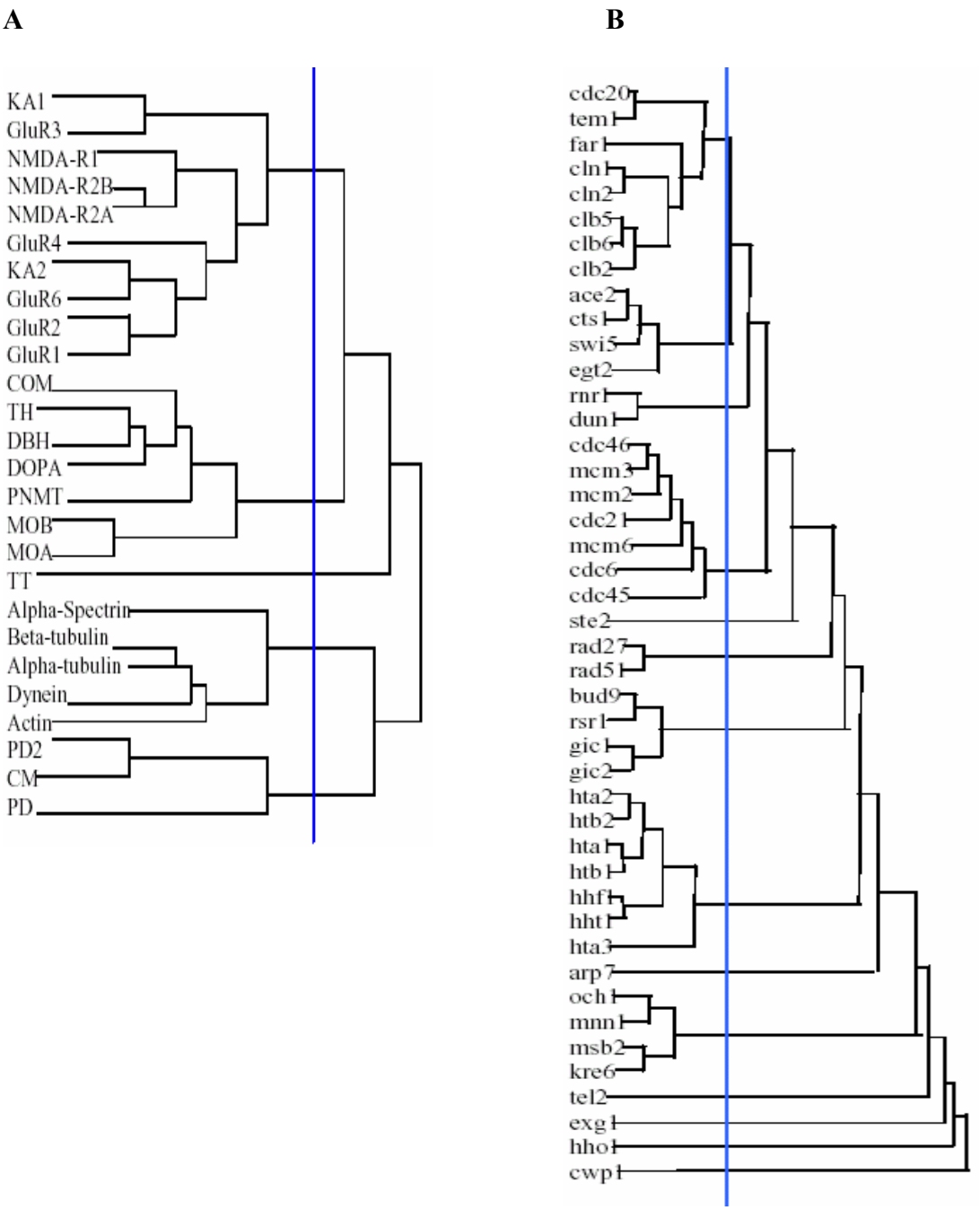


Figure 4-4. Gene clusters by keyword associations using hierarchical clustering algorithm. Keywords with z-scores >=10 were extracted from MEDLINE abstracts for 26 genes in 4 functional classes (A) and 44-gene in 9 classes (B). The resulting word x gene sparse matrix is used as input to the hierarchical algorithm.

Table 4-3. The quality of the gene clusters derived by different clustering algorithms, measured by Purity, Entropy, and Mutual Information

Input Matrix	Test gene set	Clustering algorithm	Average Purity	Average Entropy	Mutual Information
Gene X keyword matrix	26-gene set	Hierarchical	1	0	0.88
		k-means	0.53	0.65	0.28
		SOM	0.76	0.35	0.18
		Autoclass	0.82	0.28	0.56
	44-gene set	Hierarchical	0.86	0.12	0.58
		k-means	0.60	0.37	0.46
		SOM	0.61	0.33	0.39
		Autoclass	0.57	0.39	0.49
Gene X Gene matrix	26-gene set	BEA-PARTITION	1	0	0.88
		Hierarchical	1	0	0.88
		k-means	0.87	0.19	0.16
		SOM	0.81	0.28	0.20
		Autoclass	0.89	0.13	0.78
	44-gene set	BEA-PARTITION	0.74	0.24	0.60
		Hierarchical	0.84	0.16	0.56
		k-means	0.84	0.12	0.30
		SOM	0.71	0.27	0.35
		Autoclass	0.72	0.26	0.51

Table 4-4. 26-gene set *k*-means result (gene X keyword matrix as input)

Cluster	Gene	Function
1	*Dynein, Alpha-Tublin*	Cytoskeletal proteins
	MOB (Monoamine oxidase B),	Catecholamine synthetic enzymes
	MOA (Monoamine oxidase A)	
2	*GluR1,GluR2,GluR6,KA2,NMDA-R1*	Glutamate receptor channels
	PNMT (Phenethanolamine N-methyltransferase)	Catecholamine synthetic enzymes
3	*Actin,Beta-Tublin*	Cytoskeletal proteins
	DBH (Dopamine beta-hydroxylase),	Catecholamine synthetic enzymes
	DOPA (DOPA decarboxylase)	
	NMDA-R2B	Glutamate receptor channels
4	*COM (Catechol-O-methyltransferase)*	Catecholamine synthetic enzymes
	GluR3,GluR4,KA1	Glutamate receptor channels
	PD1 (Prephenate dehydratase),	Enzymes in tyrosine synthesis
	PD2 (Prephenate dehydrogenase)	
5	*Alpha-Spectrin*	Cytoskeletal proteins
	TH (Tyrosine hydroxylase)	Catecholamine synthetic enzymes
	NMDA-R2A	Glutamate receptor channels
	CM (Chorismate mutase),	Enzymes in tyrosine synthesis
	TT (*tyrosine transaminase*)	

Table 4-5. 26-gene SOM result (gene X keyword matrix as input)

Cluster	Gene	Function
1	*Actin,Alpha-Spectrin,Alpha-Tubulin Beta-tubulin,Dynein*	Cytoskeletal proteins
	GluR1,GluR2,GluR3,NMDA-R1,NMDA-R2A,NMDA-R2B	Glutamate receptor channels
	DBH (Dopamine beta-hydroxylase), *COM (Catechol-O-methyltransferase)* *DOPA (DOPA decarboxylase)* *MOB (Monoamine oxidase B),* *MOA (Monoamine oxidase A)* *TH (Tyrosine hydroxylase)* *PNMT (Phenethanolamine N-methyltransferase*	Catecholamine synthetic enzymes
	TT (*tyrosine transaminase*) *CM (Chorismate mutase)*	Enzymes in tyrosine synthesis
2	*GluR6*	Glutamate receptor channels
3	*GluR4* *KA2*	Glutamate receptor channels
4	*KA1*	Glutamate receptor channels
	PD2 (Prephenate dehydrogenase) *PD1 (Prephenate dehydratase)*	Enzymes in tyrosine synthesis

Table 4-6. 26-gene AUTOCLASS result (gene X keyword matrix as input)

Cluster	Gene	Function
1	*Alpha-Spectrin*	Cytoskeletal proteins
	DBH (Dopamine beta-hydroxylase), *DOPA (DOPA decarboxylase)* *TH (Tyrosine hydroxylase)*	Catecholamine synthetic enzymes
	NMDA-R1	Glutamate receptor channels
2	*GluR2,GluR3,GluR4,GluR6,NMDA-R2A,NMDA-R2B*	Glutamate receptor channels
3	*GluR1,KA1,KA2*	Glutamate receptor channels
	PD2 (Prephenate dehydrogenase) *PD1 (Prephenate dehydratase)* *TT (tyrosine transaminase)*	Enzymes in tyrosine synthesis
	CM (Chorismate mutase) *PNMT (Phenethanolamine N-methyltransferas)*	Catecholamine synthetic enzymes
4	*Actin,Alpha-Tubulin,Beta-tubulin* *Dynein*	Cytoskeletal proteins
5	*MOB (Monoamine oxidase B),* *MOA (Monoamine oxidase A)* *COM (Catechol-O-methyltransferase)*	Catecholamine synthetic enzymes

Table 4-7. 26-gene set *k*-means result (gene X gene matrix as input)

Cluster	Gene	Function
1	*PD1 (Prephenate dehydratase),*	Enzymes in tyrosine synthesis
2	*PD2 (Prephenate dehydrogenase)*	Enzymes in tyrosine synthesis
3	*Actin,Beta-Tublin,Dynein, Alpha-Tublin Alpha-Spectrin*	Cytoskeletal proteins
	MOB (Monoamine oxidase B), *MOA (Monoamine oxidase A),* *PNMT (Phenethanolamine N-methyltransferase)* *DBH (Dopamine beta-hydroxylase),* *DOPA (DOPA decarboxylase)* *COM (Catechol-O-methyltransferase)* *TH (Tyrosine hydroxylase)*	Catecholamine synthetic enzymes
	GluR1,GluR2,GluR3,GluR4,GluR6,NMDA-R1,NMDA-R2A,NMDA-R2B	Glutamate receptor channels
	CM (Chorismate mutase), *TT* (*tyrosine transaminase*)	Enzymes in tyrosine synthesis
4	*KA1*	Glutamate receptor channels
5	*KA2*	Glutamate receptor channels

Table 4-8. 26-gene set Hierarchical cluster result (gene X gene matrix as input)

Cluster	Gene	Function
1	*CM (Chorismate mutase),PD1 (Prephenate dehydratase), PD2 (Prephenate dehydrogenase), TT* (*tyrosine transaminase*)	Enzymes in tyrosine synthesis
2	*Actin,Beta-Tublin,Dynein, Alpha-Tublin*	Cytoskeletal proteins
3	*MOB (Monoamine oxidase B),* *MOA (Monoamine oxidase A),* *PNMT (Phenethanolamine N-methyltransferase)* *DBH (Dopamine beta-hydroxylase),* *DOPA (DOPA decarboxylase)* *COM (Catechol-O-methyltransferase)* *TH (Tyrosine hydroxylase)*	Catecholamine synthetic enzymes
4	*GluR1,GluR2,GluR3,GluR4,GluR6,KA1, KA2,NMDA-R1,NMDA-R2A,NMDA-R2B*	Glutamate receptor channels
5	*Alpha-Spectrin*	Cytoskeletal proteins

Table 4-9. 26-gene SOM result (gene X gene matrix as input)

Cluster	Gene	Function
1	*GluR3,GluR4*	Glutamate receptor channels
1	*Actin,Alpha-Spectrin,Alpha-Tubulin Beta-tubulin,Dynein*	Cytoskeletal proteins
	GluR1,GluR2,NMDA-R1,NMDA-R2A NMDA-R2B	Glutamate receptor channels
	DBH (Dopamine beta-hydroxylase), COM (Catechol-O-methyltransferase) DOPA (DOPA decarboxylase) MOB (Monoamine oxidase B), MOA (Monoamine oxidase A) TH (Tyrosine hydroxylase) PNMT (Phenethanolamine N-methyltransferase	Catecholamine synthetic enzymes
	TT (*tyrosine transaminase*) *CM (Chorismate mutase)*	Enzymes in tyrosine synthesis
2	*GluR6*	Glutamate receptor channels
3	*KA2*	Glutamate receptor channels
4	*KA1*	Glutamate receptor channels
	PD2 (Prephenate dehydrogenase) PD1 (Prephenate dehydratase)	Enzymes in tyrosine synthesis

Table 4-10. 26-gene AUTOCLASS result (gene X gene matrix as input)

Cluster	Gene	Function
1	*DBH (Dopamine beta-hydroxylase),* *DOPA (DOPA decarboxylase)* *TH (Tyrosine hydroxylase)*	Catecholamine synthetic enzymes
2	*GluR2,GluR3,GluR4,GluR6,NMDA-R1* *NMDA-R2A,NMDA-R2B*	Glutamate receptor channels
3	*GluR1,KA1,KA2*	Glutamate receptor channels
	PD2 (Prephenate dehydrogenase) *PD1 (Prephenate dehydratase)* *TT (tyrosine transaminase)* *CM (Chorismate mutase)*	Enzymes in tyrosine synthesis
	PNMT (Phenethanolamine N-methyltransferas)	Catecholamine synthetic enzymes
4	*Actin,Alpha-Spectrin,Alpha-Tubulin* *Beta-tubulin,Dynein*	Cytoskeletal proteins
5	*MOB (Monoamine oxidase B),* *MOA (Monoamine oxidase A)* *COM (Catechol-O-methyltransferase)*	Catecholamine synthetic enzymes

4.7 A comparative study of BEA-PARTITION with other Clustering algorithms: Using 44-gene set.

To determine whether our test mining/gene clustering approach could be used to group genes identified in microarray experiments, we cluster 44 yeast genes taken from Eisen et al. (1998) via Cherepinsky et al. (2003), again using BEA-PARTITION, hierarchical algorithm, SOM, and *k*-means. Keyword lists are generated for each of the 44 yeast genes (Table 4-2) and a 3,882 (words appearing in the query sets with z-score greater or equal 10) x 44 (genes) matrix is created. The clusters produced by the BEA-PARTITION, *k*-means, SOM, and AUTOCLASS are shown in tables 7, 8, 9, and 10 respectively, whereas those produced by hierarchical algorithm are shown in Figure 4B. The average purity, average entropy, and mutual information of the BEA-PARTITION result are 0.74, 0.24, and 0.60, whereas those of hierarchical algorithm, SOM, *k*-means, and AUTOCLASS results (gene X keyword matrix as input) are 0.86, 0.12, and 0.58; 0.60, 0.37, and 0.46; 0.61, 0.33, and 0.39; 0.57, 0.39, and 0.49, respectively (Table 4-3).

A new notation to represent the resulting cluster sets and a scoring function were introduced by Cherepinsky et al. (2003). We argue that the scoring function (Error Score = FP + FN), like purity and entropy, is also biased to favor the small cluster. If each cluster only had one gene, for each cluster, there is no false positive or false negative (FP = 0 and FN = 0). Therefore Error Score = 0.

We used the new notation to represent our resulting clusters:

BEA result:

{1 -> {{4, *}, {4, 3}, {2, 2}, {2, 4}},

2 -> {{4, 2}, {4, 4}, {1, 7}},

3 -> {{5, 3}, {1, 3}},

4 -> {{2, 6}, {1, *}, {1, 4}},

5 -> {{1, 5}},

6 -> {{4, 2}},

7 -> {{2, 4}},

8 -> {{4, 4}},

9 -> {{1, 5}, {1, 3}}

}

The Error Score = 67 + 87 = 154.

The error score of Cherepinsky clusters is:

Error Score = 76 + 88 = 164.

While the error score of Eisen cluster is:

Error Score = 370 + 79 = 449.

So the cluster result produced by BEA, clustering the genes by the functional keyword association reduces the false positives (FPs) and false negatives (FNs).

Table 4-11. 44 Yeast genes BEA-PARTITION result (gene X keyword matrix as input)

Clusters	Activators	Genes
1	Swi4, Swi6	*Cwp1, Exg1, Mnn1, Och1*
2	Fkh1	*Arp7*
3	Ndd1, Fkh2, Mcm1	*Cdc20, Swi5, Ace2, Clb2*
	Ace2, Swi5	*Egt2, Cts1*
4	Swi4, Swi6	*Bud9, Rsr1, Gic1, Gic2*
	Mcm1	*Far1*
	Fkh1	*Tem1*
5	Swi4, Swi6	*Cln1, Cln2*
	Swi6, Mbp1	*Clb5, Clb6, Rnr1, Dun1*
6	Swi4, Swi6	*Hta1, Hta3, Hta2, Htb2, Htb1*
	Fkh1	*Hhf1, Hht1*
	Swi6, Mbp1	*Rad51*
7	Swi4, Swi6	*Kre6, Msb2*
	Swi4, Swi6	*Hho1*
	Mcm1	*Ste2*
8	Fkh1	*Tel2*
9	Swi6, Mbp1	*Rad27, Cdc45, Mcm2, Cdc21*
	Mcm1	*Cdc46, Mcm3, Mcm6, Cdc6*

Table 4-12. 44 Yeast gene SOM result (gene X keyword as input)

Clusters	Activators	Genes
1	Swi4, Swi6	*Gic1,Gic2,Msb2*
2	Fkh1	*Hhf1,Hht1*
	Swi4, Swi6	*Hta2,Hta3,Htb2*
3	Swi4, Swi6	*Hta1,Htb1*
4	Ndd1, Fkh2, Mcm1	*Cdc20,Clb2,Cln1,Cln2,Cwp1,Exg1,Mnn1,Och1,Rsr1*
	Swi6, Mbp1	*Cdc21,Cdc45,Clb5,Clb6,Dun1,Mcm2,Rad27,Rad51,Rnr1*
	Mcm1	*Cdc46,Cdc6,Mcm3,Mcm6*
	Mcm1	*Far1,Ste2*
	Fkh1	*Tem1*
5	Ndd1, Fkh2, Mcm1	*Ace2,Swi5*
	Ace2, Swi5	*Cts1*
	Swi4, Swi6	*Kre6*
6	Ace2, Swi5	*Egt2*
	Swi4, Swi6	*Hho1*
	Fkh1	*Tel2*
7	Fkh1	*Arp7*
	Swi4, Swi6	*Bud9*

Table 4-13. 44 Yeast gene *k*-means result (gene X keyword matrix as input)

Clusters	Activators	Genes
1	Ndd1, Fkh2, Mcm1	*Ace2, Swi5*
	Ace2, Swi5	*Cts1, Egt2*
	Swi6, Mbp1	*Rad51*
	Fkh1	*Tel2*
2	Swi6, Mbp1	*Cdc21, Cdc45, Mcm2*
	Mcm1	*Cdc46, Mcm3, Mcm6*
	Mcm1	*Ste2*
3	Swi4, Swi6	*Hho1, Hta3*
4	Swi4, Swi6	*Gic1, Gic2*
	Swi6, Mbp1	*Rad27*
5	Swi4, Swi6	*Bud9, Mnn1, Rsr1*
	Swi6, Mbp1	*Rnr1*
6	Swi4, Swi6	*Exg1, Kre6, Och1,*
	Fkh1	*Tem1*
7	Fkh1	*Arp7*
	Swi4, Swi6	*Cwp1, Msb2*
8	Swi6, Mbp1	*Dun1,*
	Fkh1	*Hhf1, Hht1*
	Swi4, Swi6	*Hta1, Hta2, Htb1, Htb2*
9	Ndd1, Fkh2, Mcm1	*Cdc20, Clb2*
	Mcm1	*Cdc6*
	Swi6, Mbp1	*Clb5, Clb6*
	Swi4, Swi6	*Cln1, Cln2*
	Mcm1	*Far1*

Table 4-14. 44 Yeast gene AUTOCLASS result (gene X keyword matrix as input)

Clusters	Activators	Genes
1	Swi4, Swi6	*Cwp1, Exg1, Mnn1, Och1*
	Swi4, Swi6	*Hhf1, Hht1*
	Fkh1	*Hta1, Hta3, Hta2, Htb2, Htb1*
2	Fkh1	*Arp7*
	Swi4, Swi6	*Bud9, Msb2, Rsr1*
	Swi4,Swi6	*Hho1*
	Mcm1	*Mcm3*
3	Ndd1, Fkh2, Mcm1	*Cdc20, Clb2*
	Swi6,Mbp1	*Clb5,Clb6*
	Fkh1	*Tem1*
4	Ndd1, Fkh2, Mcm1	*Ace2,Swi5*
	Swi6,Mbp1	*Cdc21*
	Ace2, Swi5	*Cts1,Egt2*
5	Mcm1	*Cdc6,Mcm6*
	Swi6, Mbp1	*Rad27,Rad51,Mcm2*
6	Swi4,Swi6	*Exg1,Kre6,Mnn1,Och1*
	Mcm1	*Ste2*
7	Swi6,Mbp1	*Cdc45*
	Mcm1	*Cdc46*
	Swi4,Swi6	*Gic1,Gic2*
8	Swi6,Mbp1	*Dun1,Rnr1*
	Fkh1	*Tel2*
9	Swi4,Swi6	*Cln1,Cln2*
	Mcm1	*Far1*

Table 4-15. 44 Yeast-gene *k*-means result (gene X gene matrix as input)

Clusters	Activators	Genes
1	Fkh1	*Hhf1*
	Swi4, Swi6	*Htb2*
2	Fkh1	*Hht1*
	Swi4, Swi6	*Hta1, Hta2, Hta3,Htb1*
3	Swi4, Swi6	*Msb2*
4	Fkh1	*Arp7*
5	Ndd1, Fkh2, Mcm1	*Ace2, Cdc20, Clb2,Swi5*
	Ace2, Swi5	*Cts1*
	Swi6, Mbp1	*Rad51, Clb5, Clb6,Cdc21, Cdc45, Dun1,Mcm2, Rnr1, Rad27*
	Fkh1	*Tel2*
	Mcm1	*Cdc46, Cdc6, Mcm3, Mcm6*
	Mcm1	*Ste2, Far1*
	Swi4, Swi6	*Exg1, Kre6, Och1, Cwp1,Cln1, Cln2,Mnn1, Rsr1*
	Fkh1	*Tem1*
6	Swi4, Swi6	*Gic1,Gic2*
7	Ace2, Swi5	*Egt2*
8	Swi4, Swi6	*Hho1*
9	Swi4, Swi6	*Bud9*

Table 4-16. 44 Yeast-gene Hierarchical clustering result (gene X gene matrix as input)

Clusters	Activators	Genes
1	Swi6, Mbp1	*Clb5, Clb6*
	Ndd1, Fkh2, Mcm1	*Ace2,Clb2,Cdc20,Swi5*
	Swi4, Swi6	*Cln1,Cln2*
	Fkh1	*Tem1*
	Ace2, Swi5	*Cts1,Egt2*
2	Swi6, Mbp1	*Rad51,Dun1, Rnr1, Rad27*
3	Mcm1	*Cdc46, Cdc6, Mcm3, Mcm6*
	Swi6, Mbp1	*Cdc21,Cdc45,Mcm2*
4	Swi4, Swi6	*Bud9,Rsr1,Gic1,Gic2*
	Mcm1	*Far1*
5	Fkh1	*Hhf1,Hht1*
	Swi4, Swi6	*Hta1,Hta2,Hta3,Htb1,Htb2*
6	Fkh1	*Arp7*
7	Swi4, Swi6	*Hho1*
8	Swi4, Swi6	*Mnn1,Och1,Kre6,Msb2*
	Mcm1	*Ste2*
9	Swi4, Swi6	*Exg1,Cwp1*
10	Fkh1	*Tel2*

Table 4-17. 44 Yeast-gene SOM result (gene X gene as input)

Clusters	Activators	Genes
1	Fkh1	***Arp7,Tel2***
	Swi4, Swi6	***Hho1***
2	Ndd1, Fkh2, Mcm1	*Rsr1*
3	Fkh1	*Hht1*
	Swi4, Swi6	*Hta1,Hta2,Hta3,Htb1*
4	Swi4, Swi6	*Bud9,Gic1,Gic2,Msb2*
5	Fkh1	*Hhf1*
	Swi4, Swi6	*Htb2*
6	Ndd1, Fkh2, Mcm1	*Cdc20,Clb2,Cln1,Cln2,Cwp1,Exg1,Mnn1,Och1, Ace2,Swi5*
	Swi6, Mbp1	*Cdc21,Cdc45,Clb5,Clb6,Dun1,Mcm2,Rad27,Rad51,Rnr1*
	Mcm1	*Cdc46,Cdc6,Mcm3,Mcm6*
	Mcm1	*Far1,Ste2*
	Fkh1	*Tem1*
	Swi4, Swi6	*Kre6*
	Ace2, Swi5	*Cts1, Egt2*

Table 4-18. 44 Yeast-gene AUTOCLASS result (gene X gene matrix as input)

Clusters	Activators	Genes
1	Ndd1, Fkh2, Mcm1	*Ace2,Cdc20,Clb2,Swi5*
	Swi6,Mbp1	*Clb5,Clb6,Rnr1*
	Swi4,Swi6	*Cln1,Cln2*
	Ace2, Swi5	*Cts1*
	Mcm1	*Far1*
	Fkh1	*Tem1*
2	Fkh1	*Arp7,Hhf1,Hht1,Tel2*
	Swi4, Swi6	*Hta1,Htb1,Htb2,*
	Swi4, Swi6	*Msb2*
3	Swi6,Mbp1	*Cdc21,Cdc45,Mcm2*
	Mcm1	*Cdc46,Cdc6 ,Mcm3,Mcm6*
4	Swi4, Swi6	*Bud9,Gic1,Gic2,Rsr1*
5	Swi4, Swi6	*Cwp1,Egt2,Exg1,Mnn1*
6	Swi6, Mbp1	*Dun1,Rad27,Rad51*
7	Swi4,Swi6	*Hho1*
	Fkh1	*Hta2,Hta3*
8	Swi4,Swi6	*Kre6,Och1*
	Mcm1	*Ste2*

4.8 Top-scoring keywords shared among members of a gene cluster

Keywords are ranked according to their highest shared z-scores in each cluster. The keyword sharing strength metric (K^a) is defined as the sum of z-scores for a shared keyword *a* within the cluster, multiplied by the number of genes (*M*) within the cluster with which the word is associated; in this calculation z-scores below a user-selected threshold are set to zero and are not counted.

$$K^a = \sum_{g=1}^{M} (Z_g^a) \times \sum_{g=1}^{M} Count(Z_g^a) \qquad (4\text{-}9)$$

Thus, larger values reflect stronger and more extensive keyword associations within a cluster. We identify the 30 highest scoring keywords for each of the four clusters and provide these four lists to approximately 20 students, postdoctoral fellows, and faculty, asking them to guess a major function of the underlying genes that gave rise to the four keyword lists.

Keywords shared among genes (26-gene set) within each cluster are ranked according to a metric based on both the degree of significance (the sum of z-scores for each keyword) and the breadth of distribution (the sum of the number of genes within the cluster for which the keyword has a z-score greater than a selected threshold). This double-pronged metric obviate the difficulty encountered with keywords that have extremely high z-scores for single genes within the cluster but modest z-scores for the remainder. The 30 highest scoring keywords for each of the four clusters are tabulated (Table 4-19).

Table 4-19. Top ranking keywords associated with each gene cluster**.**

Cluster 1 (Catecholamine Biosynthesis)	**Cluster 2** (Tyrosine/Phenylalanine Metabolism)	**Cluster 3** (Cytoplasmic Proteins)	**Cluster 4** (Glutamate Receptors)
mao	mutase	tubulin	ampa
clorgyline	monofunct	dynein	ionotrop
phenylethanolamine	dehydratase	spectrin	kainate
methyltransferase	bifunct	microtubule	glutam
monoamine	phenylalanine	axonem	isoxazole
hydroxylase	tyrosine	axoneme	subunit
deprenyl	phenylpyruv	chlamydomona	glutamaterg
catechol	herbicola	demembran	homomer
dopamine	fluorophenylalanine	flagellar	receptor
oxidase	tryptophan	flagella	methyl
chromaffin	erwinia	cytoskeleton	propion
selegiline	catalyt	isotype	hydroxi
dihydroxyphenyl	brevibacterium	cytoskelet	neuron
catecholamine	substrate	microtubular	domoate
tyrosine	enzyme	protofila	hippocampu
phenylethylamine	dehydrogenase	tetrahymena	gyru
adrenomedullari	decarboxyl	depolymer	hippocamp
dopa	biosynthet	subunit	synapt
tyramine	flavum	isoform	methylisoxazole
medulla	aromat	cilia	hek
pargyline	hcl	polymer	aspart
inhibitor	subtili	sequence	postsynapt
homovanill	ammonium	mutant	cerebellum
catecholaminerg	sulfate	tyrosin	cortex
adren	monom	diverg	isoxazolepropion
enzyme	molecular	kinesin	cyclothiazide
dihydroxyphenylalanine	arg	pvuii	ca
coeruleu	mutant	intron	heteromer
parkinson	nicotinamide	codon	bergmann
moclobemide	subunit	multigene	coloc
noradrenerg	tyr	encod	forebrain
mptp	effector	cytoplasm	purkinje
neuron	inhibitor	physarum	cerebellar

The respective keyword lists appear to be highly informative about the general function of the original, pre-selected clusters. Twenty volunteer faculty, postdoctoral fellows and medical graduate students form a hypothesis of the major function of the genes in each cluster based on the respective keyword lists. Even though this is an informal survey, the finding that a large majority of guesses are accurate (Table 4-20) adds credence to the conclusion that our clustering and keyword lists can be useful in allowing rapid sorting and evaluation of large lists of genes. Hypotheses about the function of the cluster containing tyrosine/phenylalanine synthesis enzymes appeared less accurate than the others, perhaps due to the relative obscurity of this cluster of genes to most of the volunteers.

Table 4-20. Hypotheses on cluster function formed by 10 volunteers presented with keyword lists of Table 4-1.

Cluster 1 (Catecholamine Biosynthesis)	**Cluster 2** (Tyrosine/Phenylalanine Metabolism)	**Cluster 3** (Cytoplasmic Proteins)	**Cluster 4** (Glutamate Receptors)
Catecholamine synthesis	Amino acid synthesis	Cell motility and chemotaxis	Glutamatergic synaptic transmission
Catecholamine synthesis and degradation	amino acid or other metabolic pathways in bacteria	Cell growth, size, shape and motility	Glutamate receptor signaling
Catecholamine metabolism/pathway, related to Parkinson's disease	Antibiotic synthesis and metabolism	Cytoskeletal function, axonal transport	Glutamate receptors, synaptic transmission, postsynaptic function
Neurotransmitter synthesis/release (catecholamine)	Enzyme catalysis and oxidation	Cell movement	Glutamine receptors in the brain
Catecholamine synthesis	Drug metabolism enzymes	Cell motility	Excitatory transmission
depression, schizophrenia	bacterial respiration/metabolism	cell motility and transport	memory/learning
neuronal signaling mediated by monoamines	drug (or endogenous substrate) metabolism	cell structure and/or morphology; cell movement or cell division	fast synaptic transmission mediated by ionotropic glutamate receptors
Dopamine and catecholamine metabolism	drug metabolism	Cytoskeleton organization, dendritic growing; exocytosis	Post synaptic glutamatergic neurotransmission AMPA/KA receptors
Catecholamines synthesis and function; relation in pathological disorders such as Parkinson's disease	Cellular metabolic pathways	Forming cytoskeleton or involved in flagellar movement	AMPA/KA receptor functions and expression in different brain areas
Metabolism, possibly catecholamine metabolism related pathways	Metabolism of amino acids	Cell structure components involved in cell motility and/or cell division	Neuronal receptor function, possibly AMPA glutamate receptor related function

4.9 Discussion of clustering algorithm comparison

4.9.1 BEA-PARTITION vs. k-means

In this chapter, the z-score thresholds are used for keyword selection. When the threshold was 0, all words, including noise (non-informative words and misspelled words), are used to cluster genes. Under the tested conditions, clusters produced by BEA-PARTITION have higher quality than those produced by *k*-means. BEA-PARTITION clusters genes based on their shared keywords. It is unlikely that genes within the same cluster shared the same noisy words with high z-scores, indicating that BEA-PARTITION is less sensitive to noise than *k*-means. In fact, BEA-PARTITION performs better than *k*-means in the two test gene sets under almost all test conditions (Figure 4-2). BEA-PARTITION performs best when z-score thresholds were 10, 15, and 20, which indicated (1) that the words with z-score less than 10 are less informative and (2) few words with z-scores between 10 and 20 are shared by at least two genes and did not improve the cluster quality. When z-score thresholds are high (>30 in the 26-gene set and >20 in the 44-gene set), more informative words are discarded, and as a result, the cluster quality is degraded.

BEA-PARTITION is designed to group cells with larger values together, and the ones with smaller values together. The final order of the genes within the cluster reflects deeper inter-relationships. Among the ten glutamate receptor genes examined, *GluR1*, *GluR2*, *GluR4* are AMPA receptors, while *GluR6*, *KA1* and *KA2* are kainate receptors. The observation that BEA-PARTITION places gene *GluR6* and gene *KA2* next to each other, confirms that the literature associations between *GluR6* and *KA2* are higher than those between *GluR6* and AMPA receptors. Furthermore, the association and

interrelationships of the clustered groups with one another can be seen in the final clustering matrix. For example, TT is an outlier in Figure 3, however, it still have higher affinity to *PD1* (affinity = 202) and *PD2* (affinity = 139) than to any other genes. Thus, TT appears to be strongly related to genes in the tyrosine and phenylalanine synthesis cluster, from which it originates.

BEA-PARTITION has several advantages over the *k*-means algorithm: (1) while *k*-means generally produces a locally optimal clustering (Xu et al., 2003), BEA-PARTITION produces globally optimal clustering by permuting the columns and rows of the symmetric matrix; (2) the *k*-means algorithm is sensitive to initial seed selection and noise (Jain et al., 1999), whereas BEA-PARTITION has no initial seed as input. It first starts off with a symmetric affinity matrix.

4.9.2 BEA-PARTITION vs. hierarchical algorithm

Hierarchical clustering algorithm, as well as *k*-means, and Self-Organizing Maps, have been widely used in microarray expression profile analysis. Hierarchical clustering organizes expression data into a binary tree without providing clear indication of how the hierarchy should be clustered. In practice, investigators define clusters by a manual scan of the genes in each node and rely on their biological expertise to notice shared functional properties of genes. Therefore, the definition of the clusters is subjective, and as a result, different investigators may interpret the same clustering result differently. Some have proposed automatically defining boundaries based on statistical properties of the gene expression profiles; however, the same statistical criteria may not be generally applicable to identify all relevant biological functions (Raychaudhuri et al., 2003). We believe that an algorithm that produces clusters with clear boundaries can provide more objective

results and possibly new discoveries, which are beyond the experts' knowledge. In this report, our results show that BEA-PARTITION can have similar performance as the hierarchical algorithm, and provide distinct cluster boundaries.

4.9.3 K-means vs. SOM

The *k*-means algorithm and SOM can group objects into different clusters and provide clear boundaries. Despite its simplicity and efficiency, the SOM algorithm has several weaknesses that make its theoretical analysis difficult and limit its practical usefulness. Various studies have suggested that it is hard to find any criteria under which the SOM algorithm performs better than the traditional techniques, such as *k*-means [11]. Balakrishnan et al. (1994) compared the SOM algorithm with *k*-means clustering on 108 multivariate normal clustering problems. The results showed that the SOM algorithm performed significantly worse than the *k*-means clustering algorithm. Our results also showed that *k*-means performed better than SOM by generating clusters with higher mutual information.

4.9.4 Computing time and complexity

The computing time or computational complexityof BEA-PARTITION, same as that of hierarchical algorithm and SOM, is in the order of N^2, which means that it grows proportionally to the square of the number of genes and commonly denoted as $O(N^2)$, and that of *k*-means is in the order of N*K*T (O(NKT)), where N is the number of genes tested, K is the number of clusters, and T is the number of improvement steps (iterations) performed by *k*-means. In our study, the number of improvement steps is 1000. Therefore, when the number of genes tested is about 1000, BEA-PARTITION runs (a * K + b) times faster than *k*-means, where a, and b are constants. As long as the number of

genes to be clustered is greater than the product of the number of clusters and the number of iterations, BEA-PARTITION will run faster than *k*-means.

4.9.5 Effect of weighting schemes on clustering results

We have shown, in Chapter 3, that as a weighting scheme, TFIDF outperforms z-score method as measured by precision and recall values. Here, we cluster genes based on the keywords generated by the two weighting schemes, TFIDF, and z-score method. The results are shown in Figure 4-5, and Figure 4-6.

From Figure 4-5, we can see, with TFIDF as the weighting scheme, that BEA-PARTITION algorithm correctly assigns the 26 genes into the right clusters. BEA-PARTITION clusters the 44 yeast genes into 9 groups (Figure 4-6). The 9 clusters generated by BEA-PARTITION using TFIDF-derived keywords have better quality as measured by Mutual information: 0.65 for TFIDF-derived keywords (Table 4-21) vs. 0.60 for z-score-derived keywords (Table 4-3).

	Alpha-Spectrin	Actin	Alpha-Tubulin	Dynein	Beta-tubulin	KA1	NMDA-R1	NMDA-R2A	NMDA-R2B	GluR6	KA2	GluR1	GluR4	GluR2	GluR3	PD1	PD2	CM	TT	PNMT	DBH	TH	DOPA	COM	MOB	MOA
Alpha-Spectrin	1000	47	22	22	24	14	29	24	19	21	22	46	31	23	18	12	5	10	25	11	21	29	35	14	12	11
Actin	47	1000	62	48	49	7	26	23	19	19	18	34	32	21	17	13	6	13	35	11	20	38	28	8	9	8
Alpha-Tubulin	22	62	1001	94	972	13	29	23	19	21	23	28	28	26	20	12	7	14	38	13	19	48	39	8	9	9
Dynein	22	48	94	1001	76	9	31	35	28	17	20	26	26	22	19	25	7	16	21	7	16	21	22	7	8	9
Beta-tubulin	24	49	972	76	1000	13	33	28	23	24	27	33	36	29	20	14	5	13	29	15	20	42	40	9	10	10
KA1	14	7	13	9	13	1000	138	61	54	102	192	266	118	94	65	4	3	8	17	7	14	23	27	9	13	11
NMDA-R1	29	26	29	31	33	138	1000	596	524	130	133	208	159	183	97	14	6	16	38	27	29	73	60	15	20	18
NMDA-R2A	24	23	23	35	28	61	596	999	856	123	137	179	163	135	79	16	13	23	59	16	27	71	50	12	15	13
NMDA-R2B	19	19	19	28	23	54	524	856	999	112	120	160	130	121	78	14	12	19	52	17	22	67	48	16	19	16
GluR6	21	19	21	17	24	102	130	123	112	999	765	197	211	302	142	7	4	10	32	18	22	51	48	14	18	15
KA2	22	18	23	20	27	192	133	137	120	765	999	290	246	278	153	5	4	11	29	14	21	39	39	10	14	12
GluR1	46	34	28	26	33	266	208	179	160	197	290	999	478	497	244	7	5	13	29	23	32	68	52	14	19	16
GluR4	31	32	28	26	36	118	159	163	130	211	246	478	999	582	225	9	7	16	41	24	31	64	61	15	17	14
GluR2	23	21	26	22	29	94	183	135	121	302	278	497	582	999	240	10	6	13	28	24	25	63	52	12	19	16
GluR3	18	17	20	19	20	65	97	79	78	142	153	244	225	240	999	26	4	9	24	19	18	42	39	13	16	13
PD1	12	13	12	25	14	4	14	16	14	7	5	7	9	10	26	1000	680	378	137	10	31	94	62	18	17	18
PD2	5	6	7	7	5	3	6	13	12	4	4	5	7	6	4	680	999	534	103	6	22	74	38	12	15	15
CM	10	13	14	16	13	8	16	23	19	10	11	13	16	13	9	378	534	1000	129	11	30	90	64	17	19	20
TT	25	35	38	21	29	17	38	59	52	32	29	29	41	28	24	137	103	129	1000	32	68	288	135	37	35	33
PNMT	11	11	13	7	15	7	27	16	17	18	14	23	24	24	19	10	6	11	32	1000	114	122	65	47	40	46
DBH	21	20	19	16	20	14	29	27	22	22	21	32	31	25	18	31	22	30	68	114	1001	360	202	81	55	57
TH	29	38	48	21	42	23	73	71	67	51	39	68	64	63	42	94	74	90	288	122	360	1000	340	72	58	54
DOPA	35	28	39	22	40	27	60	50	48	48	39	52	61	52	39	62	38	64	135	65	202	340	1000	213	126	117
COM	14	8	8	7	9	9	15	12	16	14	10	14	15	12	13	18	12	17	37	47	81	72	213	1000	156	144
MOB	12	9	9	8	10	13	20	15	19	18	14	19	17	19	16	17	15	19	35	40	55	58	126	156	999	945
MOA	11	8	9	9	10	11	18	13	16	15	12	16	14	16	13	18	15	20	33	46	57	54	117	144	945	999

Figure 4-5. Gene clusters of 26-genes by keyword associations using BEA-PARTITION (TFIDF-derived keywords).

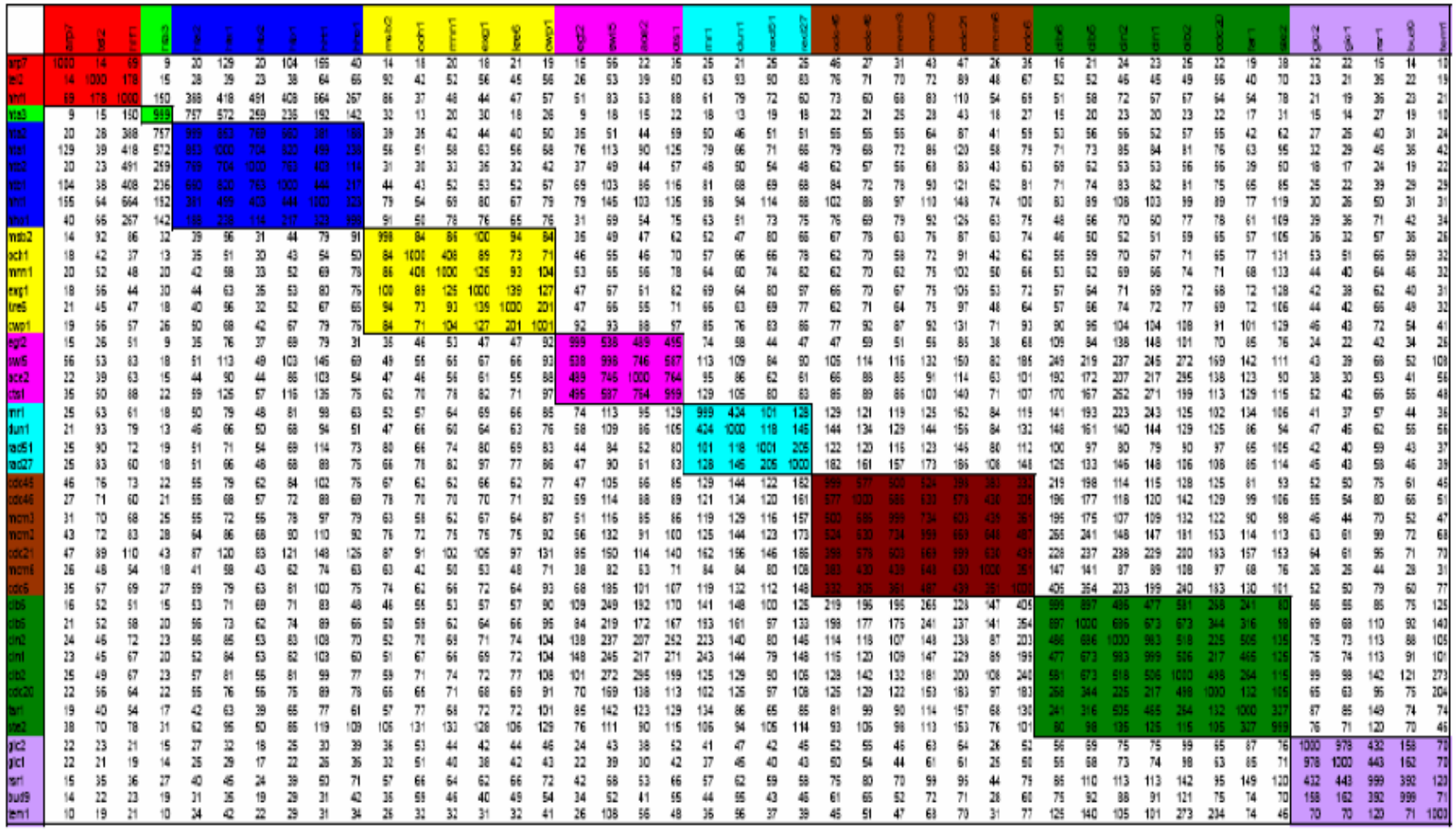


Figure 4-6. Gene clusters of 44-genes by keyword associations using BEA-PARTITION(TFIDF-derived keywords).

Table 4-21 44 Yeast genes BEA-PARTITION result

Clusters	Activators	Genes	Purity	Entropy
1	Fkh1	*Arp7,Tel2,Hhf1*	1	0
2	Swi4, Swi6	*Hta3*	1	0
3	Swi4, Swi6 Fkh1	*Hta1,Hta2,Htb2,Htb1,Hho1* *Hht1*	0.83	0.21
4	Swi4, Swi6	*Msb2,Och1,Mnn1,Exg1,Kre6,Cwp1*	1	0
5	Ace2, Swi5 Ndd1,Fkh2,Mcm1	*Egt2,Cts1* *Ace2,Swi5*	0.5	0.32
6	Swi6, Mbp1	*Rnr1,Dun1,Rad51,Rad27*	1	0
7	Swi6, Mbp1 Mcm1	*Cdc45,Mcm2,Cdc21* *Cdc46,Mcm3,Mcm6,Cdc6*	0.57	0.31
8	Swi4, Swi6 Swi6, Mbp1 Ndd1,Fkh2,Mcm1 Mcm1	*Cln1,Cln2* *Clb5,Clb6* *Clb2,Cdc20* *Far1,Ste2*	0.25	0.63
9	Swi4, Swi6 Fkh1	*Gic2,Gic1,Rsr1,Bud9* *Tem1*	0.8	0.23
Average Purity			0.77	
Average Entropy				0.19
Normalized Mutual Information				0.65

4.10 Future experiment

The Genetrek algorithms can find new gene-to-gene relationship, such as the relationship between Exg1, Cwp1, Mnn1, and Och1. The shared keyword lists reveal that these four genes have something to do with polysaccharide metabolism. The future experiment we can do is to use the shared keywords as input to the Genetrek algorithms to see if the word "polysaccharide" can be a highly ranked keyword. This process could be an iterative process.

4.11 Summary

In this chapter, we cluster the genes by shared functional keywords. We describe an approach BEA-PARTITION that applies an algorithm called the Bond Energy Algorithm (BEA) (McCormick et al., 1972; Navathe et al., 1984) for functional gene clustering based on keyword association. We have developed our own criteria for affinity computation at the boundaries and for splitting matrix at each step of the algorithm. We also compare the performance of BEA-PARTITION, hierarchical clustering algorithm, self-organizing map, and the *k*-means algorithm for clustering functionally-related genes based on shared keywords, using purity, entropy, and mutual information as metrics for evaluating cluster quality. The results show that BEA-PARTITION outperforms the other popular clustering algorithms. The BEA-PARTITION algorithm represents our extension to the BEA approach specifically for dealing with the problem of discovering functional similarity among genes based on functional keywords extracted from literature. We believe that this important clustering technique has promise for application to other bioinformatics problems where starting matrices are available from experimental observations.

CHAPTER 5

YEAST GENE FUNCTION PREDICTION FROM DIFFERENT DATA SOURCES

The field of functional genomics studies gene function on a large scale by conducting parallel analysis of gene expression for a large number of genes (Strachan and Read, 1999; Hvidsten and Komorowski, 2001). This research is a natural successor to the genome sequencing efforts such as, for example, the Human Genome Project, and is made possible by the DNA microarrays. Such arrays, which allow researchers to simultaneously measure the expression levels of thousands of different genes, produce overwhelming amounts of data. In response, much recent research has been concerned with automating the analysis of microarray data. Current approaches mainly concentrate on applying clustering techniques to the expression data, in order to find clusters of genes demonstrating similar expression patterns. The assumption motivating such search for co-expressed genes is that simultaneously expressed genes often share a common function.

Different data sources can be used to predict gene function. High-throughput gene and protein assays give a view into the organization of molecular cellular life through quantitative measurements of gene expression levels (Hvidsten and Komorowski, 2001). Increasing quantities of high-throughput biological data have become available to assess functional relationships between gene products on a large scale.

First, gene function can be inferred from DNA microarray expression data. The representation of DNA microarray results is based on the assumption that genes with similar functions have similar expression profiles in cells. This is utilized by inductive

learning methods that predict the function of genes that have an unknown function (unknown genes), from their expression-similarity with genes with a known function (known genes). Currently, techniques pursued for microarray data analysis concentrate on applying clustering methods directly on the expression data. However, cluster analysis alone cannot fully address the issue of gene function prediction (Shatkay et al., 2000). Furthermore, many high-throughput methods sacrifice specificity for scale. Whereas gene co-expression data are an excellent tool for hypothesis generation, microarray data alone often lack the degree of specificity needed for accurate gene function prediction (Troyanskaya et al., 2003). Furthermore, genes that are functionally related may demonstrate strong anti-correlation in their expression levels, (a gene may be strongly suppressed to allow another to be expressed).

Secondly, gene function can be inferred from phylogenetic profiles. The complete genomic sequences of human and other species provide a tremendous opportunity for understanding the functions of biological macromolecules (Pavlidis et al., 2002). Phylogenetic profiles are derived from a comparison between a given gene and a collection of complete genomes. Each profile characterizes the evolutionary history of a given gene. There is evidence that two genes with similar phylogenetic profiles may have similar functions, the idea being that their similar pattern of inheritance across species is the result of a functional link (Pellegrini et al., 1999).

Finally, an important data source that can be used to infer the gene function is the scientific literature. The function of many genes has been described either in explicit terms, or in indirect ways in the literature. By relating documents that report about well understood genes to documents discussing other genes, we can predict, detect, and

explain the functional relationships between the genes that are involved in large-scale experiments. A number of groups are developing algorithms that link information from medical literature with gene names. Andrade and Valencia (1998) introduced a statistical profiling strategy that accepts user-supplied abstracts related to a protein of interest and returns an ordered set of keywords that occur in those abstracts more often than would be expected by chance. With the goal of automating the functional annotation of new proteins, Andrade et al. (1999) presented an interactive suite of programs called "Genequiz", which searches and organizes information from many sequence and text databases. Andrade and Bork (2000) and Perez-Iratxeta et al. (2002) developed a program that links the OMIM database of human inherited diseases to keywords derived from MEDLINE. A variety of nonstatistical approaches have also been used to organize genes. The web tool, PubGene, finds links between pairs of genes based on their co-occurrence in MEDLINE abstracts (Jenssen and Vinterbo, 2000; Jenssen et al., 2001). Another approach (Masys et al., 2001), the basis of the HAPI web tool, organizes gene names according to predefined hierarchical classification systems of enzymes and diseases, and includes hyperlinks to specific MEDLINE citations responsible for the individual classifications. Still another approach (Tanabe et al., 1999), used by the MedMiner system, automatically retrieves functional information (both keywords and gene names related to a user-defined function) from the GeneCards database, and configures it for a PubMed search. The algorithm presents the results by the specific sentence containing the information rather than by the title, speeding review of the results if the user prefers to extract relevant sentences rather than scan through the whole abstract text. A similar method of presenting the statistically most significant sentence was used by Andrade and

Valencia (1998), which we will also incorporate into our data displays. In the previous chapters, we reported on the system that we have developed a system to retrieve functional keywords automatically from biomedical literature for each gene, and then cluster the genes by shared functional keywords (Chapter 4). Using a similarity-based search in document space, Shatkay et al. (2000) developed an approach for utilizing literature to establish functional relationships among genes on a genome-wide scale.

In this chapter, we performed a comparative study for functional classification of *Saccaromyces cerevisiae* (budding yeast) genes from different data sources. Data from three different types of sources were compared: microarray data, phylogenetic profile data, and biomedical literature data. The goal was to determine the relative effectiveness or usefulness of this data in terms of prediction of gene function.

5.1 Data sources.

1. The first data set derives from a collection of DNA microarray hybridization experiments (Eisen et al., 1998). Each data point represents the logarithm of the ratio of expression levels of a particular gene under two different experimental conditions. The data consists of a set of 79-element gene expression vectors for 2,465 yeast genes. These genes were selected by Eisen *et al.* [16] based on the availability of accurate functional annotations. The data were generated from spotted arrays using samples collected at various time points during the diauxic shift (DeRisi et al., 1997), the mitotic cell division cycle (Spellman et al., 1998), sporulation (Chu et al., 1998) and temperature and reducing shocks. The feature values are the 79 tested conditions, such as diauxic shift, the mitotic cell division cycle, sporulation, and temperature and reducing shocks.

2. In addition to the microarray expression data, each of the 2,465 yeast genes is characterized by a phylogenetic profile (Pellegrini et al., 1999). In its simplest form, a phylogenetic profile is a bit string, in which the Boolean value of each bit indicates whether the gene of interest has a close homolog in the corresponding genome. If no homolog is found, the bit value is zero. The profiles employed in this chapter contain, at each position, the negative logarithm of the lowest E-value reported by BLAST version 2.0 (Alschul et al., 1997) in a search against a complete genome, with negative values (corresponding to E-values greater than 1) truncated to 0. Two genes in an organism can have similar phylogenetic profiles for one of two reasons. First, genes with a high level of sequence similarity will have, by definition, similar phylogenetic profiles. Second, for two genes which lack sequence similarity, the similarity in phylogenetic profiles reflects a similar pattern of occurrence of their homologs across species. This coupled inheritance may indicate a functional link between the genes, based on the hypothesis that the genes are always present together or always both absent because they cannot function independently of one another. The feature values are the Boolean values which show if the gene has close homologs with the known genomes.

3. Classification experiments were carried out using gene functional categories from the Munich Information Center for Protein Sequences Yeast Genome Database (MYGD) (*http://mips.gsf.de/genre/proj/yeast/index.jsp*). The database contains several hundred functional classes, whose definitions come from biochemical and genetic studies of gene function. For each of the 2,465 genes, the abstracts used to curate the MYGD were extracted and combined into one document per gene. Abstracts may occur in more than one document if they refer to multiple genes. All the documents form a document

database. Since each document represents one gene, we use the words *document* and *gene* interchangeably.

Background Sets: A background set is a set of abstracts used as the baseline to measure relative frequency of words in other documents (see Chapter 3). The background sets of abstracts were used to build a hash table of words and their respective statistics for comparison with the corresponding words in the test gene sets. We built a large background set, which incorporated all the MEDLINE abstracts (about 5.6 Million abstracts) up to year 2000.

Text analysis. The abstracts in each document were tokenized into single words, stemmed by Porter's stemming algorithm, and filtered by a stop list (Liu et al., 2004b). The standard term frequency-inverse document frequency (TFIDF) function was used (Salton and Buckley, 1988) to assign the weight to each word in the document. TFIDF combines term frequency (TF), which measures the number of times a word occurs in the gene's documents (reflecting the importance of the word to the gene), and inverse document frequency (IDF), which measures the information content of a word – its rarity across all the abstracts in the background set (See Chapter 3 for more details).

TFIDF vector: Each document, which corresponded to one gene, in the document database was modeled as an M-dimensional TFIDF vector, where M is the number of distinct words in the database. Formally, a document was a vector ($tfidf_1$, $tfidf_2$, … , $tfidf_M$), where $tfidf_i$ is the *tfidf* value of word i.

Prior to learning, the gene expression, phylogenetic profile, and text TFIDF vectors are adjusted to have a mean of 0 and a variance of 1. The gene expression and

phylogenetic profile data were from http://*www.cs.columbia.edu/compbio* (Pavlidis et al., 2002).

5.2 Design of a classifier to classify genes by function

In this study, Support Vector Machine (SVM) was used for gene function classification.

In this study, SVMLight v.3.5 was used (Joachims, 1998). Linear kernel and polynomial kernel are applied.

5.3 Cross-validation of the models

The normal method to evaluate the classification results is to perform cross-validation on the classification algorithms (Tan and Gilbert, 2003). Tenfold cross-validation has been shown to be statistically good enough in evaluating the classification performance (Witten and Frank, 1999). In this study, each of the three data sets (microarray, phylogenetic, and text data sets) was partitioned into ten subsets with both positive and negative genes spread as equally as possible between the sets. Each of these sets in turn was set aside while a model was built using the other nine sets. This model was then used to classify the genes in the tenth set, and the accuracy computed by comparing these predictions with the actual category. This process was repeated ten times and the results averaged (Bahler et al., 2000).

5.4 Feature selection

The feature selection method we used in this study is MIT correlation, which is also known as the signal-to-noise statistic (Golub et al., 1999) that helps to eliminate the "noisy" features. For a given feature *i*, we compute the mean and standard deviation of

that feature across the positive examples (μ_i^+ and σ_i^+, respectively) and across the negative examples (μ_i^- and σ_i^-, respectively). The MIT correlation score is defined as MIT (i) = $\frac{|\mu_i^+ - \mu_i^-|}{\sigma_i^+ + \sigma_i^-}$. When making selection, we simply take those features with the highest scores as the most discriminatory features.

5.5 Performance Measures

Several statistics were used as performance measures:

(1). Accuracy: the proportion of correctly classified instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where true positives (TP) denote the correct classifications of positive examples; true negatives (TN) are the correct classifications of negative examples; false positives (FP) represent the incorrect classification of negative examples into the positive class; and false negatives (FN) are the positive examples incorrectly classified into the negative class.

(2). Sensitivity: (known as recall in information retrieval literature) the percent of positive examples which were correctly classified;

$$Sensitivity = \frac{TP}{TP + FN}$$

(3). Specificity: the percent of negative examples which were correctly classified;

$$Specificity = \frac{TN}{TN + FP}$$

(4). Positive Predictive Value (PPV): (known as precision in information retrieval literature) the percentage of the examples predicted to be positive that were correct;

$$PPV = \frac{TP}{TP + FP}$$

(5). Negative Predictive Value (NPV): the percentage of the examples predicted to be negative that were in fact negative.

$$NPV = \frac{TN}{TN + FN}$$

Paired t-tests were performed to evaluate whether the results obtained from microarray data, phylogenetic data, and text data were significantly different from each other.

5.6 Gene function categories tested

The database contains different functional classes, whose definitions come from biochemical and genetic studies of gene function. The experiments reported here use classes containing 400 or more genes available in the MYGD data set as of July 30th, 2004, amounting to 4 functional categories (Table 1). Categories with less than 400 genes are not analyzed. For each class, a support vector machine is trained to discriminate between class members and nonmembers.

A primary goal in biology is to understand the molecular machinery of the cell. The sequencing projects provide us one view of this machinery. A complementary view is provided by data from microarray hybridization experiments. High-throughput techniques, such as DNA microarray and sequencing, accompanied by an increase in the number of publications discussing gene-related discovery, provide the researchers great resources to understand the gene function better. In this chapter, we predicted yeast gene functions from different data sources. MYGD database categorizes the yeast genes into

different categories, of which we analyzed four (category numbers 1, 11, 14, and 20) that amount to more than 400 genes per category.

Table 5-1. The gene function categories studied in this study

Function Category	Function	Number of genes
1	Metabolism	636
11	Transcription	556
14	Protein fate(folding, modification, destination)	449
20	Cellular Transport, Transport Facilitation and Transport Routes	441

5.7 Gene function prediction

The results of gene function prediction from different data sources are shown in Table 2. When microarry data is used and linear kernel was applied for gene function prediction, all the genes in each category were mis-classified (true positive = 0), which can be observed, in Table 2, that the sensitivity values are 0's. Similar results can be observed when phylogenetic data is used and linear kernel was applied to predict gene function except for category #1. When linear kernel is applied and text data was used, the results derived from text data significantly outperforms those derived from microarray data and phylogenetic data ($p<0.01$). SVM can correctly predict the function of the genes in category #20 with an accuracy of 0.927 and a sensitivity of 0.669.

When polynomial kernel is applied, the results derived from text data outperform those derived from microarray data and phylogenetic data ($p < 0.05$) except category #1. No significant difference is observed between the gene function prediction results derived from microarray data and phylogenetic data ($p > 0.05$).

For text data, linear kernel outperforms polynomial kernel ($p<0.01$) as measured by sensitivity, PPV, and accuracy. Polynomial kernel works significantly better than linear kernel ($p<0.01$) for microarry data, and phylogenetic data.

Our gene function prediction by text data strategy is similar to the document categorization in information retrieval. In our case, each document is the collection of abstracts which are related to a specific gene. Document categorization, defined as classifying documents into categories according to their topics or main contents in a supervised manner, organizes large amounts of information into a small number of meaningful categories and improves the information retrieval performance either via term-weighting, or query expansion.

Table 5-2. Gene function prediction by SVM using different data sources.

Kernel Type	Function Category	Microarray Data					Phylogenetic Data					Text Data				
		Sn	Sp	PPV	NPV	Accuracy	Sn	Sp	PPV	NPV	Accuracy	Sn	Sp	PPV	NPV	Accuracy
Linear Kernel	1	0	1	-	0.740 (0.034)	0.739 (0.034)	0.238 (0.060)	0.986 (0.009)	0.718 (0.081)	0.782 (0.038)	0.777 (0.037)	0.462 (0.067)	0.975 (0.013)	0.876 (0.058)	0.837 (0.028)	0.839 (0.023)
	11	0	1	-	0.774 (0.003)	0.774 (0.003)	0	1	-	0.774 (0.003)	0.774 (0.003)	0.500 (0.034)	0.981 (0.006)	0.884 (0.034)	0.871 (0.010)	0.872 (0.012)
	14	0	1	-	0.818 (0.002)	0.818 (0.02)	0	1	-	0.818 (0.002)	0.818 (0.02)	0.490 (0.092)	0.986 (0.007)	0.890 (0.052)	0.897 (0.016)	0.896 (0.018)
	20	0	1	-	0.821 (0.009)	0.821 (0.009)	0	1	-	0.821 (0.009)	0.821 (0.009)	0.669 (0.053)	0.984 (0.008)	0.906 (0.044)	0.932 (0.012)	0.927 (0.012)
Polynomial Kernel	1	0.193 (0.089)	0.967 (0.011)	0.651 (0.107)	0.772 (0.046)	0.764 (0.046)	0.356 (0.040)	0.969 (0.013)	0.807 (0.056)	0.809 (0.034)	0.808 (0.033)	0.133 (0.053)	0.993 (0.007)	0.901 (0.109)	0.764 (0.033)	0.768 (0.032)
	11	0.133 (0.056)	0.970 (0.017)	0.580 (0.160)	0.794 (0.011)	0.781 (0.018)	0.096 (0.028)	0.982 (0.012)	0.640 (0.141)	0.789 (0.007)	0.782 (0.010)	0.159 (0.062)	0.997 (0.002)	0.950 (0.051)	0.803 (0.012)	0.808 (0.014)
	14	0.109 (0.066)	0.997 (0.004)	0.874 (0.137)	0.834 (0.010)	0.835 (0.012)	0.020 (0.016)	0.998 (0.003)	0.533 (0.428)	0.820 (0.002)	0.819 (0.002)	0.209 (0.088)	0.994 (0.004)	0.903 (0.080)	0.850 (0.015)	0.851 (0.016)
	20	0.002 (0.007)	0.999 (0.002)	0.1 (0.335)	0.821 (0.009)	0.820 (0.009)	0.121 (0.040)	0.995 (0.003)	0.855 (0.120)	0.839 (0.010)	0.839 (0.010)	0.348 (0.063)	0.995 (0.004)	0.950 (0.042)	0.876 (0.015)	0.880 (0.016)

Each value is an average value over ten-fold cross-validation. Values in the brackets are the standard errors.
Sn: Sensitivity; Sp: Specificity; PPV: Positive Predictive Value; NPV: Negative Predictive Value.

The results show that, using SVM as the classifier, text data can provide better prediction results than microarray data and phylogenetic data, particularly when linear kernel is applied (Table 5-2) as measured by sensitivity, PPV, NPV and accuracy. These results confirm that the MYGD classifications we tested are not learnable from either microarray data or phylogenetic data (Pavlisis et al., 2002). Pavlidis et al. (2002) pointed out that the failure to predict the gene functions from microarray data or phylogenetic data was not a failure of the SVM model. Rather, for many functional categories, the data are simply not informative. The microarray data is only informative if the genes in the category are coordinately regulated at the level of transcription under the condition tested. Similarly, phylogenetic data are limited in resolution in part because relatively few genomes are available. In particular, among the genomes from which phylogenetic profiles were derived, all but one is bacterial. Thus it is difficult to generate useful phylogenetic profiles for genes that are specific to eukaryotes.

One complement data source we can use to predict gene functions is literature data. With the advancement of genome sequencing techniques comes an overwhelming increase in the amount of literature discussing the discovered genes. As an illustrative example, the number of PubMed documents containing the word *gene* published between the years 1970-1980 is a little over 35,000, while the number of such documents published between the years 1990-2000 is 402,700 – over a ten fold increase. The gene functions have been described in the literature. Therefore, we believe that the gene functions can be predicted by revealing coherent themes within the literature. Content-based relationships among abstracts are then translated into functional connections among genes. We develop a system to retrieve functional keywords automatically from

biomedical literature for each gene, and then cluster the genes by shared functional keywords (Chapter 3 and Chapter 4). The keywords extracted by the system revealed a wealth of potential functional concepts, which were not represented in existing public databases (Chapter 3). The system also clustered the genes into appropriate functional groups based on the functional keyword association (Chapter 4).

In this application, accuracy is not a good performance evaluation metric. When microarry data is used and linear kernel was applied for gene function prediction, all the genes in each category were mis-classified (true positive = 0), which can be observed, in Table 2, that the sensitivity values are 0's. Similar results can be observed when phylogenetic data is used and linear kernel was applied to predict gene function except for category #1. But the accuracy is still over 0.70. This is because the number of positive instances is much smaller than the number of negative instances. Therefore, sensitivity is an appropriate measure in this application.

5.8 Feature selection effect on gene function prediction

In this study, the MIT feature selection metric is used as the feature selection method to test if feature selection can improve SVM performance on gene function prediction using text data. Linear kernel is applied. Figure 5-1 shows the effect of feature selection on SVM performance. From Figure 5-1, we can observe that SVM does not benefit from feature selection. Highest sensitivity, accuracy, PPV, and NPV are obtained when all the features (21,457) are used.

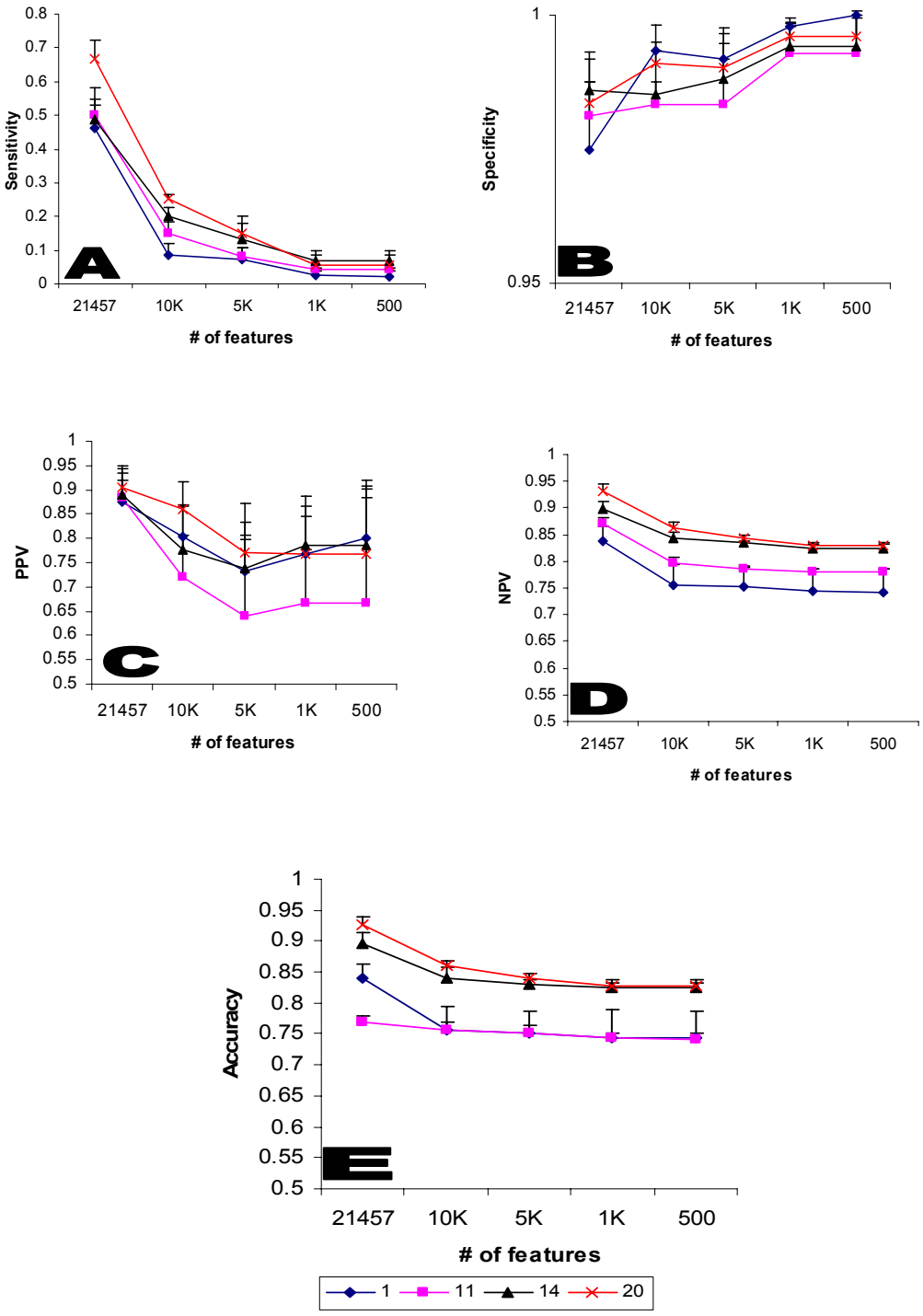


Figure 5-1. Effect of feature selection in combination of SVM classifier on sensitivity (A), specificity (B), PPV (C), NPV (D), and accuracy (E) of different functional categories tested (categories 1, 11, 14, and 20). Note the different scales on the vertical axes. The horizontal axes refer to the number of features used by SVM to classify the genes. Error bars indicated the standard errors.

The results of the experiments indicate that SVM does not benefit from feature selection, which has been reported in text classification. The best results were obtained when all the features were provided to the SVM (Yang and Pederson, 1997; Rogati and Yang, 2002; Brank et al., 2002). Taira and Haruno (1999) compared SVM and decision tree in text categorization, and the best average performance was achieved when all the features were given to SVM, which was a distinct characteristic of SVM compared with the decision tree learning algorithm. Joachims (1998) argued that, in text classification, feature selection was often not needed for SVM, as SVM tends to be fairly robust to overfitting and can scale up to considerable dimensionalities.

5.9 Future experiment

The assumption to use phylogenetic profile to classify gene function is that genes which have similar function are likely to evolve in a correlated fashion. However, the results from this thesis show that phylogenetic profile does not perform well in terms of yeast gene function classification. One future experiment is to find the genes that are highly correlated in the phylogenetic profile sense, but are not functionally correlated.

5.10 Summary

The results in this chapter show a rather counter-intuitive result that the literature text data can provide more accurate prediction results over microarray and phylogenetic data in case of the MYGD database containing all the genes of yeast whose function is already known. It establishes, albeit not in a very rigorous scientific manner, one's belief that text mining has the potential for discovering relationship among genes that have been little to not discovered or reported.

CHAPTER 6

BIOMEDICAL LITERATURE CLASSIFICATION USING SUPPORT VECTOR MACHINES

PubMed (Medline) is a large repository of publicly available scientific literature. Currently, new data is being added to it at the rate of over 1500 abstracts per week. Most biomedical researchers want to access PubMed with specific goals based on their areas of interest. The ability to efficiently review the available literature is essential for rapid progress of research in the scientific community, and particularly so in the biological community where the onslaught of new data is increasing at a phenomenal rate.

Traditional literature database search involves the use of simple Boolean queries, formulated using certain frequently used functionally important keywords the researcher is familiar with, followed by manual scanning of the retrieved records for relevance, which is time consuming, incomplete and error prone. Even with the formulation of complex queries, by a researcher over several years by continually adding new keywords encountered to the original query, the increase in the sensitivity of the searches is only marginal. Therefore, there is a pressing need for the development of automated literature mining techniques that can help the researchers to effectively harvest the heap of the knowledge available in the scientific literature.

Supervised algorithms such as Support Vector Machines (SVM) can be used for classification of biomedical literature into user defined categories. SVM is a machine learning algorithm that performs binary and multiway classification (pattern recognition) of the data into user defined categories (Vladimir and Vapnik, 1995). Support Vector

Machines map non-linearly separable training vectors in input space to linearly separable higher dimensional feature space and find a separating hyper plane with maximal margin in that higher dimensional space. We surveyed this technique in our survey of clustering algorithms in Chapter 2.

SVM has been widely used in text classification. The SVM method has been introduced in text classification by Joachims (1998) and subsequently used in other applications (Dumais et al., 1998; Drucker and Vapnik, 1999; Taira and Haruno, 1999; Dumais and Chen, 2000; Klinkenberg and Joachims, 2000; Yang and Liu, 1999; Tong and Koller, 2000; Joachims, 2002). Joachims (1998) applied SVM to text classification and reported that SVM yielded lower error than many other classification techniques. Yang and Liu (1999) compared different classifiers, Naive Bayes (NB), kNN, and SVM and found that SVM performed at least as well as all other classifiers they tried. Dumais et al. (1998) tested a novel algorithm for training SVM text classifiers and showed that this brings about training speeds comparable to computationally easy methods such as Rocchio. Han et al. (2003) applied SVM for automatically extracting Medline citations of biomedical articles and reranking them according to their relevance to a certain biomedical property difficult to express as PubMed query. They reported that major improvements were achieved in reranking citations with respect to protein disorder-function relationships where the average relative ranking of a relevant citation was improved significantly.

In this study, we report the results of application of SVM for incorporation of Human Genome Epidemiology (HuGE) relevant articles from PubMed database into the Center for Disease Control and Prevention's (CDC) Human Genome Epidemiology

Network, or HuGENet™ (http://www.cdc.gov/genomics/hugenet/) published literature database. Although the present study is limited to classifying the epidemiology related articles, the method described here has a wider applicability and can be used for classifying the articles by disease, by topic or even by domain of expertise.

6.1 Human screening of PubMed

New abstracts appearing in the PubMed database are currently being manually processed and some of them are categorized as HuGE and populated into the CDC's HuGENet™ database by a human expert using a complex search query (Figure 6-1). The complex query CDC uses for screening the PubMed database was developed over four years by iteratively adding the new HuGE relevant keywords encountered that were absent in the original query. As of March'2004 it consisted of 98 different keywords combined with boolean operators. An important point to note here is that after manual processing by human expert, on average, only 5 - 10% of the articles retrieved from the PubMed database by the complex query are HuGE relevant and are being added to the HuGENet^{TM} database (Figure 6-1).

((((((((((((((((((genetic[All Fields] AND (((((("disease"[MeSH Terms] OR ("disease susceptibility"[MeSH Terms] OR predisposition[Text Word])) OR disease[Text Word]) OR defect[Text Word]) OR susceptibility[Text Word]) OR ("counseling"[MeSH Terms] OR counseling[Text Word]))) OR (("disease susceptibility"[MeSH Terms] OR susceptibility[Text Word]) AND (("genes"[MeSH Terms] OR gene[Text Word]) OR ("genes"[MeSH Terms] OR genes[Text Word])))) OR ((((("mutation"[MeSH Terms] OR mutation[Text Word]) OR (("genes"[MeSH Terms] OR gene[Text Word]) AND ("mutation"[MeSH Terms] OR mutation[Text Word]))) OR (("mutation"[MeSH Terms] OR mutations[Text Word]) AND ("genes"[MeSH Terms] OR gene[Text Word]))) OR (("mutation"[MeSH Terms] OR mutation[Text Word]) AND ("genes"[MeSH Terms] OR gene[Text Word])))) OR ("hereditary diseases"[MeSH Terms] OR genetic disorder[Text Word])) OR (genetic[All Fields] AND ((("TEST"[Substance Name] OR ("TEST"[Substance Name] OR test[Text Word])) OR ("research design"[MeSH Terms] OR testing[Text Word])) OR study[All Fields]))) OR ("genetic screening"[MeSH Terms] OR genetic screening[Text Word])) OR (genetic[All Fields] AND ("risk"[MeSH Terms] OR risk[Text Word]))) OR ("polymorphism (genetics)"[MeSH Terms] OR ("polymorphism (genetics)"[MeSH Terms] OR polymorphism[Text Word]))) OR ((((("genotype"[MeSH Terms] OR ("genotype"[MeSH Terms] OR genotype[Text Word])) OR genotyping[All Fields]) OR ("haplotypes"[MeSH Terms] OR haplotype[Text Word])) OR ("haplotypes"[MeSH Terms] OR haplotypes[Text Word]))) OR ((("genome"[MeSH Terms] OR genome[Text Word]) OR genomic[All Fields]) OR ("Genomics"[MeSH Terms] OR genomics[Text Word]))) OR (((gene-environment) OR (gene AND environment)) AND interaction[Text Word])) OR (((genetic[Text Word] OR gene[Text Word]) OR allelic[All Fields]) AND ((variant[All Fields] OR variants[All Fields]) OR (("epidemiology"[MeSH Subheading] OR "epidemiology"[MeSH Terms]) OR frequency[Text Word])))) OR (variant[All Fields] AND (("alleles"[MeSH Terms] OR allele[Text Word]) OR ("alleles"[MeSH Terms] OR alleles[Text Word])))) OR ("heterozygote detection"[MeSH Terms] OR Heterozygote Detection[Text Word])) OR ((Neonatal[All Fields] OR ("infant, newborn"[MeSH Terms] OR newborn[Text Word])) AND (("diagnosis"[MeSH Subheading] OR "mass screening"[MeSH Terms]) OR Screening[Text Word]))) OR germline[All Fields]) OR somatic[All Fields]) OR ("human genome project"[MeSH Terms] OR human genome project[Text Word])) AND ((((((((((((((((((("epidemiology"[Subheading] OR "epidemiology"[MeSH Terms]) OR epidemiology[Text Word]) OR ("public health"[MeSH Terms] OR public health[Text Word])) OR ((("alleles"[MeSH Terms] OR allele[Text Word]) OR allelic[All Fields]) AND ((("epidemiology"[MeSH Subheading]

Figure 6-1 The complex query the CDC currently uses for screening the PubMed database

1848 Total number of articles captured by complex query

1544 Excluded based on reading titles

304 Selected for further reading based on reading titles

Manual Reading of full abstract of the above selected 304 articles gives following:

174 HuGE articles – included in HuGENet database

130 NonHuGE articles – Not included in HuGENet database

Figure 6-2. Distribution of PubMed articles retrieved using complex query: Weekly PubMed update of April 1, 2004

6.2 Text analysis for keyword extraction

As discussed in Chapter 3, the keyword extraction is one of the most important steps in text mining. In text classification, keywords are used as features to describe each abstract. The list of keywords, along with the weights, forms a feature vector to represent the abstracts in the training set and testing set.

In this chapter, the keywords were generated using two different weighting schemes, Z-Score and TFIDF as mentioned in Chapter 3. The weighting schemes estimate the significance of words by comparing the frequency of words in a test set (HuGE) of abstracts with their frequency in a background set of abstracts. The background sets of abstracts were used to build a hash table of words and their respective statistics for comparison with the corresponding words in the training and test sets. The abstracts present in the PubMed database from 1969 till 2004 were used as the background set at the suggestion of CDC. Porter stemming algorithm (Porter, 1980) was used to truncate suffixes and trailing numerals so that words having the same root (e.g., epidemic, epidemics, epidemiology, epidemiological) are collapsed to the same stem "epidem" for frequency counting. The stop word list customized in the previous study (see Chapter 3) was used to filter out non-scientific English words that carry low domain-specific information content.

6.3 SVM model design for text classification

The keywords extracted using Z-Score and TFIDF weighting schemes were selected as features for the Support Vector Machines.

Eight different top ranked sets of keywords with varying number of keywords were used as features for SVM. They were:

1. Z-Score top 100 keywords;
2. Z-Score top 500 keywords;
3. Z-Score top 784 keywords;
4. TFIDF top 100 keywords;
5. TFIDF top 500 keywords;
6. TFIDF top 750 keywords;
7. TFIDF top 1010 keywords;
8. TFIDF top 2010 keywords.

The training and test sets were converted into an abstract X keyword matrix, a format readable by SVM^{light} software (Joachims, 1998). In conversion of the abstracts in the training set into an abstract vs keyword matrix, +1 was used to denote the class label for positive (HuGE) abstracts and –l was used to denote the class label for negative abstracts (Non HuGE). The abstracts in the test sets were also converted to the similar format except for the class label, which is '0' for all the abstracts. Unless otherwise mentioned the SVM was tested with linear kernel.

6.3.1 Training set

The 11000 abstracts present in the CDC's HuGENet™ database, as of March 2004, were used as the positive training set. The Non-HuGE abstracts were obtained by searching the PubMed database using the complex query for the abstracts that appeared in it between 2000 and 2004 followed by removing the HuGE abstracts from them. A total of 11000 abstracts were then randomly selected from the Non HuGE abstracts and were used as the negative training set for the SVM. Two sets of training sets were compared, one consisting of equal number of positive and negative abstracts (11000 positives and 11000 negatives) and the other consisting of twice the number of positives over negative abstracts (11000 positives and 5600 negatives).

6.3.2 Test Set

The abstracts retrieved from the PubMed database using the complex query during four different weeks, February 12' 2004, April 1' 2004, April 8' 2004 and Jun 3' 2004 were used as the test sets for the SVM.

6.3.3 SVM performance measure

Three different metrics were used to evaluate the performance of SVM in classifying the abstracts. The classification of the abstracts by human expert was used as the "gold standard" against which the SVM classifications were evaluated by Sensitivity, Specificity, and Positive Predictive Value (PPV). These evaluation metrics were also mentioned in Chapter 5. The users, the researchers in CDC, expect SVM identifies all the positive articles. They are fine with a lot of false positives. Therefore, sensitivity is a better measure metric than other metrics. In the performance analysis, we pay more

attention to sensitivity. Without sacrificing the sensitivity, we also try to improve specificity.

6.4 Effect of different sets of keywords on SVM performance

As mentioned in section 6.3, we selected eight different top ranked sets of keywords with varying number of keywords were used as features to represent the abstracts in the training set and testing set. To find out which set of keyword should be used to train SVM, we evaluated the effect of different sets of keywords on SVM performance. We focused more on sensitivity (section 6.3.3).

The performance of SVM with the eight different sets of keywords was compared. Training set containing equal number of positive and negative abstracts was used (Figure 6-3). From Figure 6-3, we can see that there was no significant difference among the eight keyword sets, as measured by sensitivity. Therefore, in order to include as much information as possible, we used the TFIDF top 2010 keywords and Z-score top 784 were selected for the remainder of our experiments.

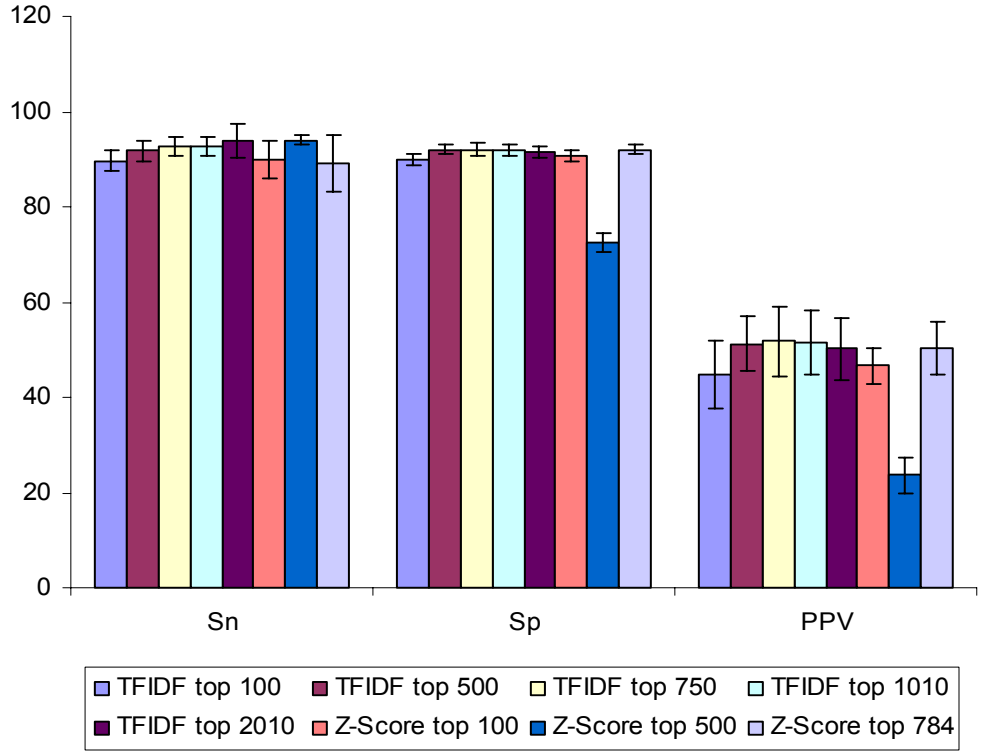


Figure 6-3. Average performance of SVM with different keyword sets as features
Sn: Sensitivity; Sp: Specificity; PPV: Positive Predictive Value

6.5 Improve the sensitivity by changing training sets

In this study, the users in CDC are more interested in sensitivity as the overriding criterion for classification. They do not want to miss any positive article (as few false negatives as possible), while it is fine with some false positives. Then, it is possible to influence the results by biasing toward the positive examples over negative ones. We approach this problem in two ways: one way is to change the number of positive samples in the training set, while the other way is to change the weight of the positive samples in the training set. Next, we will discuss these two ways.

First, we can control the relative sizes of the training set. We compared the performance of SVM with two training sets using TFIDF top 2010 keywords as features. With twice the number of positives than the negatives in the training set the sensitivity of the SVM increased consistently for each of the four test sets, while reducing the specificity and PPV (Table 6-1). We can see that, by increasing the positive samples and negative samples ratio, the sensitivity values were improved.

Table 6-1. SVM classifications with different training sets

	Feb12			Apr1			Apr8			Jun3		
Training Set	Sn	Sp	PPV	Sn	Sp	PPV	Sn	Sp	PPV	Sn	Sp	PPV
Training set 11400 positive 11300 negative	97	92	44	92	91	52	90	93	59	96	90	47
Training set 11400 positive 5300 negative	**98**	87	33	**96**	85	39	**94**	88	46	**97**	87	39

Second, we can weigh the positive samples heavily over negative samples in training the SVM. We tried weighing the positives over negatives by a factor of two, four and eight on a training set of equal positives and negatives and found that the sensitivity results consistently improved at the cost of Specificity and PPV(For Apr1 test set, the Sensitivity values are : 89.08, 97.13, 98.85 and 99.43).

These results indicate that the outcome of the classification can be changed in response to the user's need by tweaking the training set or by assigning different weights to the training sets.

6.6 Improve the sensitivity by combining results using keywords based on TFIDF and Z-Score methods

To improve the sensitivity, we control the number and/or the weight of the positive samples in the training set (section 6.5). Another way to improve sensitivity is that we can combine the classification results derived from different keyword sets. Then we can get the "union" result. Briefly, the union of results is done as follows. If the SVM identified an article as false positive with both the keywords sets then it was considered as false positive. On the other hand if the SVM disagreed with the keyword sets i.e. found as true positive with one keyword set and as false positive with the other set, then the article was still considered as the true positive. The same rule applies to true negatives and false negatives. The performance of SVM was estimated by taking the union of the results obtained from using TFIDF top 2010 and Z-Score all 784 keyword sets. This is done to minimize the false positive rate thereby increasing the sensitivity of the SVM (Table 6-2, and Figure 6-4).

Table 6-2. Union of results using keywords based on TFIDF and Z-Score methods

	Feb12			Apr1			Apr8			Jun3		
	Sn	Sp	PPV	Sn	Sp	PPV	Sn	Sp	PPV	Sn	Sp	PPV
TFIDF top 2010	97	91.8	43.8	92	91	50.3	89.5	93	59.3	96	90	47
Z-Score 784	95	92.6	45	82.5	91	50	85.5	93	58.2	93.3	91.2	48.6
Union of results	**99.3**	95.5	58.6	**95.4**	94.7	65	**92.5**	96.3	73.7	**97.3**	95.6	66.3

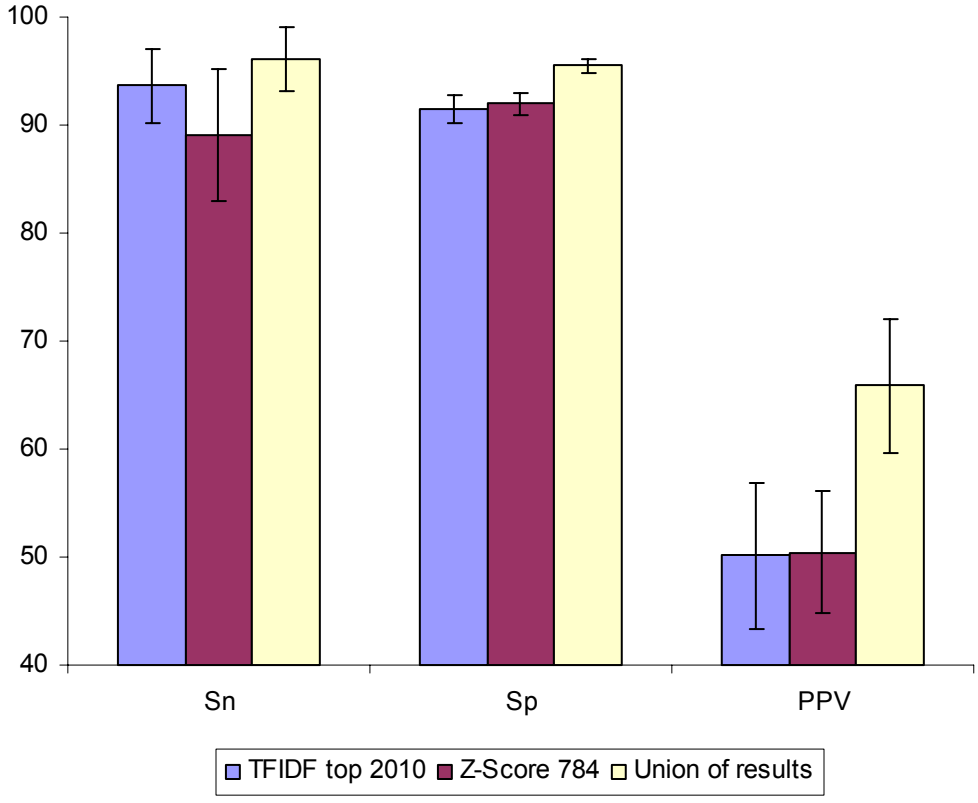


Figure 6-4. Average performance of SVM from the union of results
Sn: Sensitivity; Sp: Specificity; PPV: Positive Predictive Value

6.7 SVM classification outperforms Human expert classification

Before we did this study, a human expert submits the complex query (Figure 6-1) to PubMed. Then this human expert manually categorizes the abstracts as HuGE and non-HuGE. The HuGE abstracts categorized by this human expert are populated into the CDC's HuGENet™ database. Since the whole process is dependent on this human expert, it is possible that the expert could miss some abstracts, e.g. some HuGE abstracts are mis-categorized as non-HuGE abstracts.

In order to test if there are mis-categorized abstracts and if SVM can classify these abstracts as HuGE abstracts, we did the "false positive" analysis. False positives are the abstracts that were categorized as non-HuGE by human experts, but are classified as HuGE by SVM. The goal is to see if any abstracts in the "false positives" are real HuGE abstracts.

The "false positives" from the above result (section 6.6) were given to the CDC appointed expert, in charge of reviewing the literature for the HuGENet^{TM} database, for her scrutiny. In her inspection, she found that on average 50% of the "false positives" produced by the SVM were in fact true positives that were missed by her in her initial review process (Table 6-3). Thus, our automated classification using SVM not only reduced the burden of manual processing, but also increased the sensitivity of the search.

Table 6-3. “False Positive” analysis

	Feb 12	Apr 1	Apr 8	Jun 3
# of “false positives” predicted by SVM	100	89	57	74
# of HuGE abstracts in the “false positives”	59	47	28	32
Percentage of false positives which are true	59%	52.8%	49%	43.2%

6.8 Summary

Automated and standardized categorization and classification of the bio-medical literature is an important challenge facing the scientific community. Due to the vast amount of data produced by emerging biomedical research, manual classification is not feasible. Support vector machines have been widely used in text classification. In this chapter, we tested the application of SVM to HuGE articles classification. The results showed that SVM performed well in terms of sensitivity, which is an important performance evaluation metric for this specific application. Furthermore, SVM can identify some HuGE articles which were missed by human expert. In our investigation into the use of SVM for efficiently classifying HuGE medical abstracts, a high degree of

sensitivity (96.3%) was achieved. In future we wish to develop a tool useful for the average biomedical researcher. Moreover, we intend to develop good benchmarks (e.g. different parameters such as kernel functions) and incorporate them into this personalized tool for the scientific community.

Chapter 7

Conclusion and Future Work

This thesis has focused on the discovery of genomics knowledge by mining biomedical literature. In the last few years, there has been a lot of interest within the scientific community in literature-mining tools to help sort through this ever-growing huge volume of literature and find the nuggets of information most relevant and useful for specific analysis tasks. We extend, expand and compare the available keyword extraction methods and present a new keyword extraction strategy. The keywords are used for gene clustering, gene function classification, and biomedical literature categorization.

7.1 Original contributions to knowledge

This thesis makes the following original contributions to knowledge:

Computer Science:

1. An optimum keyword extraction strategy is presented. The optimum strategy includes the background set, stemming algorithm, stop list and weighting scheme. This strategy can also be applied to other information retrieval problems for the artificial intelligence community;
2. A new clustering algorithm (BEA-PARTITION) is introduced to the bioinformatics community.
3. A comparative study of different clustering algorithms is performed. BEA-PARTITION outperforms k-means and other popular clustering algorithms. We

believe that this important clustering technique has promise for application to other data clustering problems for machine learning community where starting matrices are available;

4. A system based on support vector machine is designed to categorize biomedical literature automatically based on the functional information described in the literature. This system can also be applied to other problems in machine learning and information retrieval, such as spam e-mail detection.

Biology:

1. The keyword extraction strategy that we proposed discovers new biological information that biologists cannot find from the publicly available databases;
2. Genes are clustered based on the shared functional information extracted from biomedical literature and the functional links among genes within each cluster is discovered. This information is very important to the biologists and medical researchers to uncover the functional relationships among genes;
3. Gene functions (in this thesis, yeast genes are used as an example) can be classified by using the functional information derived from literature. Data derived from literature mining outperforms microarray data and phylogenetic data;
4. The biomedical literature by itself can be categorized automatically by the functional information described in the text (in this thesis, text categorization is applied to the related articles).

7.2 Areas for future work

7.2.1 Algorithmic work

There are several aspects of BEA that we are currently exploring with more detailed studies. For example, although the BEA described here performs relatively well on small sets of genes, the larger gene lists expected from microarray experiments need to be tested. In addition, in this report, the magnitude of keyword-gene associations was determined by their z-scores, and term frequency – inverse document frequency (TFIDF). Various other weighting schemes could be compared with the z-score, and TFIDF by precision-recall to determine the conditions under which each performs best (Dumais, 1991). In addition to the weighting schemes, the quality of the keyword lists are also affected by the noisy words (i.e. non-informative words, mis-spelled words), and common properties of English language: (1) Synonymy, different words can be used to describe the same underlying concept; (2) Polysemy, the same word may have more than one meaning. Latent Semantic Indexing (LSI) is a technique developed in information retrieval to address the problems deriving from the use of synonymous, near-synonymous, and polysemous words (Deerwester et al., 1990). Therefore, LSI could be applied to the keyword lists to improve the keyword quality, and as a result, improve the cluster quality. Several other approaches may also improve the performance of the algorithm. First, attention could be focused only on sentences directly referring to the gene name, since if both keyword and gene name occur within the same sentence of the abstracts it is more likely that this keyword relates to the function of the gene. Second, natural language processing could be used to exploit the added information in compound phrases, syntax, and grammatical structures such as negative sentences.

Furthermore, we derived a heuristic to partition the BEA result matrix into clusters. We anticipate that this heuristic will generally work regardless of the type of items being clustered. In Navathe et al. (1984), the heuristic was governed by the nature of transactions against in the database and the goal was to minimize an overall cost function. Generally, optimizing the heuristic to partition a sorted matrix after BEA will be valuable. Finally, we are developing a web-based tool that will include a text mining section to identify functional keywords, and a gene clustering section to cluster the genes based on the shared functional keywords. A preliminary prototype of the tool is shown in Figure 7-1. We believe that this tool should be useful for discovering novel relationships among sets of genes because it links genes by shared functional keywords rather than just reporting known interactions based on published reports. Thus, genes that never co-occur in the same publication could still be linked by their shared keywords.

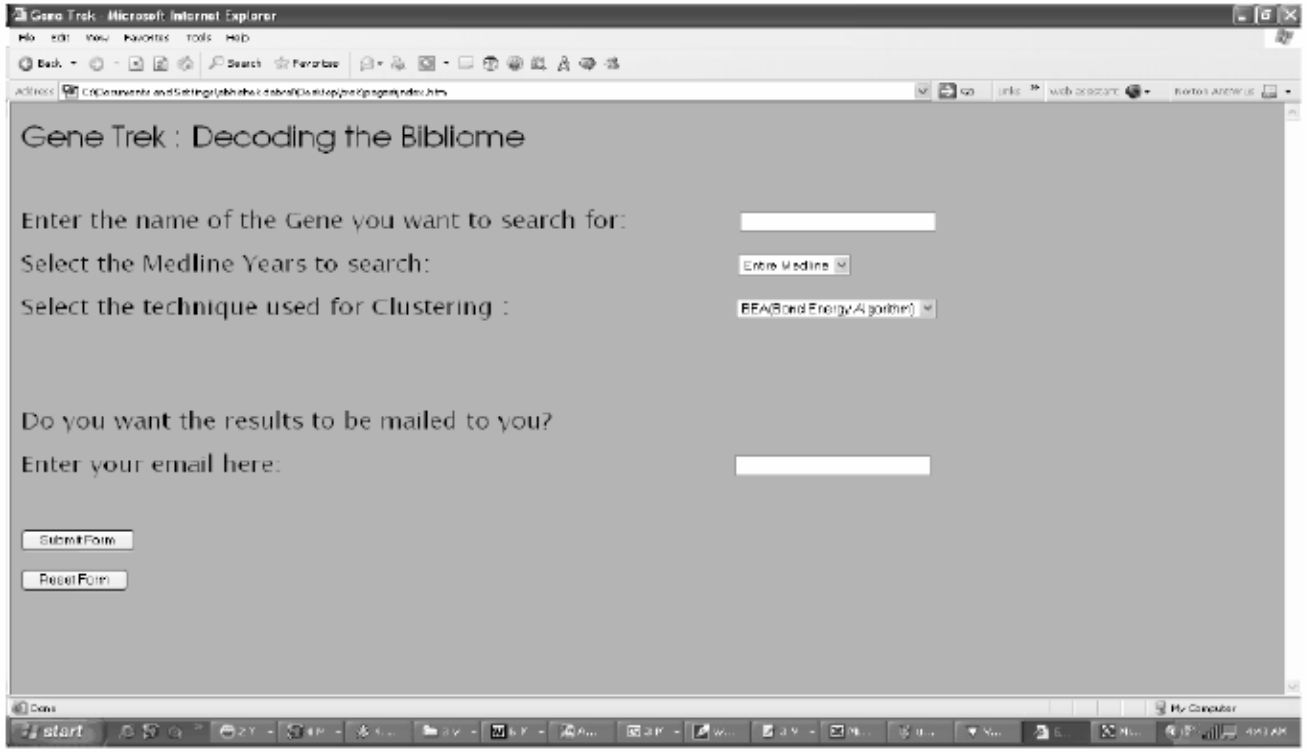


Figure 7-1 User interface of the first prototype system

7.2.2 New system needs to be built

Modern experimental techniques provide the ability to gather vast amounts of biological data in a single experiment. Finding the contextually-sensitive functional relationships between clusters of genes, or *functional clustering* in short, is very important. Functional clustering is well known to us in everyday life, and has been investigated in the psychological literature (Barsalou, 1983). Likewise, in biology, gene function is also contextually-sensitive, and the functions, relationships, and categories of a gene, say, osteopontin depend on the context. For example, we were able to discover functional roles of osteopontin not typically thought of in the neurological context (Liu et al. 2004b).

Considering the challenges we need to deal with, complex system needs to be built. The overall architecture of the system is shown in Figure 7-2.

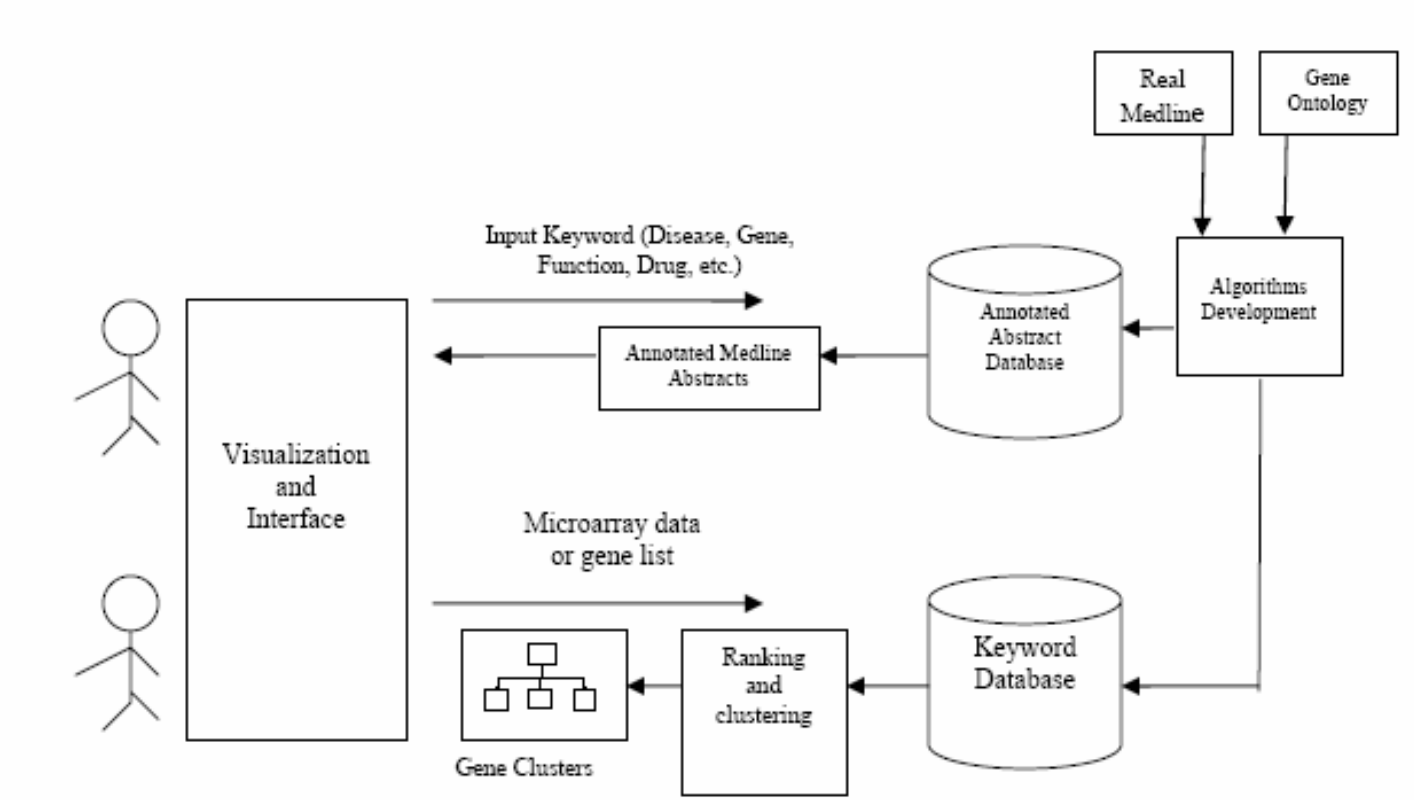


Figure 7-2. Overview of our System

First, we address the users who need to perform advanced search on Medline by providing the disease, gene, drug or other concept names. For such users, we will provide an advanced search and query engine that can retrieve a set of abstracts related to these along with gene ontology and other gene, disease, and drug relation information available from public databases. We create an enriched biomedical literature database for them. Second, we address the scientists who can provide the system a list of gene names derived from microarray or other experiments. Our system will cluster the genes based on the functional information discovered from our keyword-gene cross reference database. Then the clustering results will presented to the users by a visualization tool.

From Figure 7-2, we can see that there are five main tasks:

1. **To develop text analysis algorithms for functional keyword extraction** from the title and abstract fields of MEDLINE searches. Both statistical and Natural Language Processing techniques will be further enhanced and optimized to achieve the best retrieval accuracy.
2. **Improvement and further development of keyword ranking clustering algorithms.** We will investigate alternative techniques of clustering and compare to the BEA (bond energy algorithm) that we have developed and analyzed extensively. Alternate approaches for automatic ranking of keywords will be investigated.
3. **To create a database of functional keywords with efficient indexing** for every known gene in Genbank suitable for querying. A general architecture is shown in Figure 7-3.

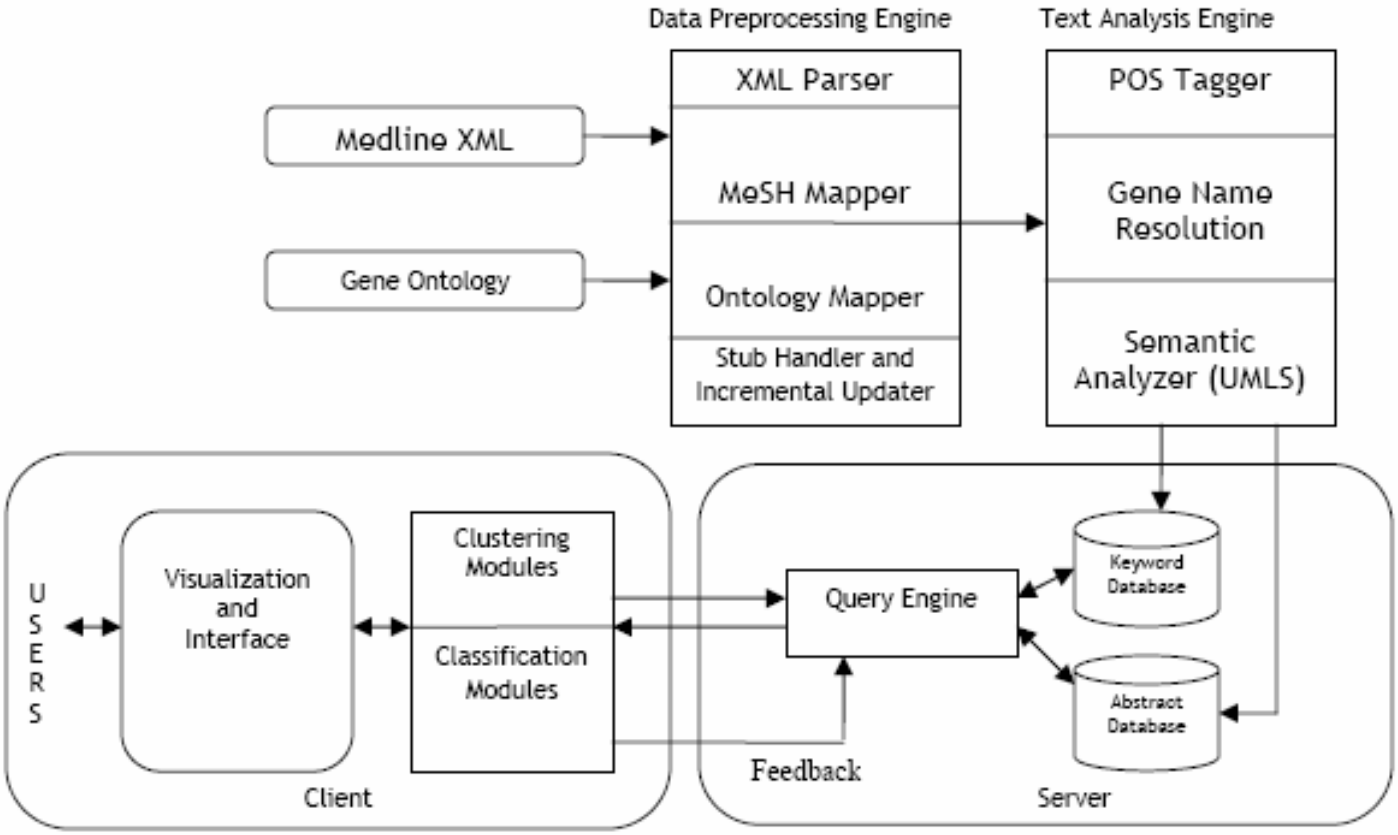


Figure 7-3 General Architecture of the proposed system

4. **To create an enhanced biomedical literature database with new annotations** by combining our keywords with other available sources of knowledge like the gene ontology and UMLS and by continuously learning from new queries answered by the system.

5 **To disseminate our databases and develop tools with easy-to-use interfaces for querying the data.** Typical input will be in the form of lists of genes derived from microarray experiments. A web-enabled test environment and visualization tools will be created and evaluated. Field testers will evaluate the user-friendliness, speed and effectiveness of our tools. Online forms and automated e-mails will be used to collect user feedback; the responses will be archived in logs, analyzed and appropriate changes made.

7.3 Summary

The abundance of biomedical literature motivates an intensive pursuit for effective text-mining tools. Such tools are expected to help uncover the information present in the large and unstructured body of text.

One of the most pressing higher-level needs is the construction of benchmarks and procedures for evaluating the utility of biomedical literature mining tools. Our 26-gene set and 44-gene set can be used as the benchmarks for the gene clustering tool evaluation.

As literature mining challenges in the context of bioinformatics vary widely in aspects such as scope, data sources, and ultimate goals, no single tool can currently perform all the required tasks. However, a combination of methods is likely to address many of the problems. To successfully mine the biomedical literature, it is important to

realize the merits and the limitations of the different literature-mining methods. Moreover, it is essential to coherently state the actual biomedical problems we expect to address by using such methods.

APPENDIX A

PUBLICATIONS FROM THE WORK IN THIS THESIS

The following publications have resulted from the work reported in this thesis and some related work. They may be consulted for further details.

Ying Liu, Shamkant B. Navathe, Alex Pivoshenko, Jorge Civera, Venu Dasigi, Ashwin Ram, Brian J. Ciliax, and Ray Dingledine. (2005) "Text Mining Biomedical Literature for Discovering Gene-to-Gene Relationships. A Comparative Study of Algorithms" IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2(1): 62-76.

Ying Liu, Brian J. Ciliax, Karin Borges, Venu Dasigi, Ashwin Ram, Shamkant B. Navathe, and Ray Dingledine. "Comparison of Two Schemes for Automatic Keyword Extraction from MEDLINE for Functional Gene Clustering." Proceedings of 2004 IEEE Computational Systems Bioinformatics Conference (CSB2004), Stanford University, August 16-19, 2004, pp394-404

Ying Liu, Martin Brandon, Shamkant Navathe, Ray Dingledine, and Brian J. Ciliax. "Automatic Keyword Extraction from MEDLINE for Functional Gene Clustering". 11th MEDInfo 2004 (American Medical Informatics Association Official Annual Conference), San Francisco, September 7-11, 2004, pp292-296.

Ying Liu et al. (2004) Evaluation of a New Algorithm for Keyword-Based Functional Clustering of Genes. RECOMB March 26-31, 2004 San Diego, CA.

Nalini Polavarapu, Shamkant B.Navathe, Ramprasad Ramnarayanan, Abrar ul Haque, Saurav Sahay,**Ying Liu**. (2005) Investigation into Biomedical Literature Classification using Support Vector Machines. Accepted by 2004 IEEE Computational Systems Bioinformatics Conference (CSB2005).

R.J. Dingledine, **Ying Liu**, B.J. Ciliax, J. Civera, A. Ram, S.B. Navathe. Evaluating MEDLINE Text-Mining Strategies for Interpreting DNA Microarray Expression Profiles. Poster presented at the annual conference of the Society of Neuroscience, 2002, Program No. 250.1. *2002 Abstract Viewer/Itinerary Planner.* Washington, DC: Society for Neuroscience, 2002

B.J. Ciliax, M. Brandon, **Ying Liu**, S.B. Navathe, R. Dingledine. Data Mining Keywords Associated with Genes Identified by Expression Profiling with DNA Microarrays. Poster presented at the annual conference of the Society of Neuroscience, 2001,

Program No. 249. *2001 Abstract Viewer/Itinerary Planner*. Washington, DC: Society for Neuroscience, 2001

REFERENCES

Alizadeh, A.A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yand, L., Marti, G. E., Moore, T., Hudson, J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O., and Staudt, L. M. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature, 403:503-511, 2000.

Allen, J. *Natural Language Understanding*, Benjamin Cummings, 1995.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array. Proc. Natl. Acad. Sci. USA, 96:6745–6750, 1999.

Alschul, S.F. et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl. Acids Res., 25: 3389-3402, 1997.

Alter, O., Brown, P. O., and Botstein, D. Singular value decomposition for genome-wide expression data processing and modeling, Proceedings of the National Academy of Sciences, 97:: 10101-10106, 2000.

Andrade, M. A., and Bork, P. Automated extraction of information in molecular biology. FEBS Letters 2000:476:12-17.

Andrade MA. Brown NP. Leroy C. Hoersch S. de Daruvar A. Reich C. Franchini A. Tamames J. Valencia A. Ouzounis C. Sander C. (1999) Automated genome sequence analysis and annotation. Bioinformatics. 15: 391-412

Andrade, M. A., and Valencia, A. Automatic annotation for biological sequences by extraction of keywords from MEDLINE abstracts. Development of a prototype system. Proc. AAAI Conf. on Intelligent Systems for Molecular Biology (ISMB), 1997.

Andrade, M. A., and Valencia, A. Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families, Bioinformatics, 14: 600-607, 1998.

Arabie, P., and Hubert, L. J. The bond energy algorithm revisited, IEEE Transactions on Systems, Man, and Cybernetics, 20: 268-274, 1990.

Aslam, J., Leblanc, A., and Stein, C. Clustering Data without Prior Knowledge, Algorithm Engineering: 4th International Workshop, Springer LNCS 1982

Baeza-Yates, R., and Ribeiro-Neto, B. Modern Information Retrieval, Addison Wesley Longman, New York, NY, 1999.

Bafna, V., and Huson, D.H. 2000. The conserved exon method for gene . nding. *Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, 3–12.

Bahler, D. et al. Symbolic, neural, and Bayesian machine learning models for predicting carcinogenicity of chemical compounds. Journal Chem. Info. Comp. Sci., 40, 906-914, 2000.

Balakrishnan, P. V., Cooper, M. C., Jacob, V. S., and Lewis, P. A. A study of the classification capabilities of neural networks using unsupervised learning: A comparison with k-means clustering, Psychometrika, 59: 509-525, 1994

Bassett, D.E., *et al.* 1999. Gene expression informatics—it's all in your mine. *Nature Genet.* 21, 51–55.

Blair, D. C., and Maron, M. E. An evaluation of retrieval effectiveness for a full-text document-retrieval system. Communic for the Soc. for Computing Machinery, 28: 289-299, 1985.

Blaschke, C., Oliveros, J. C., andValencia, A. Mining functional information associated with expression arrays, Funct. Integr. Genomics, 1:256-268, 2001.

Brank, J. et al. Interaction of feature selection methods and linear classification models. Workshop on Text Learning (TextML-2002), 2002.

Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C., Furey, T. S., Ares, M., and Haussler, D. Knowledge-based analysis of microarray gene expression data using support vector machines, Proceedings of the National Academy of Sciences, 97: 262-267, 2000.

Burge, C.B., and Karlin, S. 1998. Finding the genes in a genomic DNA. *Current Opin. Struct. Biol.* 8, 346–354.

Burges,C. J. C. *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery, Vol. 2, Number 2, p. 121-167, 1998.

Cardie, C. 1997. Empirical methods in information extraction. *AI Magazine* 18(4), 65–80.

Cedeno, W. and V. Vemuri. 1993. An investigation of DNA mapping with genetic algorithms: preliminary results. In: Proc. Of the *Fifth Workshop on Neural Networks*, Vol. 2204 of SPIE.

Chabas, D., Baranzini, S. E., Mitchell, D., Bernard, C. C. A., Rittling, S. R., Denhardt, D.

T., Sobel, R. A., Lock, C., Karpuj, M., Pedotti, R., Heller, R., Oksenberg, J. R., and Steinman, L. The influence of the proinflammatory cytokine, osteopontin, on autoimmune demyelinating disease, Science, 294: 1731-1735, 2001.

Chang, Chih-Chung, and Lin, Chih-Jen LIBSVM : a library for support vector machines, 2001.

Charniak, E. *Statistical Language Learning*, MIT Press, New Haven, CT, 1993.

Chaussabel, D., and Sher, A. Mining microarray expression data by literature profiling, Genome Biology, 3: 1-16, 2002.

Cheeseman, P., and Stutz, J. Bayesian classification (autoclass): Theory and results, in Advances in Knowledge Discovery and Data Mining, 153–180, 1996.

Cherepinsky, V., Feng, J., Rejali, M., and Mishra, B. Shrinkage-based similarity metric for cluster analysis of microarray data, Proc. Natl. Acad. Sci. USA, 100: 9668-9673, 2003.

Chen, Y., Dougherty, E., and Bittner, M. L. Ratio-based decisions and the quantitative analysis of cDNA micro-array images, journal of Biomedical Optics, 2: 364-374, 1997.

Chu, S. et al. The transcriptional program of sporulation in budding yeast. Science, 282: 699-705, 1998.

Clare, A. Machine learning and data mining for yeast functional genomics. Ph.D thesis, Department of Computer Science, University of Wales, 2003

Claverie, J. M. Computational methods for the identification of differential and coordinated gene expression, Human Molecular Genetics, 8: 1821-1832, 1999.

Cohen, W.W., and Singer, Y. 1999. Context-sensitive learning methods for text categorization. *ACM Transaction on Information Systems*, 17(2), 141–173.

Cole, S.T., Brosch, R., Parkhill, J., Garnier, T., Churcher, C., Harris, D., Gordon, S.V., Eiglmeier, K., Gas, S., Barry, C.E.III., Tekaia, F., Badcock, K., Basham, D., Brown, D., Chillingworth, T., Connor, R., Davies, R., Devlin, K., Feltwell, T., Gentles, S., Hamlin, N., Holroyd, S., Hornsby, T., Jagels, K., Krogh, A., McLean, J., Moule, S., Murphy, L., Oliver, K., Osborne, J., Quail, M.A., Rajandream, M-A., Rogers, J., Rutter, S., Seeger, K., Skelton, J., Squares, S., Squares, R., Sulston, J.E., Taylor, K., Whitehead, S., & Barrell, B.G. 1998. Deciphering the biology of Mycobacterium tuberculosis from the complete genome sequence. Nature, 393(June), 537–544.

Cowie, J., and Lehnert, W. Information extraction. *Communications of the ACM*, 39: 80–91, 1996.

Craven, M., and Kumlien, J. Constructing biological knowledge bases by extracting information from text sources. Proc. AAAI Conf. on Intelligent Systems for Molecular Biology (ISMB), 77–86, 1999.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41:391-407, 1990.

DeRisi, J., Iyer, V., & Brown, P. 1997. Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale. Science, 278(October), 680–686.

DeRisi, J., Penland, L., Brown, P. O., Bittner, M. L., Meltzer, P. S., Ray, M., Chen, Y., Su, Y., and Trent, J. M. Use of a cDNA microarray to analyze gene expression patterns in human cancer, Nature Genetics, 14: 457-460, 1996.

Drucker, H., Vapnik, V., and Wu, D. 1999. Automatic text categorization and its applications to text retrieval. IEEE Transactions on Neural Networks 10, 5, 1048–1054

Dubes, R., and Jain, A. Algorithms for Clustering Data. Prentice Hall, 1988.

Dumais, S.T. Improving the retrieval of information from external sources. Behavior Research Methods, Instruments, & Computers 23: 229-236, 1990.

Dumais, S. T. and Chen, H. Hierarchical classification of Web content. In Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval (Athens, GR, 2000), pp. 256–263, 2000.

Dumais, S. T., Platt, J., Heckerman, D., and Sahami, M. 1998. Inductive learning algorithms and representations for text categorization. In Proceedings of CIKM 98, 7th ACM International Conference on Information and Knowledge Management (Bethesda, US, 1998), pp. 148–155.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acsd. Sci USA, 95: 14863-14868, 1998.

Emanuelsson, O., *et al.* 2000. Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J. Mol. Biol.* 300, 1005–1016.

Feldman, R., Dagan, I., and Kloegsen, W. 1996. Efficient algorithm for mining and manipulating associations in texts. 13th European meeting on cybernetics and research.

Fickett, J. & M. Cinkosky. 1993. A genetic algorithm for assembling chromosome physical maps. In: Proc. Of the Second International Conference on

Bioinformatics, Supercomputing, and Complex Genome Analysis. St. Petersburg, FL: *World Scientific*. p. 272-285.

Friedman, C., et al. Genies: A natural-language processing system for the extraction of molecular pathways from journal articles. Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB), S74–S82, 2001.

Fukuda, K., et al. 1998. Toward information extraction: Identifying protein names from biological papers. Proc. Pacific Symposium on Biocomputing (PSB), 705–716.

Funk, M. E., and Reid, C. R. Indexing consistency in MEDLINE. Bull. Med. Libr. Assoc. 71: 176-183, 1983.

Furger, K. A., Menon, R. K., Tuck, A. B., Bramwell, V. H., and Chambers, A. F. The functional and clinical roles of osteopontin in cancer and metastasis, Curr Mol Med, 1: 621-632, 2001.

Goffeau, A., Barrell., B., Bussey, H., Davis, R., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J., Jacq, C., Johnston, M., Louis, E., Mewes, H., Murakami, Y., Philippsen, P., Tettelin, H., & Oliver, S. 1996. Life with 6000 genes. Science, 274, 563–7.

Golub, T.R. et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science, 286: 531-537, 1999.

Han, B., Vucetic, S., and Obradovic, Z., Reranking Medline Citations By Relevance to a Difficult Biological Query, IASTED Int'l Conf. Neural Networks and Computational Intelligence, Cancun, Mexico, 2003.

Hanisch, D., et al. Playing biology's name game: Identifying protein names in scientific text. Proc. Pacific Symposium on Biocomputing (PSB), 403–411, 2003.

Harrington, C.A., Rosenow, C., and Retief, J. Monitoring gene expression using DNA microarrays. Curr Opin Microbiol, 3(3):285–91, 2000.

Hartner, A., Porst, M., Gauer, S., Prols, F., Veelken, R., and Hilgers, K. F. Glomerular osteopontin expression and macrophage infiltration in glomerulosclerosis of DOCA-salt rats, Am J Kidney Dis. 38: 153-164, 2001.

Hassoun, M.H. March 1995. *Fundamentals of Artificial Neural Networks*. MIT Press.

Hayes, P. 1992. Intelligent high-volume processing using shallow, domain-speci.c techniques, in *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*, 227–242. Lawrence Erlbaum Assoc., Hillsdale, NJ.

Hayes, P., and Weinstein, S. 1990. CONSTRUE: A system for content-based indexing of a database of news stories. *Proc. 2nd Annual Conf. on Innovative Applications of Arti. cial Intelligence*.

Hearst, M.A. Untangling text data mining. Proc. 37th Annual Meeting of the Association for Computational Linguistics, 3–10, 1999.

Herrero, J., Valencia, A., and Dopazo, J. A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics, 17:126–136, 2001.

Holland, J.H. 1975. *Adaptation in natural and artificial systems*. Cambridge, MA: MIT Press.

Horton, P., and Nakai, K. 1997. Better prediction of protein cellular localization sites with the K nearest neighbors classi. er. *Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*.

Hvidsten, T. R., and Komorowski, J. Predicting gene function from gene expression and ontologies. Pacific Symposium on Biocomputing 6:299-310, 2001.

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

Jaakkola, T., *et al.* 2000. A discriminative framework for detecting remote protein homologies. *J. Comp. Biol.* 7(1/2), 95–114.

Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: a review. ACM Computing Surveys, 31(3):254–323, September 1999.

Jenssen, T. K., Laegreid, A., Komorowski, J., and Hovig, E. A literature network of human genes for high-throughtput analysis of gene expression, Nat. Genet., 178: 139-143, 2001.

Jenssen, T. K., and Vinterbo, S. A Relational Approach to Defining Document Set Relevance: An Application in Human Genetics., IDI-rapport 7/00, Dept. of Computer and Information Science, Norwegian University of Science and Technology, 2000.

Joachims, T. Text categorization with support vector machines: learning with many relevant features. Proceedings of ECML-98, 137-142, 1998.

Joachims, T. Estimating the Generalization performance of an SVM Efficiently. International Conference on Machine Learning (ICML), 2000.

Joachims T. Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms. Kluwer Academic Publishers, 2002.

Kageura, K., and Umino, B. Methods of automatic term recognition–a review. Terminology, 3: 259-289, 1996.

Kaufman, L., and Rousseeuw, P. J. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley and Sons, 1990.

Klinkenberg, R. and Joachims, T. 2000. Detecting concept drift with support vector machines. In Proceedings of ICML-00, 17th International Conference on Machine Learning (Stanford, US, 2000).

Kohonen, T. Self-Organization and Associative Memory. Spring-Verlag, Berlin, 1984.

Komorowski, J., & Øhrn, A. 1999. Diagnosing Acute Appendicitis with Very Simple Classification Rules. Page 462467 of: Proc. Third European Symposium on Principles and Practice of Knowledge Discovery in Databases.

Korf, I., *et al.* 2001. Integrating genomic homology into gene structure prediction. *Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, S140–S148.

Larkey, L.S., and Croft,W.B. 1996. Combining classi. ers in text categorization.*Proc. 19th ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR-96)*, 289–297.

Leek, T. R. Information extraction using hidden Markov models. Master's thesis, Department of Computer Science, University of California, San Diego, 1997.

Lehnert, W., *et al.* 1991. Description of the CIRCUS System as used for MUC-3. *Proc. 3rd Message Understanding Conference (MUC-3)*, 223–233.

Lewis, D.D. 1995. Evaluating and optimizing autonomous text classi. cation systems. *Proc. 18th ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR-95)*, 246–254.

Lewis, D.D., and Hayes, P.J. 1994. Guest editorial for the special issue on text categorization. *ACM Transactions on Information Systems*, 12(3).

Lewis, D.D., and Ringuette, M. 1994. A comparison of two learning algorithms for text categorization. *Proc. 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94)*, 81–93.

Lewis, D.D., *et al.* 1996. Training algorithms for linear text classi. ers. *Proc. 19th ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR-96)*, 298–306.

Li L.P., *et al.* 2001. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics* 17(12), 1131-1142.

Liu, H. et al. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns, Genomic Informatics, 13, 51-60, 2002.

Lockhart, D.J., *et al.* 1996. Expression monitoring by hybridization to high-density oligonucleotide array. *Nature Biotechnol.* 14, 1675–1680.

Macgregor, P.F., and Squire, A.. Application of microarrays to the analysis of gene expression in cancer. Clin Chem, 48(8):1170–7, 2002.

McCormick, W. T., Schweitzer, P. J., and White, T. W. Problem decomposition and data reorganization by a clustering technique, Oper. Res., 20: 993-1009, 1972.

McCulloch, W.S. and W. Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*. 5: 115-133.

McQueen J.B. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Univ.of California, Berkeley, 1967. Univ.of California Press, Berkeley.

Navathe, S. Ceri, S., Wiederhold, G., and Dou, J. Vertical partitioning algorithms for database design, ACM Trans. On Database Systems, 9: 680-710, 1984.

Masys, D. R., Welsh, J. B., Lynn, F. J., Gribskov, M., Klacansky, I., and Corbeil, J. Use of keyword hierarchies to interpret gene expression patterns. Bioinformatics 2001:17: 319-26.McQueen, J. B. Some methods for classification and analysis of multivariate observations, In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1: 281–297, Univ.of California, Berkeley, 1967.

Myers, E. 1999. Whole-genome DNA sequencing. *IEEE Computational Engineering and Science* 3(1), 33–43.

Oliver, S., Winson, M., Kell, D., & Baganz, F. 1998. Systematic functional analysis of the yeast genome. Trends Biotechnol., 16(September), 373–378.

Ozsu, A. T., and Valduriez, P. Principles of Distributed Database Systems, (2nd edition). Prenticel Hall Inc, 1999.

Parsons R.J., S. Forrest, and C. Burks. 1995. Genetic algorithms, operators, and DNA fragment assembly. *Machine Learning*. 21: 11-33.

Pavlidis, P. et al. (2002) Learning gene functional classifications from multiple data types. J. Computational Biology, 9: 401-411

Pellegrini, M. et al. (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. Proc. Natl. Acad. Sci. USA, 96: 4285-4288

Perez-Iratxeta, C, P Bork and MA Andrade (2002) Association of genes to genetically inherited diseases using data mining. Nature Genetics 31: 316-319.

Perou, C. M., Jeffrey, S. S., Rijn, M. V. D., Rees, C. A., Eisen, M. B., Ross, D. T., Pergamenschikov, A., Williams, C. F., Zhu, S. X., Lee, J.C.F., Lashkari, D., Shalon, D., Brown, P. O., and Bostein, D. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. Proc. Natl. Acad. Sci. USA, 96:9212–9217, 1999.

Porter, M. An algorithm for suffix stripping, Program, 14:130-137, 1980.

Quackenbush, J. Computational analysis of microarray data, Nature Reviews Genetics, 2: 418-427, 2001.

Rajman, M., and Besancon, R. 1997. Text Mining: Natural Language techniques and Text Mining Proceedings of the seventh IFIP 2.6 Working Conference on Database Semantics (DS-7), Chapam & Hall IFIP Proceedings serie, (1997) Oct 7-10.

Raychaudhuri, S., Chang, J. T., Imam, F., and Altman, R. B. The computational analysis of scientific literature to define and recognize gene expression clusters, Nucleic Acids Research, 15: 4553-4560, 2003.

Raychaudhuri, S., Schutze, H., and Altman, R. B. Using text analysis to identify functionally coherent gene groups. Genome Res. 12: 1582-1590, 2002.

Riloff, E., and Lehnert, W. 1994. Information extraction as a basis for high-precision text classi. cation. *ACM Transactions on Information Systems* 12(3), 296–333.

Rindflesch, T.C., et al. Edgar: Extraction of drugs, genes and relations from the biomedical literature. Proc. Pacific Symposium on Biocomputing (PSB), 514–525, 2000.

Rogati, M. and Yang, Y. High-performing feature selection for text classification. CIKM'02, p659-661, 2002.

Rojas R. 1996. *Neural Networks - A Systematic Introduction*. Springer-Verlag, Berlin, New-York.

Rose, K. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. Proc. IEEE, 96:2210–2239, 1998.

Rose, K., Gurewitz, E., and Fox, G. Phys. Rev. Lett., 65:945–948, 1990.

Ross-Macdonald, P et al. 1999. Large-scale analysis of the yeast genome by transposon tagging and gene disruption. Nature, 402(Nov), 413–418.

Rumelhart, D.E. and J.L. McClelland. 1988. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vols. 1 and 2. MIT Press, Cambridge, MA.

Russell, S.J., and Norvig, P. 1995. *Arti. cial Intelligence, A Modern Approach*, chap. 22–23, Prentice Hall, Englewood Cliffs, NJ.

Sahami, M. Using Machine Learning to Improve Information Access. Ph.D. thesis, Stanford University, Computer Science Department, 1998.

Salton, G. Automatic Text Processing, Addison-Wesley, Reading, MA, 1989.

Salton, G., and Buckley, C. Text-weighting approaches in automatic text retrieval. Information Processing and Management, 24:513-523, 1988.

Saracevic, T. Individual differences in organizing, searching and retrieving information. Proc Amer. Soc. Information Science, 28: 82-86, 1991.

Sasik, R. *et al*. 2001. Percolation Clustering: A Novel Approach to the Clustering of Gene Expression Patterns in Dictyostelium Development. *PSB Proceedings* 6, 335-347.

Schena, M., *et al.* 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270, 467–470.

Schulze, A., and Downward, J. Navigating gene expression using microarrays–a technology review. Nat Cell Biol, 3(8):E190–5, 2001.

Sebastiani, F. Machine Learning in Automated Text Categorization, ACM Computing Surveys, 34: 1-47, 1999.

Selaru F.M., Y. Xu, J. Yin *et al*. 2002 Artificial neural networks distinguish among subtypes of neoplastic colorectal lesions. *Gastroenterology* 122:606-613.

Shatkay, H., *et al.* Genes, themes and microarrays: Using information retrieval for large scale gene analysis. Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB), 317–328, 2000.

Shatkay H., and Feldman R. 2003 Mining the biomedical literature in the genomic era: an overview. Journal of Computational Biology, 10: 821-855.

Sherlock, G. Analysis of large-scale gene expression data, Curr Opin Immunol, 12:201–205, 2000.

Smet, F. D., Mathys, J., Marchal, K., Thijs, G., Moor, B. De, and Moreau, Y. Adaptive quality-based clustering of gene expression profiles. Bioinformatics, 18:735–746, 2002.

Sonnhammer, E.L.L., *et al.* 1998. A hidden Markov model for predicting transmembrane helices in protein sequences. *Proc. Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*.

Spellman, P.T. et al. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. Mol. Biol. Cell, 9: 3273-3297, 1998.

Stormo, G., Schneider, T., Gold, L. & Ehrenfeucht, A. 1982. Use of the perceptron algorithm to distinguish translational initiation in E.coli. Nuclei Acids Research 10: 2997-3011.

Strachan, S., and Read, A. P. Human Molecular Genetics, BIOS Scientific Publishers Ltd, 1999.

Strehl, A. Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining, Ph.D dissertation, Dept. Electric and Computer Eng., The University of Texas at Austin, Austin, Texas, 2002.

Swanson, D. R. Searching natural language text by computer. Science 132: 1099-1104, 1960.

Taira, H., and Haruno, M. Feature selection in SVM text categorization. AAAI-99, 480-486, 1999.

Tamayo, P., Solni, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., and Golub, T. R. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. Proc. Natl. Acad. Sci. USA, 96:2907–2912, 1999.

Tan, A. C., and Gilbert, D. Ensemble machine learning on gene expression data for cancer classification. Applied Bioinformatics, 3, S75-S83, 2003.

Tanabe, L., Scherf, U., Smith, L. H., Lee, J. K., Hunter, L., and Weinstein, J. N. MedMiner: an Internet text-mining tool for biomedical information, with application to gene expression profiling. Biotechniques 1999:27: 1210-1214.

Tong, S., and Koller, D. Support Vector Machine Active Learning with Applications to Text Classification. Proceedings of ICML-00, 17th International Conference on Machine Learning, 2000.

Troyanskaya, O.G. et al., A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *Saccharomyces cerevisiae*). Proc. Natl. Acad. Sci. USA, 100: 8348-8353, 2003.

Valafar, F. Pattern recognition techniques in microarray data analysis: a survey. Special issue of Annals of New York Academy of Sciences, techniques in Bioinformatics and Medical Informatics. 2002, 980: 41-64.

Valafar H., O. Ersoy, F. Valafar. June 1998. Parallel Implementation of Distributed Global Optimization (DGO). Proceedings of the international conference on *Parallel Distributed Processing Techniques and Applications*. Las Vegas, Nevada, 1782-1787

Valafar, H., F. Valafar, and O. Ersoy. 1996. Distributed Global Optimization (DGO). Proceedings of the *International Conference on Neural Networks.* Washington, DC, June 2-6, 530a-536.

van Hal, N.L., Vorst, O., van Houwelingen, A.M., Kok, E. J., Peijnenburg, A., Aharoni, A., van Tunen, A. J., and Keijer, J. The application of dna microarrays in gene expression analysis. J Biotechnol, 78(3):271–80, 2000.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*, Springer-Verlag, NY.

Venter, J.C., *et al.* 2001. The sequence of the human genome. *Science* 291, 1304–1351.

Vladimir, N., and Vapnik, The Nature of Statistical Learning Theory. Springer, 1995.

Willett, P. Recent Trends in Hierarchic Document Clustering: A Critical Review, Information Processing and Management, 24: 577-597, 1988.

Winzeler, E., & Davis, R. 1997. Functional analysis of the yeast genome. Current Opinion in Genetics and Development, 7, 771–776.

Witten, I. H., and Frank, E. Data mining: practical machine learning tools and techniques with java implementations. Morgan Kaufmann, San Francisco, 1999.

Witten, I. H., et al. 1999. Managing Gigabytes, Compressing and Indexing Documents and Images (2nd ed.), Morgan-Kaufmann, San Diego, CA.

Wu, C. H. 1995. Chapter titled "Gene Classification Artificial Neural System" in *Methods In Enzymology: Computer Methods for Macromolecular Sequence Analysis*, Edited by Russell F. Doolittle, Academic Press, New York.

Wu, C., S. Zhao, H. L. Chen, C. J. Lo and J. McLarty. 1996. Motif identification neural design for rapid and sensitive protein family search. *CABIOS,* 12 (2), 109-118.

Xu, Y., Olman, V., and Xu, D. EXCAVATOR: a computer program for efficiently mining gene expression data, Nucleic Acid Res., 31: 5582-5589, 2003.

Yandell, M. D., and Majoros, W. H. Genomics and natural language processing. Nature Reviews 3, 601–610, 2002.

Yang, Y., and Chute, C.G. 1994. An example-based mapping method for text categorization and retrieval. *ACM Trans. Inf. Systems* 12(3), 252–277.

Yang, Y. and Liu X. A re-examination of text categorizationmethods. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99), 1999.

Yang, Y. and Pederson, J.O. A comparative study on feature selection in text categorization. International Conference on machine Leaning (ICML'97), p412-420, 1997.

Zhang, M. 1999. Large Scale Gene Expression Data Analysis: A New Challenge to Computational Biologists. Genome Research, 9, 681–688.

Zhang, C. and A.K. Wong. 1997. A genetic algorithm for multiple molecular sequence alignment. *Comput. Appl. Biosci*. 13: 565-581.

VITA

Ying Liu received his Ph.D. in Computer Science from the College of Computing at Georgia Institute of Technology in 2005. From September 2005, He will be working as a tenure-trek assistant professor in Department of Computer Science, University of Texas at Dallas. In 2001, Ying received the degree of M.S. in Bioinformatics from the School of Biology at Georgia Institute of Technology, and M.S. in Computer Science from the College of Computing at Georgia Institute of Technology. Ying graduated from the Nanjing University, China with a B.S in Biology. His research interests include bioinformatics, medical informatics, computational biology, and data mining. His thesis work focuses on applying machine learning algorithms for DNA microarray data analysis and text mining biomedical literature to discover gene-to-gene relationships. He has been working closely with the biomedical researchers from Emory University School of Medicine and Center of Disease Control and Preventation (CDC). In 2002, he worked as a research intern in General Electric Global Research Center, where he designed a cardiovascular pathway database. He also developed algorithms to mine different biological databases to discover such information as consensus sequences and protein domains.