APPLICATION OF COMMITTEE k-NN CLASSIFIERS FOR GENE EXPRESSION PROFILE CLASSIFICATION

A Thesis

Presented to

The Graduate Faculty of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Manik Dhawan

December, 2008

APPLICATION OF COMMITTEE k-NN CLASSIFIERS FOR GENE EXPRESSION

PROFILE CLASSIFICATION

Manik Dhawan

Thesis

Approved:

____
Advisor
Dr. Zhong-Hui Duan

____
Committee Member
Dr. Kathy J. Liszka

____
Committee Member
Dr. Timothy W. O'Neil

____
Department Chair
Dr. Wolfgang Pelz

Accepted:

____
Dean of the College
Dr. Ronald F. Levant

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

____
Date

ABSTRACT

The study of this thesis was an effort to design a stable classification system to categorize microarray gene expression profiles. Currently, high-throughput microarray technology has been widely used to simultaneously probe the expression values of thousands genes in a biological sample. However, due to the nature of DNA hybridization, the expression profiles are highly noisy and demand specialized data mining methods for analysis. This study focuses on developing an effective and stable sample classification system using gene expression data. The system includes a sequence of data preprocessing steps and a committee of k-nearest neighbor (k-NN) classifiers that are of different architectures and use different sets of features. A case study of the system was performed to illustrate the effectiveness of the committee approach. A real microarray dataset, the MIT leukemia cancer dataset, was used in the study. The expression profiles were first subjected to the sequence of preprocessing steps. About 38% of the genes were removed. The remaining informative genes were then ranked and used for constructing k-NN classifiers. The k-NN classifiers that gave the best results were further recruited to form a decision-making committee. The performance of the committee of k-NN classifiers were later evaluated using a new dataset. The results of the case study indicate that the system developed consistently outperforms individual k-NN classifiers in terms of both accuracy and stability.

ACKNOWLEDGEMENTS

First I would like to thank my advisor, Dr Zhong-Hui Duan for giving me the opportunity to work on this Masters thesis and for her invaluable input in the entire course of the project. The course Introduction to Bioinformatics under Dr. Zhong-Hui Duan was the turning point behind my decision to work in the field of Bioinformatics. This thesis would not have been possible without her guidance and persistent help.

A special thanks to my committee- Dr Kathy J. Liszka and Dr Timothy W. O'Neil for their time and effort and especially for their invaluable suggestions.

I would like to take a chance to thank my friends Sudarshan Selvaraja, Rochak Vig and Satish Reddy Sangem for their valuable suggestions. Special thanks to my seniors Saket Kharsikar and Mihir Sewak who guided me throughout the thesis work.

Lastly, I would like to express my gratitude towards my parents and all my family for their faith and who were always there for me all through the progress of my thesis and eventually my degree.

Working on the thesis was a process which helped me to learn to think out of the box and how we can look at facts from different points of views. This is a trait which for sure will help me achieve my goals in life.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction to Bioinformatics

The field of bioinformatics has come into existence very recently and has gained enormous popularity and attention. This field is all about finding the solution to biological problems with the help of information systems based on computers. Bioinformatics has led to a vast amount of research advances and has proven effective for diagnosing, classifying and discovering many aspects that lead to diseases like cancer [1]. The focus from a macro level to a molecular level has led to a better understanding of the functions of genes.

Various developments in the field of bioinformatics have led to efficient data mining and classification algorithms and techniques. The answers to very basic questions like the origin of life, color of skin and causes of different diseases are known to lie in the genetic codes which are the part of the DNA in all living organisms. Advancements in technology have made it possible to gather all this genetic information into computers and further use it for research purposes.

Since the start of the GenBank genomic sequences have been added to its databases. Hence, the information is growing day by day. New sequences are added to the data bank daily. With that the research in the field has now reached a whole new level. As we come to know more and more about the genetic sequences, we can explore the possibilities. Comparative studies aid a lot in the classification and identification of new gene patterns. The major research areas in the field of bioinformatics are sequence analysis, analyzing gene expressions, protein expression analysis and protein structure prediction [2].

The present study involves the application of machine learning methods for the classification of cancer samples using the gene expression data obtained from the microarray experiment. A brief explanation of gene expression and microarrays will help aid in the proper understanding of the current classification problem.

1.2 Gene Expressions and Microarrays

Before we proceed to the objectives of the current study, we need to know the basics of gene expressions and the microarray technology.

1.2.1 Understanding gene expressions

Genetic material is the same in all cells of the body. The only thing that makes the organs in the body act differently is that some genes are dormant in certain cells. Some genes are expressed in a cell while others are not, creating the whole variation.

These dormant genes in the cell are sometimes triggered in some circumstances which lead to several diseases and disorders like cancer [3]. This leads to malfunctions in the proper working of the cells. Bioinformatics research shows that the expression levels of genes away from normal samples might be a reason for several abnormalities.

1.2.2 Analyzing gene expression levels

With the help of new age technologies, we are now able to study the expression levels of thousands of genes at once. In this way, we can try to compare the expression levels in normal and abnormal cells. The expression values in affected genes can help us compare them with regular expression values and thus tell us the reason for the abnormality. The quantitative information of gene expression profiles can help boost the fields of drug development, diagnosis of diseases and further understanding the functioning of living cells. A gene is considered *informative* when its expression helps to classify samples to a disease condition or not. All of these informative genes help us develop classification systems which can distinguish normal cells from the abnormal ones. The goal of this study is to build a classification model which can efficiently classify the normal and tumor samples using gene expression data obtained from microarray study.

1.2.3 Introduction to microarrays

A *microarray* is a tool used to sift through and analyze the information contained within a genome. A microarray consists of different nucleic acid probes that are chemically attached to a substrate, which can be a microchip, a glass slide or a microsphere-sized bead [4]. The first DNA microarray chip was engineered at Stanford University, whereas Affymetrix Inc. was the first to create the patented DNA microarray wafer chip called the Gene Chip [5]. The microarray data used for the current study was collected using Affymetix Gene Chips also knows as an oligonucleotide microarray. Figure 1.1 shows a typical experiment with an oligonucleotide chip. Messenger RNA is extracted from the cell and converted to cDNA. After the amplification and labeling of the sample it is hybridized on the chip. After the washing of unhybridized material, the chip is scanned with a laser scanner and the image analyzed by computer.

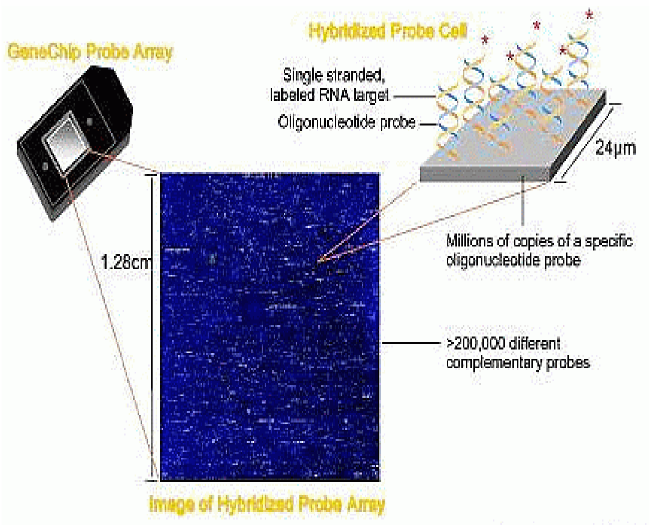

GeneChip Probe Array
Hybridized Probe Cell
Single stranded, labeled RNA target
Oligonucleotide probe
24μm
1.28cm
Millions of copies of a specific oligonucleotide probe
>200,000 different complementary probes
Image of Hybridized Probe Array

Figure 1.1 Microarray Chip. [6]

In a dual channel microarray experiment, the first step is to gather samples from both the control cell and the experiment cell. Both the control sample and the experiment sample are colored using dyes of different color. The labeled product is generated by reverse transcription. Labeled samples are then mixed with hybridization solution. The solution is transferred onto the microarray chip and left for hybridization. Hybridization is the process where the denatured DNA strands associate with their complimentary strands via specific base-pair bonding. Hybridization occurs between labeled denatured DNAs of target samples and the cDNA strands of known sequences on the spots of the array. The chip is kept overnight and all the non specific binding is washed off. The different colored dyes emit varying wavelengths based on a mixture of known and unknown samples.

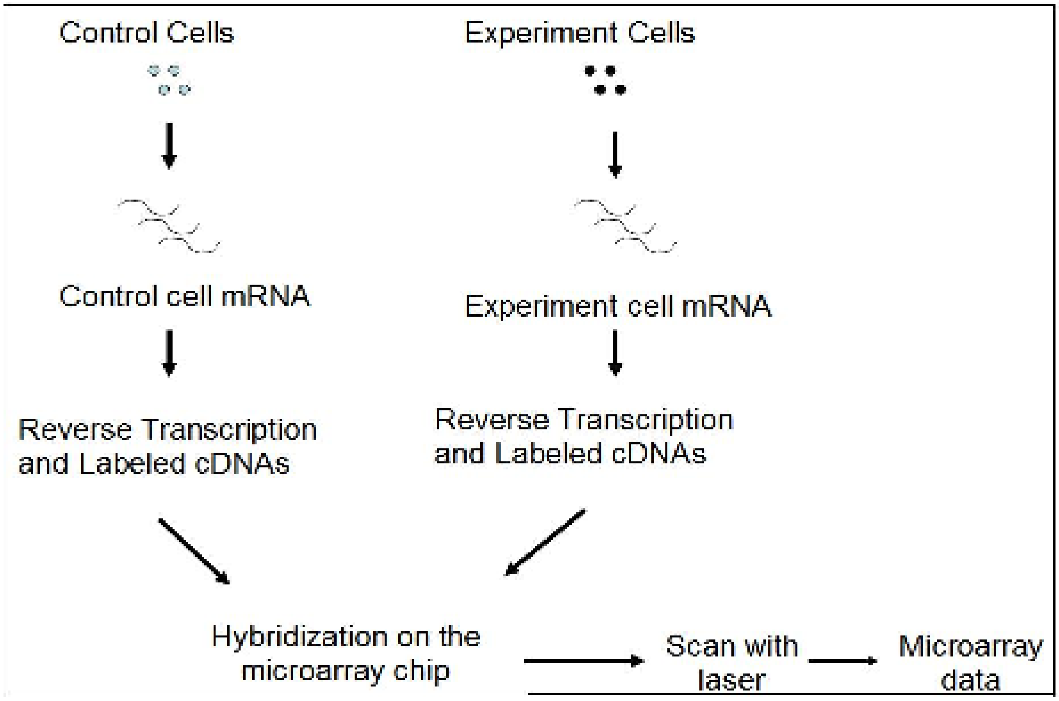

Control Cells
Experiment Cells
Control cell mRNA
Experiment cell mRNA
Reverse Transcription and Labeled cDNAs
Reverse Transcription and Labeled cDNAs
Hybridization on the microarray chip
Scan with laser
Microarray data

Figure 1.2 Hybridization using microarray

The scanning and imaging equipment then detects the varying intensities of fluorescence. This intensity information is further used to detect the variation of hybridization of unknown target samples from control samples [7]. The process can be seen in figure 1.2.

1.3 Need for automated analysis of microarray data

Microarrays have paved the way for researchers to gather a lot of information from thousands of genes at the same time. The main task is the analysis of this information. Looking at the size of the data retrieved from the genetic databases, we can definitely say that there is no way to analyze and classify this information manually. In the current study, an effort has been made to classify gene expression data of leukemia patients into two classes of ALL and AML samples. This study tries to unveil the potential of classification by automatic machine learning methods. In particular, we use the k-NN classifier committee approach.

1.4 Classification techniques

In the current study, we deal with a classification problem which focuses on dividing the samples of patients suffering from Leukemia cancer into two categories. Any classification method uses a set of parameters to characterize each object. These features are relevant to the data being studied. Here we are discussing methods of supervised learning where we know the classes into which the objects are to be classified.

We also have a set of objects with known classes. A training set is used by the classification programs to learn how to classify the objects into desired categories. This training set is used to decide how the parameters should be weighted or combined with each other so that we can separate various classes of objects. In the application phase, the trained classifiers can be used to determine the categories of objects using new patient samples called the testing set. The various well-known classification methods are discussed as follows [8].

1.4.1 Neural networks

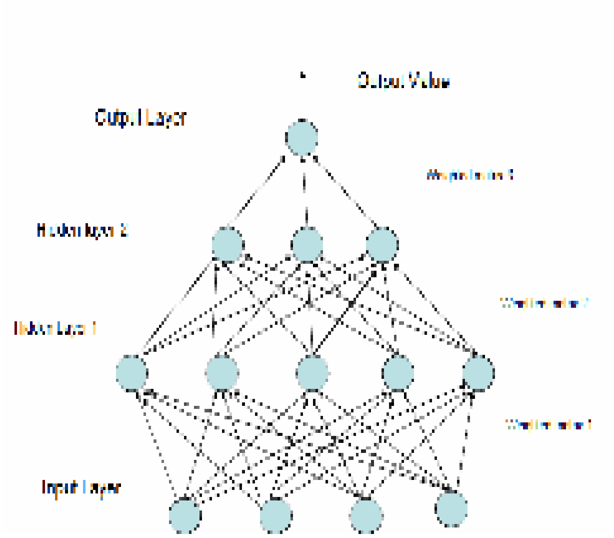


Figure 1.3 Components of a neural network

There are a number of classification methods in use but probably neural networks are most widely known. The biggest advantage of neural networks is that they can handle problems that have a wide range of parameters and are able to efficiently classify objects even if they have a complex distribution in multidimensional space. The main disadvantage of neural networks is that they are quite slow in their processing in both the training and testing phases. Another disadvantage of neural networks is that it is very difficult to determine how the net is making decisions. A simple neural network is shown in Figure 1.3.

1.4.2 Decision trees

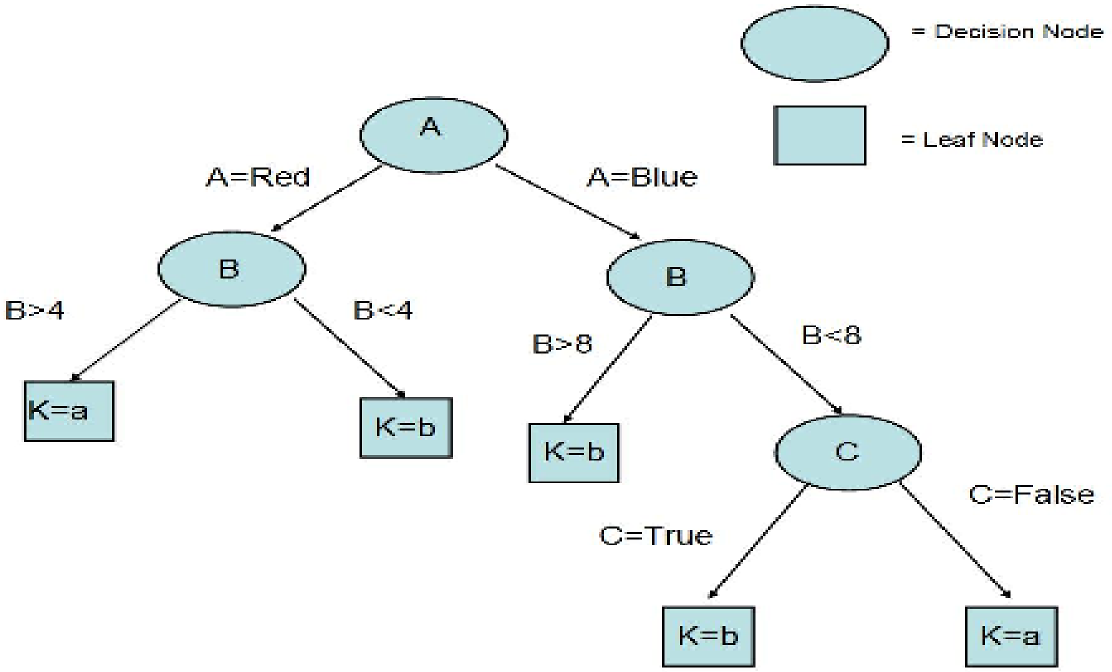


Figure 1.4 Sample decision tree

A decision tree is a predictive machine-learning algorithm that generates the target value of a sample based on various attribute values of the available data. It is a tree of various decisions as the name implies. A decision tree consists of leaves and branches where the leaves represent the classification results. The branches represent the conjunctions of the features that lead to those classification results. The technique of inducing a decision tree from data is known as *decision tree learning*. Figure 1.4 shows a decision tree which decides the value of K as a or b depending on its color and value. The disadvantage of decision trees is that they are not flexible at modeling complex parameter space distributions.

1.4.3 Nearest neighbor classifiers

Nearest neighbor classifier is a simple machine learning algorithm which is used for classification purposes based on the training samples in the feature space. In this method, the target object is classified by the majority vote of its neighbors and assigned to the class to which most of the neighbors belong. For the purpose of identification of neighbors, objects are represented by position vectors in a multidimensional feature space. The distance most commonly used for this purpose is the Euclidean distance.

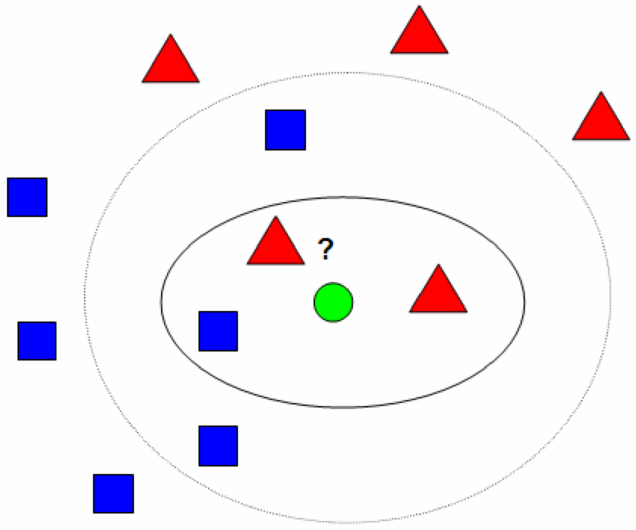


Figure 1.5 k-NN classification algorithm

In Figure 1.5, the center object is the one that has to be classified between the two classes are presented as squares and triangles. The k-NN classification algorithm takes as input the value k which represents the number of neighbors which have to be considered for the decision. Here the inner circle represents the case where k=3. Hence, the target object is assigned to the group which is represented by triangles. The outer circle represents the case where k=5. By doing so, the target object is classified as belonging to the group represented by squares.

1.5 Description of current study

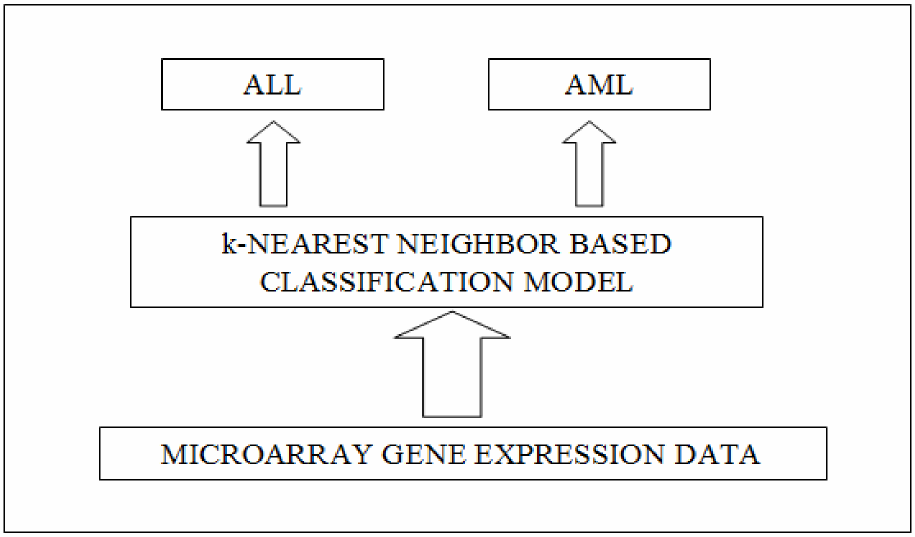

ALL
AML
k-NEAREST NEIGHBOR BASED CLASSIFICATION MODEL
MICROARRAY GENE EXPRESSION DATA

Figure 1.6 Broad overview of the classification system

In the current study, we have applied an approach based on k-NN classifier committees. Euclidean distances were calculated in all k-NN classifiers for classification purpose. The objective is to classify the data samples into two categories of leukemia, i.e. Acute Lymphoblastic Leukemia (ALL) and Acute Myeloid Leukemia (AML). For this purpose, the dataset was cleaned and further informative genes were extracted. These genes were used to recruit the best performing k-NN classifiers. The top performing k-NN classifiers were used to form a committee. This committee was then tested by using fresh data which was not used in the training of classifiers. Figure 1.6 shows the procedure followed in the study. Microarray gene expression data is used to form a committee of k-NN classifiers. This committee is further used to classify the testing data as ALL or AML. The objective of the study was to check the stability of committee k-NN classifiers.

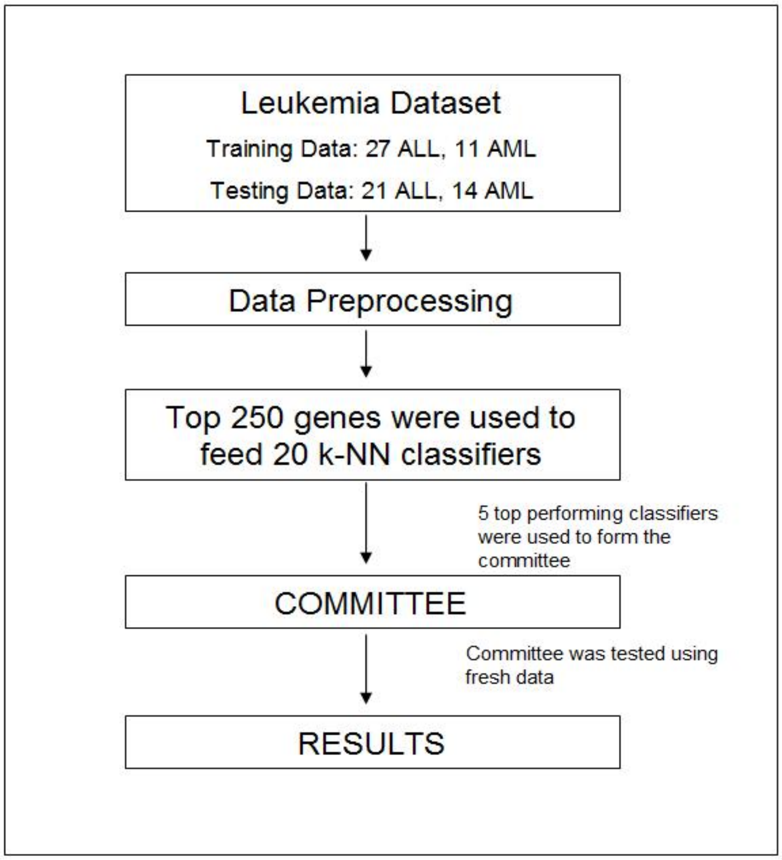

Leukemia Dataset
Training Data: 27 ALL, 11 AML
Testing Data: 21 ALL, 14 AML
Data Preprocessing
Top 250 genes were used to feed 20 k-NN classifiers
5 top performing classifiers were used to form the committee
COMMITTEE
Committee was tested using fresh data
RESULTS

Figure 1.7 Basic approach

Figure 1.7 describes the steps of the study in a broad way. The leukemia dataset is preprocessed and the informative genes obtained are used to form the committee of top performing k-NN classifiers. This committee is then used to classify samples in the testing dataset as ALL or AML.

1.6 Objectives of the study and outline of the thesis

The specific objectives of the study were to:

1. Extract the most informative genes from a selection of gene expression profiles of leukemia patients.
2. Use the identified informative genes to feed a series of k-NN classifiers each having a different architecture.
3. Recruit the top performing k-NN classifiers to form a committee.
4. Evaluate the k-NN classifier based committee using a set of fresh data for classification.

The rest of this thesis is organized as follows

1. Chapter 2 will give us detailed information on the Leukemia dataset and the previous work done on the same dataset. It also describes the process of knowledge discovery in databases (KDD).
2. Chapter 3 will provide the detailed description of the classification method used in this study.

3. Chapter 4 presents the results of our research. The major observations from the study are also discussed.
4. Chapter 5 will provide the conclusions that are inferred from this research and provides information on enhancements that can be done to this research.

CHAPER II

LITERATURE OVERVIEW

2.1 Previous work

The leukemia dataset available at the Broad Institute website [9] has been processed for classification using many different approaches. Some of the major studies conducted are listed as follows.

The study which used committee neural networks for gene expression based leukemia classification gave really good classification accuracy [10]. In this study, two intelligent systems were designed that classified Leukemia cancer data into its subclasses. The first was a binary classification system that differentiated Acute Lymphoblastic Leukemia from Acute Myeloid Leukemia. The second was a ternary classification system which further considered the subclasses of Acute Lymphoblastic Leukemia. The informative genes obtained after preprocessing were used to train a series of artificial neural networks. The networks that produced the best results were recruited to form the decision making committee. The systems correctly predicted the subclasses of Leukemia in 100 percent of the cases for the binary classification system and in more than 97 percent of the cases for the ternary classification system.

The study performed by Huilin Xiong and Xue-wen Chen was about a kernel based distance metric learning classification method based for microarray data. This paper presented a modified K-nearest neighbor (KNN) scheme which is based on an adaptive distance metric learning in the data space [11]. The distance metric, derived from the procedure of a data-dependent kernel optimization, can substantially increase the class separability of the data and lead to an increased performance as compared to the regular KNN classifier. The proposed kernel classifier method classified the leukemia data with a precision around 96% and was comparable to well known classifiers like support vector machines.

The study conducted by Dudoit et al. [12] compared the performance of different discrimination methods for the classification of tumors based on gene expression data. The methods used for the study include the k-nearest neighbor classifier method, linear discriminant analysis and classification trees. Machine learning approaches like bagging and boosting were also considered. Investigation of prediction votes was done to assess the confidence of each prediction. This study used the leukemia dataset for classification purposes. The approach was able to classify all except 3 out of 72 samples and gave an accuracy of 95.8% using the k-nearest neighbor classifier approach.

The original study of the Leukemia cancer dataset was performed by Golub etc [13]. Their study is one of the first sample classification studies that had been performed using microarray data. The microarray datasets consist of a 38-sample training dataset including 27ALL and 11 AML samples and a 34-sample testing dataset including 24 ALL and 10 AML samples. The study first identified a list of genes whose expression levels correlated with the class vector, which was constructed based on the known classes of the samples. This list of genes was considered as informative genes. The sample classification was then performed using a proposed neighborhood analysis method based on the information provided by each gene on the list. Each gene votes for the class value of an unknown sample. If the expression value of a gene in the unknown sample is closer to a group of known AML samples, the vote from this gene is AML, otherwise the class is ALL. The votes for each class were summarized; the class with majority votes was then assigned to the unknown sample. Their study verified the conjecture that there were a set of genes whose expression pattern was strongly correlated with the class distinction to be predicted and this set of informative genes can be used for sample classifications. 100% accuracy for classifying two classes was achieved. In addition to the supervised classification problem, an automatic class discovery method, self organizing maps (SOM) method, was also explored in the study. The study concluded that it was possible to classification cancer subtypes based solely on gene expression patterns.

2.2 Knowledge discovery in databases (KDD)

Knowledge discovery in databases is the process of identifying valid, novel, potentially useful, and ultimately understandable structure in data [14]. The ultimate goal of the KDD process is to extract knowledge from data in the context of large databases. In the KDD process, the flow of information can be in any direction. At any stage we can make changes and repeat the KDD process steps to achieve better results. Figure 2.1 shows the pictorial representation of the entire process.

The overall process of KDD consists of the following steps: [15]

- Understanding of the application domain
- Selection of target data (selecting a dataset based on the requirements and goal)
- Preprocessing:
    - Removal of noise and outliers
    - Collecting necessary information
    - Transforming data from one type to another type
- Data mining:
    - Selecting methods to be used for searching for patterns in the data
    - Deciding which models and patterns may be useful
- Searching for patterns of interest in a particular representation form as classification rules, decision tress, regression or clustering
- Consolidating discovered knowledge

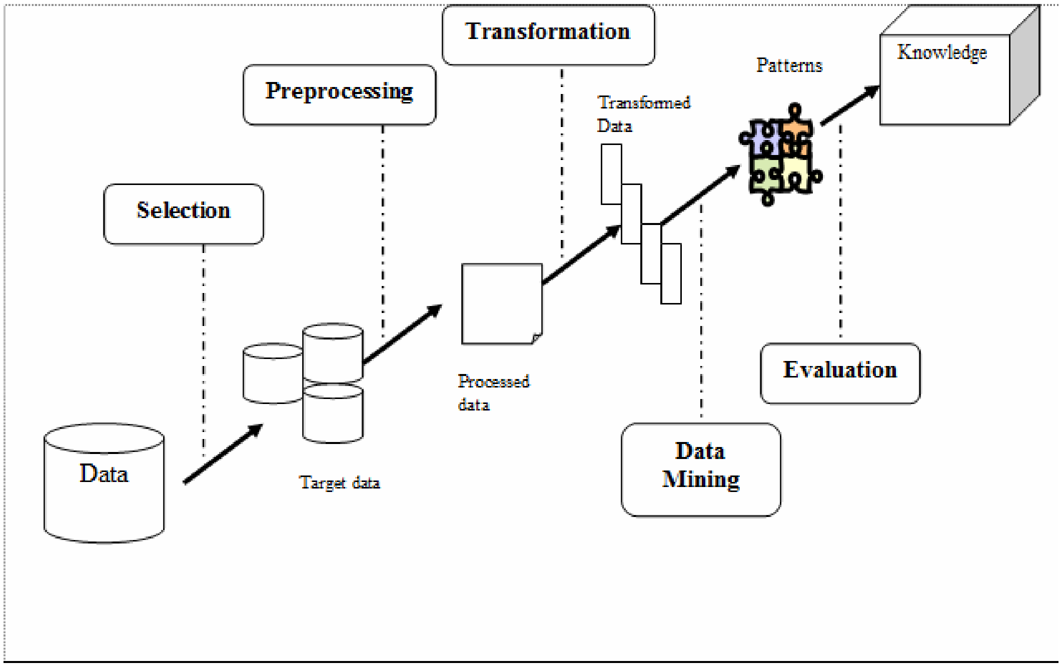


Figure 2.1 Overview of KDD process

Characteristics of KDD applications [16, 17] include

- Large data sets in terms of numbers of attributes and records;
- Attempts to deal with real world problems and data;
- Multiple access to input data;
- Use of dynamic and recursive data structures such as hash tables, linked lists, and trees;
- Size and access of the data structure is data dependent;
- Processes that consist of a number of interacting, iterative stages involving various data manipulation and transformation operations.

In the current study the KDD process has been followed to extract the informative genes from the given datasets. These genes were further processed and a classification model based on k-NN classifiers committee configured. Hence, knowledge was extracted from raw data and decisions were made.

CHAPTER III

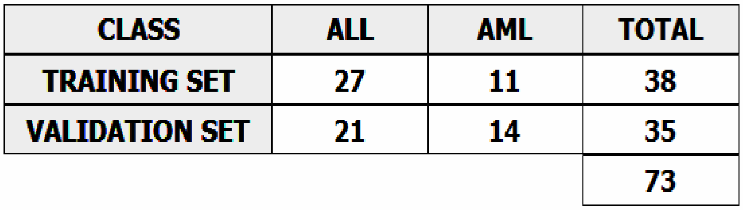
MATERIALS AND METHODS

The objective of the study is to develop a k-Nearest Neighbor based classification model which could classify the Leukemia cancer samples with maximum stability.

3.1 About the Dataset

The dataset used in the current study was obtained and uploaded to the public domain by the Broad Institute of MIT and Harvard [9]. This dataset consists of gene expression profiles from 73 patients diagnosed with Leukemia cancer. Each profile consisted of expression levels for 7129 human DNA probe sets which were spotted on high density oligonucleotide Affymetrix Hu6800 microarrays. All the samples were either from tissue samples collected from the bone marrow or from the peripheral blood. This dataset was further divided into training (38) and validation sets (35). The distribution of samples as it was used in the original study is shown in Table 3.1

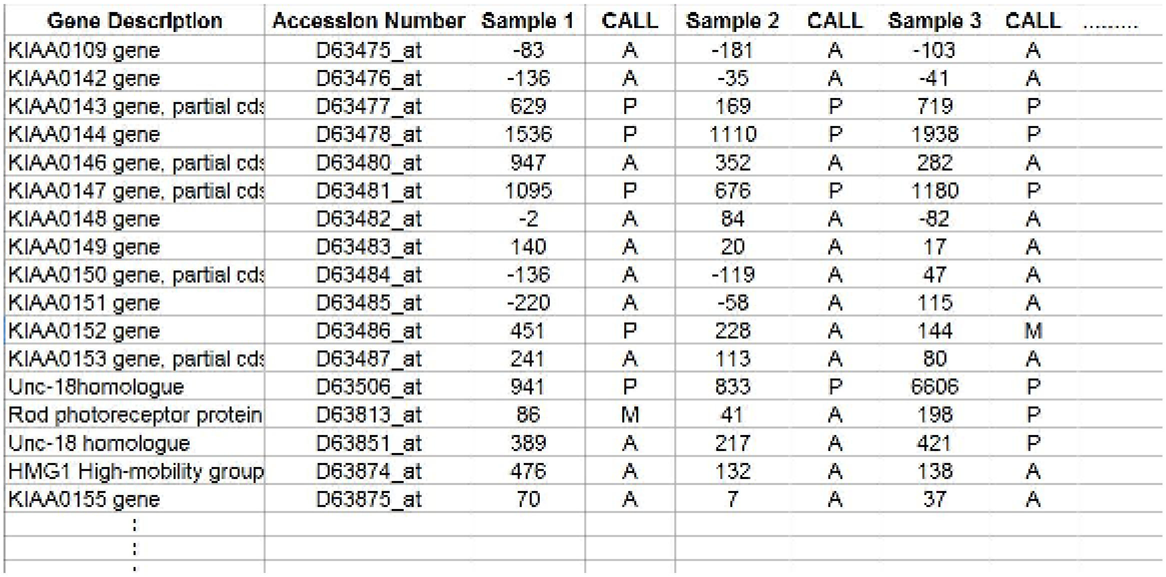
Table 3.1 Distribution of samples used in Original study



CLASS	ALL	AML	TOTAL
TRAINING SET	27	11	38
VALIDATION SET	21	14	35
			73

3.2 Format of original dataset

The dataset for all cancer patients was downloaded from the Broad Institute website in text and Microsoft Excel formats. Figure 3.1 shows a brief snapshot of the dataset. The major fields that are displayed in the dataset are the gene description, gene accession number, sample number and the call.



Gene Description	Accession Number	Sample 1	CALL	Sample 2	CALL	Sample 3	CALL	....
KIAA0109 gene	D63475_at	-83	A	-181	A	-103	A	
KIAA0142 gene	D63476_at	-136	A	-35	A	-41	A	
KIAA0143 gene, partial cds	D63477_at	629	P	169	P	719	P	
KIAA0144 gene	D63478_at	1536	P	1110	P	1938	P	
KIAA0146 gene, partial cds	D63480_at	947	A	352	A	282	A	
KIAA0147 gene, partial cds	D63481_at	1095	P	676	P	1180	P	
KIAA0148 gene	D63482_at	-2	A	84	A	-82	A	
KIAA0149 gene	D63483_at	140	A	20	A	17	A	
KIAA0150 gene, partial cds	D63484_at	-136	A	-119	A	47	A	
KIAA0151 gene	D63485_at	-220	A	-58	A	115	A	
KIAA0152 gene	D63486_at	451	P	228	A	144	M	
KIAA0153 gene, partial cds	D63487_at	241	A	113	A	80	A	
Unc-18homologue	D63506_at	941	P	833	P	6606	P	
Rod photoreceptor protein	D63813_at	86	M	41	A	198	P	
Unc-18 homologue	D63851_at	389	A	217	A	421	P	
HMG1 High-mobility group	D63874_at	476	A	132	A	138	A	
KIAA0155 gene	D63875_at	70	A	7	A	37	A	
:								
:								
:								

Figure 3.1 Snapshot of the original dataset

Table 3.2 showing the notations used in the gene expression data

NOTATION	**DESCRIPTION**
P	**Present**
A	**Absent**
M	**Marginal**

The next section explains how we classify the gene expression data obtained from the microarray as present, absent or marginal.

3.2.1 Explanation of fields

The dataset has as its first column the gene description, which gives us a brief description about the gene. The next field is the gene accession number by which we can look for the gene in any genetic database. It is just an ID for the gene. After this the samples are listed horizontally with a column called CALL next to every sample number. The CALL field actually acts as a flag that tells us whether the intensity value was due to the actual presence of a gene or noise. The oligonucleotide microarrays have pairs of probe sets for every sequence. One probe set associated with every gene is the perfect match (PM). The other is the mismatch (MM). PM is designed for the perfect matching with the target transcript while the MM measures the non-specific binding signal of partner probes. In the microarray we have 11-20 probes in the PM and MM probe sets for each gene. If the PM probe signals for a gene greatly exceed the MM probe signals for the same gene then there is a match of transcript referred to as “present”. In the other case if the MM probe signals exceed the PM then there is not a match of transcript and “absent”. The snapshot of the dataset above has the alphabets A and P for each gene in a particular patient sample. This signifies “absent” and “present” for the genes. One more case exists where the mean of PM probe signals is neither less than nor more than the mean of MM probe signals. We refer to this case as “marginal” [18, 19].

3.3 Procedure

The overview of the procedure followed in the study can be made clear by the following points:

1. The original training and testing dataset was merged to form a pool of samples.
2. A set comprised of ALL and AML samples were taken out to use at a later stage for final validation.
3. The data pool was further randomized to create 4 different datasets.
4. For each collection of training and testing dataset preprocessing was done to remove the genes that were not informative for the study.
5. Each preprocessed dataset was then further worked on to get the most informative genes ranked according to their p-values obtained from statistical t-test.
6. The most informative genes were used to feed a series of k-NN classifiers.
7. The five top performing k-NN classifiers were then used to form a committee and decide the final class of cancer samples.
8. The evaluation of the formed committee was done using fresh data, which was set aside from the data pool in the very initial phase.
9. Steps 2 to 8 were then repeated 3 times to verify the stability of the committee of k-NN classifiers.

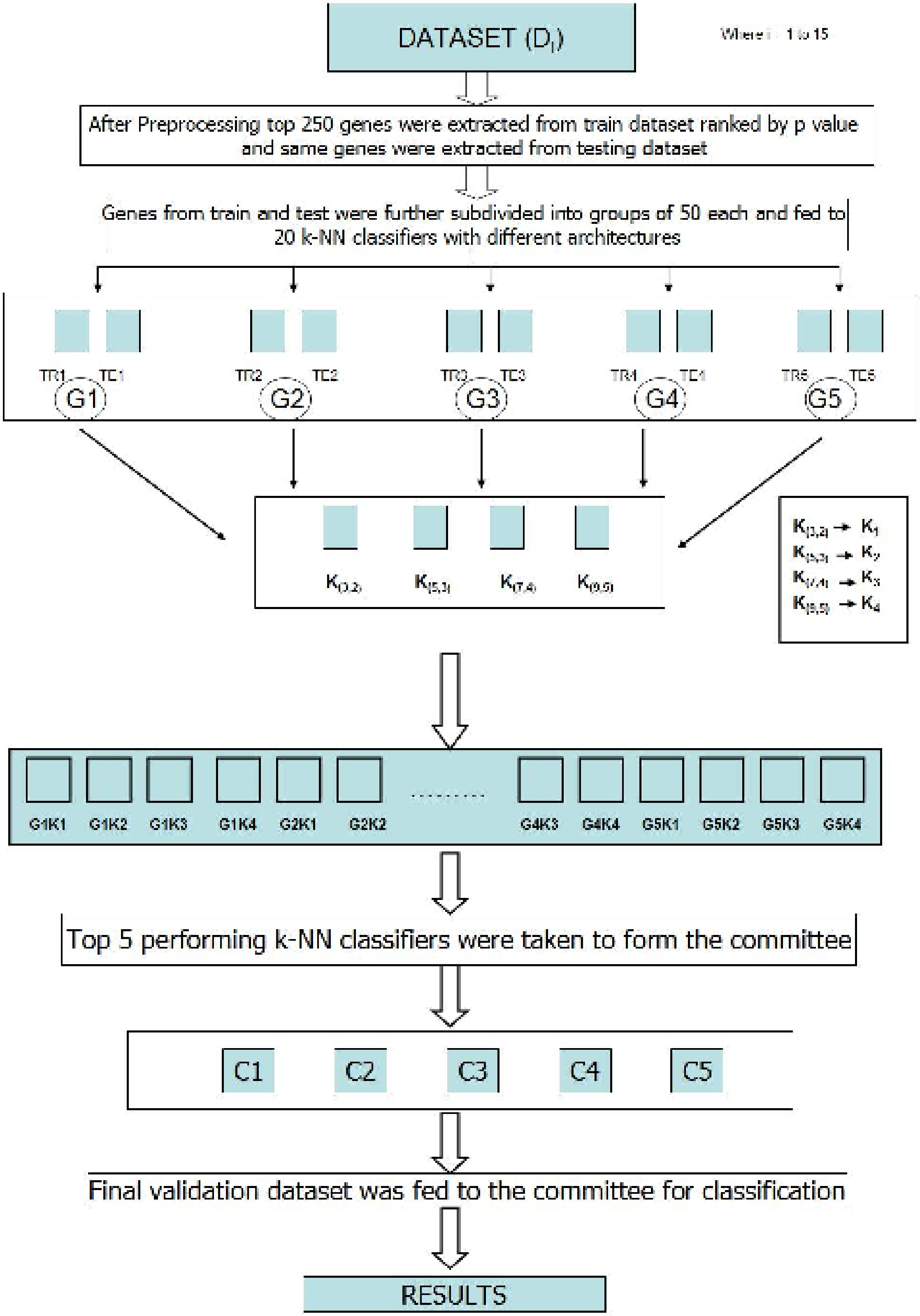

DATASET (D_i)
Where i = 1 to 15
After Preprocessing top 250 genes were extracted from train dataset ranked by p value and same genes were extracted from testing dataset
Genes from train and test were further subdivided into groups of 50 each and fed to 20 k-NN classifiers with different architectures
TR1 TE1
G1
TR2 TE2
G2
TR3 TE3
G3
TR4 TE4
G4
TR5 TE5
G5
$K_{(3,2)}$
$K_{(5,3)}$
$K_{(7,4)}$
$K_{(9,5)}$
$K_{(3,2)} \rightarrow K_1$
$K_{(5,3)} \rightarrow K_2$
$K_{(7,4)} \rightarrow K_3$
$K_{(9,5)} \rightarrow K_4$
G1K1
G1K2
G1K3
G1K4
G2K1
G2K2
....
G4K3
G4K4
G5K1
G5K2
G5K3
G5K4
Top 5 performing k-NN classifiers were taken to form the committee
C1
C2
C3
C4
C5
Final validation dataset was fed to the committee for classification
RESULTS

Figure 3.2 Flow chart showing the working of whole system

3.3.1 Dataset randomization

Training and validation data were downloaded from the Broad Institute website. The training set consisted of 38 patient samples while the validation set consisted of 35 patient samples. In order to make the experiment more robust we decided to make several random datasets from the existing patient samples. For this purpose all patient samples from both the training and validation sets were pooled together to form a big dataset of 73 patient samples. Out of these 73 samples in total we had 48 samples of patients suffering from Acute Lymphoblastic Leukemia (ALL) and 25 samples of patients suffering from Acute Myeloid Leukemia (AML).

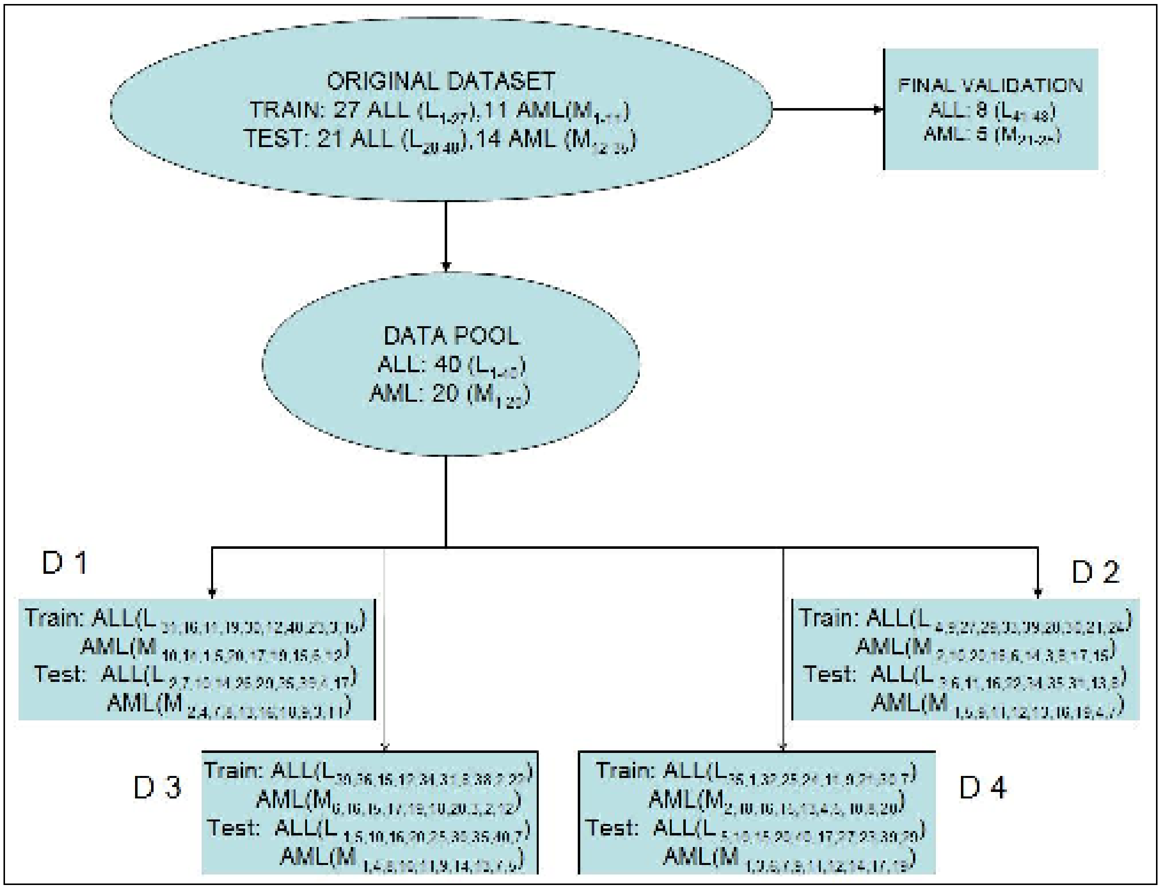

ORIGINAL DATASET
TRAIN: 27 ALL (L_{1-27}),11 AML(M_{1-11})
TEST: 21 ALL (L_{20-40}),14 AML (M_{12-25})
FINAL VALIDATION
ALL: 8 (L_{41-48})
AML: 5 (M_{21-25})
DATA POOL
ALL: 40 (L_{1-40})
AML: 20 (M_{1-20})
D 1
Train: ALL($L_{31,16,11,19,30,12,40,23,3,19}$)
AML($M_{10,14,1,5,20,17,19,15,6,12}$)
Test: ALL($L_{2,7,10,14,26,20,36,39,4,17}$)
AML($M_{2,4,7,8,13,16,10,9,3,11}$)
D 2
Train: ALL($L_{4,9,27,28,33,39,20,30,21,24}$)
AML($M_{2,10,20,18,6,14,3,5,17,15}$)
Test: ALL($L_{3,6,11,16,22,34,35,31,13,8}$)
AML($M_{1,5,9,11,12,13,16,19,4,7}$)
D 3
Train: ALL($L_{30,36,15,12,34,31,8,38,2,22}$)
AML($M_{6,16,15,17,18,10,20,3,2,12}$)
Test: ALL($L_{1,5,10,16,20,25,30,35,40,7}$)
AML($M_{1,4,8,10,11,9,14,13,7,5}$)
D 4
Train: ALL($L_{35,1,32,25,24,11,9,21,30,7}$)
AML($M_{2,10,16,15,13,4,5,10,8,20}$)
Test: ALL($L_{5,10,15,20,40,17,27,23,39,28}$)
AML($M_{1,3,6,7,8,11,12,14,17,19}$)

Figure 3.3 Detailed descriptions of datasets D1, D2, D3 and D4

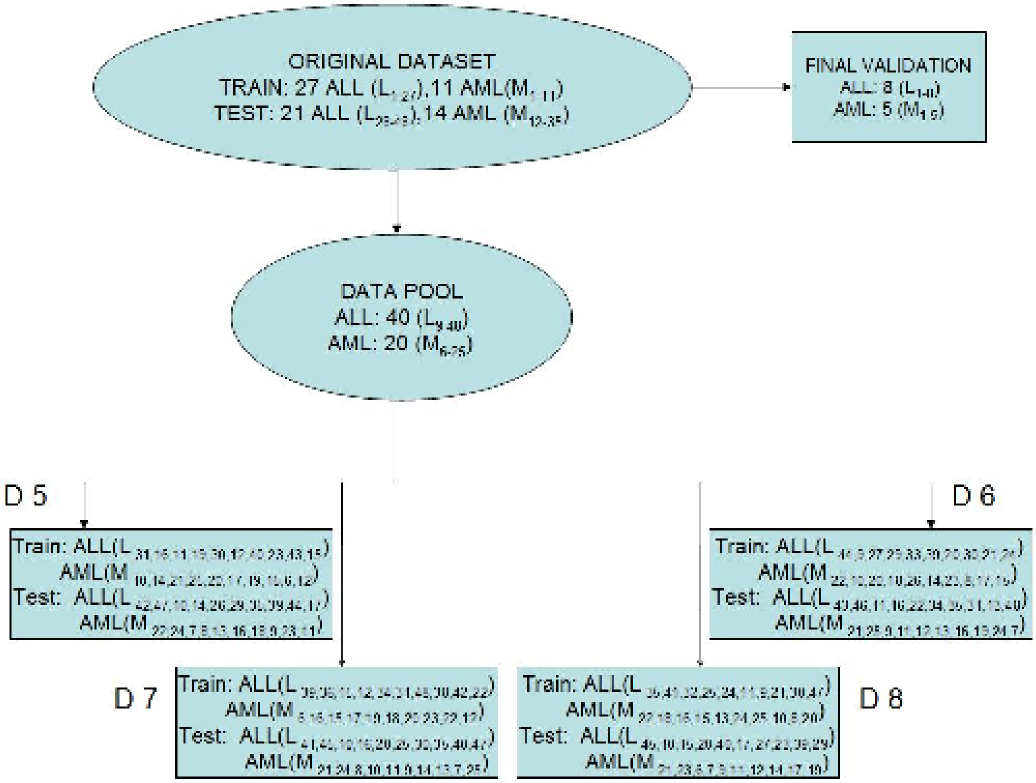

ORIGINAL DATASET
TRAIN: 27 ALL (L1-27),11 AML(M1-11)
TEST: 21 ALL (L28-48),14 AML (M12-25)
FINAL VALIDATION
ALL: 8 (L1-8)
AML: 5 (M1-5)
DATA POOL
ALL: 40 (L9-48)
AML: 20 (M6-25)
D 5
Train: ALL(L 31,16,11,19,30,12,40,23,43,15)
AML(M 10,14,21,25,20,17,19,15,6,12)
Test: ALL(L 42,47,10,14,26,29,35,39,44,17)
AML(M 22,24,7,8,13,16,18,9,23,11)
D 6
Train: ALL(L 44,9,27,29,33,30,20,30,21,24)
AML(M 22,10,20,10,26,14,23,8,17,15)
Test: ALL(L 43,46,11,16,22,34,35,31,13,40)
AML(M 21,25,9,11,12,13,16,19,24,7)
D 7
Train: ALL(L 39,36,11,12,34,31,45,30,42,22)
AML(M 6,16,15,17,19,18,20,23,22,12)
Test: ALL(L 41,46,10,16,20,25,33,35,40,47)
AML(M 21,24,8,10,11,9,14,13,7,25)
D 8
Train: ALL(L 36,41,32,25,24,11,9,21,30,47)
AML(M 22,18,16,15,13,24,25,10,8,20)
Test: ALL(L 45,10,15,20,40,17,27,23,35,29)
AML(M 21,23,6,7,9,11,12,14,17,19)

Figure 3.4 Detailed descriptions of datasets D5, D6, D7 and D8

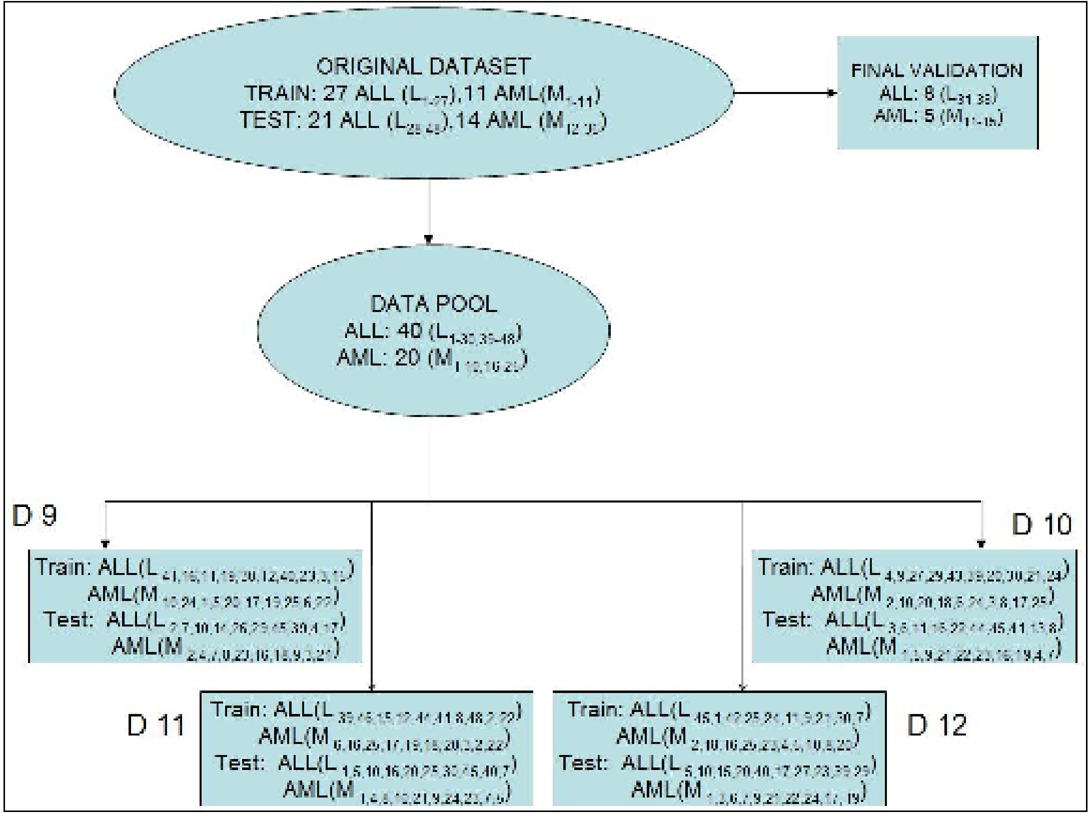

ORIGINAL DATASET
TRAIN: 27 ALL (L1-27),11 AML(M1-11)
TEST: 21 ALL (L28-48),14 AML (M12-25)
FINAL VALIDATION
ALL: 8 (L31-38)
AML: 5 (M11-15)
DATA POOL
ALL: 40 (L1-30,39-48)
AML: 20 (M1-10,16-25)
D 9
Train: ALL(L 41,16,11,19,30,12,40,23,3,15)
AML(M 10,24,1,5,20,17,19,25,6,22)
Test: ALL(L 2,7,10,14,26,29,45,30,4,17)
AML(M 2,4,7,8,23,16,18,9,3,21)
D 10
Train: ALL(L 4,9,27,29,43,39,20,30,21,24)
AML(M 2,10,20,18,5,24,3,8,17,25)
Test: ALL(L 3,6,11,16,22,44,45,41,13,8)
AML(M 1,5,9,21,22,23,16,19,4,7)
D 11
Train: ALL(L 39,46,15,12,44,41,8,48,2,22)
AML(M 6,16,25,17,19,18,20,3,2,22)
Test: ALL(L 1,5,10,16,20,25,30,45,40,7)
AML(M 1,4,8,10,21,9,24,23,7,5)
D 12
Train: ALL(L 45,1,42,25,24,11,9,21,30,7)
AML(M 2,10,16,25,23,4,5,10,8,20)
Test: ALL(L 5,10,15,20,40,17,27,23,39,29)
AML(M 1,3,6,7,9,21,22,24,17,19)

Figure 3.5 Detailed descriptions of datasets D9, D10, D11 and D12

A final validation dataset was made by picking 8 ALL and 5 AML patient samples from the original datasets. This was done in order to keep a fresh dataset for evaluating the k-NN based committee model.

After the final validation set was removed from the data pool of 73 samples we were left with 40 ALL patient samples and 20 AML patient samples. The total pool of patient samples was now comprised of 60 samples. A random number generator tool was used to randomly select 10 ALL and 10 AML samples from the data pool for each of the training and testing sets in 4 datasets. Thus, by this procedure, 4 datasets comprising of random ALL and AML samples were created.

This whole process was repeated three times by randomly selecting different final validation sets. Repeating the process, we got different sets of samples left in our new data pool. Hence, we created twelve different datasets in three rounds and final validation sets were kept aside which were used for the final committee validation.

3.3.2 Data Preprocessing

Microarray gene expression data is known to contain of a lot of noise, which can affect the performance of experiments to be conducted using the same data. Preprocessing the raw gene expression data is the technique by which we can remove the noise from the dataset. Hence, this makes preprocessing one of the most important steps to be performed in every experiment to get the most accurate results. The more we can clean the noise from the dataset the better the results will be. The preprocessing steps used in this study are the same as followed in the research done for a committee of neural networks [10].

The dataset consisted of 7129 genes from 38 patients. The initial step of preprocessing was to remove all the “endogenous control” genes from the dataset. These genes are also known as the housekeeping genes. The expression values of housekeeping genes are almost constant across all the cells. The housekeeping genes are part of each cell and perform daily cellular activities. These genes are part of all the samples and do not aid classification. Thus they are tagged as non-informative genes and are eliminated in the first step from the dataset. In the second step all genes with “absent” calls across all samples were assumed to be affected by background noise and were eliminated from consideration. Background noise is also known to affect the expression values and hence results in extreme valued outliers in the dataset. Therefore it is important to deal with these outlier values. Studies conducted earlier show that imaging equipment cannot measure intensity values above 16000. Values less than 20 are also a result of background noise [20, 21]. All of the outliers less than 20 were replaced by 20 and values more than 16000 were replaced by 16000. This way all the outliers in the dataset were dealt with. For microarray data used for classification purposes it is required that the gene expression to be classified should have differential information across the class of interest. For this purpose previous studies have used an n-fold change technique to eliminate genes where n varied from 2 to 5. For a moderate approach all genes having less than a 2.5 fold change were eliminated from the gene set [12]. After removing uninformative genes and outliers, the expression levels of the remaining genes were scaled between 1 and -1 using the following formula:

$$x_{ij}^{'} = 2\frac{x_{ij} - \min\limits_{1\leq j\leq M}\{x_{ij}\}}{\max\limits_{1\leq j\leq M}\{x_{ij}\} - \min\limits_{1\leq j\leq M}\{x_{ij}\}} - 1$$

where $x_{ij}^{'}$ is the normalized expression value of gene *i* in sample *j*, x_{ij} is the original expression value of gene *i* in sample *j*. M is the total number of samples in the dataset.

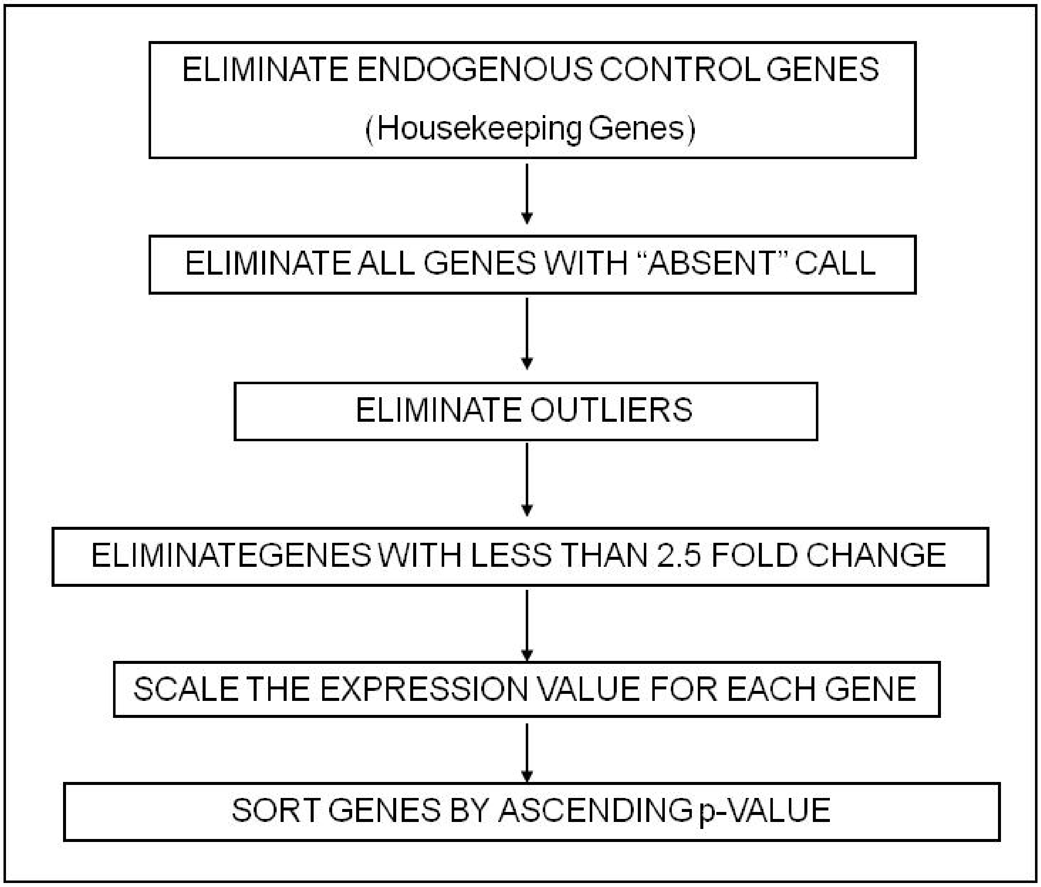

ELIMINATE ENDOGENOUS CONTROL GENES
(Housekeeping Genes)
ELIMINATE ALL GENES WITH "ABSENT" CALL
ELIMINATE OUTLIERS
ELIMINATEGENES WITH LESS THAN 2.5 FOLD CHANGE
SCALE THE EXPRESSION VALUE FOR EACH GENE
SORT GENES BY ASCENDING p-VALUE

Figure 3.6 Block Diagram showing the Data preprocessing procedure

Figure 3.6 describes the detailed steps that were performed in the preprocessing of each dataset. After preprocessing we were left with the most informative genes ranked according to p-values. Table 3.3 contains the number of genes that were left after preprocessing all 12 datasets.

Table 3.3 Number of genes left in all the datasets after preprocessing

Dataset	Number of genes after preprocessing
D 1	**4403**
D 2	**4391**
D 3	**4469**
D 4	**4390**
D 5	**4419**
D 6	**4494**
D 7	**4458**
D 8	**4472**
D 9	**4454**
D 10	**4466**
D 11	**4516**
D 12	**4390**

Preprocessing was done for the training dataset. Similar features were selected from the testing dataset as well. The outliers in the testing dataset were also replaced by the threshold values.

3.3.3 Gene selection and ranking

In the preprocessing steps the genes were ranked in increasing order of their p-values across all samples using the student t-test. This method helped us obtain the most informative genes in order of their significance. For the current study after preprocessing the top 250 genes were used to further form the committee of k-NN classifiers.

3.3.4 Committee Formation

The result of preprocessing left us with a training set having the top 250 genes and a testing dataset with the same 250 genes. The training dataset and the testing dataset were further broken into five groups of 50 genes each represented as G1, G2, G3, G4 and G5 respectively in the flow chart (Figure 3.2). Each group of 50 genes was fed to four k-NN classifiers represented by K1, K2, K3 and K4 respectively in the flow chart (Figure 3.2). The k-NN classifiers used for classification purposes were assigned different values of the input parameters k and l where k is the total number of neighbors to be considered for classification and l is the minimum number of neighbors which have to be considered in the voting process to classify the sample among different groups.

Each group of 50 genes was fed to the above listed classifiers and the classification results computed. In total we got classification results from 20 k-NN classifiers. On the basis of the accuracy of the classifiers and the probability factor the five best performing k-NN classifiers were picked, one from each group, to form the committee. In the flow chart (Figure 3.2) the committee members are denoted by C1, C2, C3, C4 and C5.

3.3.5 Committee Validation

The top performing k-NN classifiers were recruited to form a 5 member committee. After the formation of the committee the next step was to check the accuracy of the committee. For this process, we used the final validation dataset which was initially kept aside from the data pool. This was an independent dataset, which consisted of 8 ALL and 5 AML samples. The top 250 genes selected from the training were extracted from the testing dataset. The top 250 genes were further broken into 5 groups of 50 genes each. These groups were now fed to each committee member and the classification results obtained. Voting was done using the results of each committee member for each sample. The sample was classified as ALL or AML, as decided by a majority vote of the committee members. The final results were then computed by the voting mechanism. The overall motive of forming a committee was to gain better classification results for the cancer classification problem. The results show that a committee of k-NN classifiers gives more accurate and consistent results as compared to an individual k-NN classifier. The stability of the committee approach was verified through multiple runs of the procedure. Each run used a different set of samples for the final validation set.

CHAPTER IV

RESULTS AND DISCUSSIONS

4.1 Results

Each of the 12 datasets were comprised of a training set having 10 ALL and 10 AML samples and a testing dataset comprised of 10 ALL and 10 AML samples. The task was to classify the samples in the testing dataset using the k-NN classifiers.

In Table 4.1 we see the detailed results for the initial 20 k-NN classifiers for the dataset 1 in the first run. The best performing classifier was taken from each of the five groups (G1, G2, G3, G4 and G5) to form the 5 member committee. The criteria for picking the best classifier were classification accuracy and the prediction probability associated with the classification result.

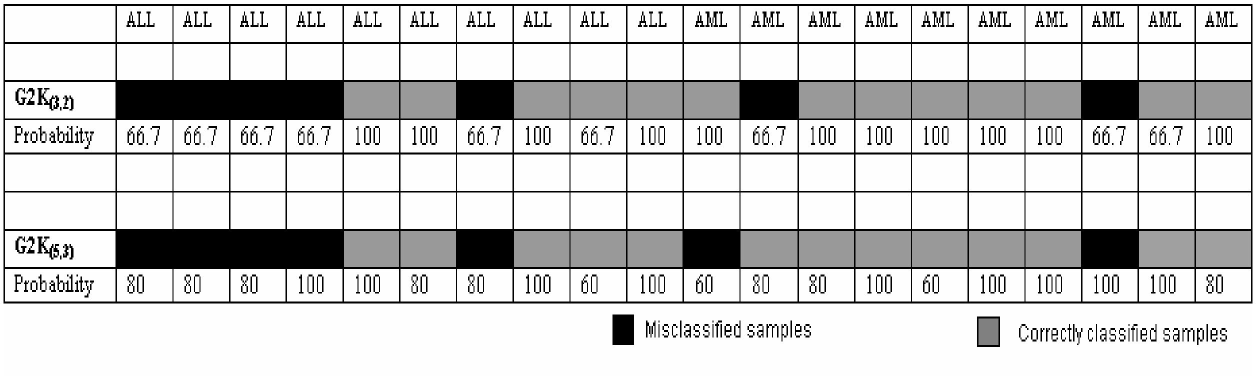
Table 4.1 Result set for dataset 1

ROUND 1 DATASET 1	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	1	1	1	1	G1 $K_{(5,3)}$
G 2	10/10	0	1	1	1	G2 $K_{(3,2)}$
G 3	10/10	0	0	0	0	G3 $K_{(5,3)}$
G 4	10/10	0	0	0	1	G4 $K_{(3,2)}$
G 5	10/10	1	1	1	2	G5 $K_{(3,2)}$

Calculation of probability of classification:

For each classification result the k-NN classifier method gave us a probability associated with it. For selecting a k-NN classifier to be a part of a committee these probabilities were considered. Those classifiers, for which the probability of correct classifications is higher than the probability of misclassification are selected as member of committee. The probabilities of all the misclassifications for both the classifiers were calculated, and the classifier having the smaller sum was selected as a committee member. In case two classifiers have the same number of misclassifications then the classifier with the higher probability of true classifications and a lower probability of misclassifications is recruited in the committee. Table 4.2 below shows the probabilities of two classifiers G2K$_{(3,2)}$ and G2K$_{(5,2)}$ with the same number of misclassifications.

Table 4.2 Selection of classifier based on probability values



	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	AML	AML	AML	AML	AML	AML	AML	AML	AML	AML
G2K$_{(3,2)}$	■	■	■	■	▨	▨	■	▨	▨	▨	▨	■	▨	▨	▨	▨	▨	■	▨	▨
Probability	66.7	66.7	66.7	66.7	100	100	66.7	100	66.7	100	100	66.7	100	100	100	100	100	66.7	66.7	100
G2K$_{(5,3)}$	■	■	■	■	▨	▨	■	▨	▨	▨	■	▨	▨	▨	▨	▨	▨	■	▨	▨
Probability	80	80	80	100	100	80	80	100	60	100	60	80	80	100	60	100	100	100	100	80

■ Misclassified samples ▨ Correctly classified samples

Both classifiers in the table have misclassified 7 samples. The classifier chosen to be a part of the committee was G2K$_{(3,2)}$ because this classifier had higher probabilities for all the true classifications and low probabilities for the misclassifications as compared to the other classifier. The probability values for the misclassifications were added for both the classifiers in consideration. The classifier which had the smaller sum was taken in the committee. In this way the probability values were used to choose the best classifier among each group for each dataset.

The next step was to check the formed committee with the final validation data which was initially kept separate. This dataset was comprised of 8 ALL and 5 AML samples. The final validation dataset was tested against the committee. The results are in Table 4.3, with the number of misclassifications shown in parentheses.

Table 4.3 Final validation of committee and result

DATASET 1	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	2(5,10)	92.3% (1 Misclassification)
G 2	C 2	1(10)	
G 3	C 3	2(5,8)	
G 4	C 4	1(10)	
G 5	C 5	3(2,8,10)	

For this dataset the committee was able to classify all 13 samples correctly. Figure 4.3 shows the classification accuracy of all the individual committee members and the committee as a whole. Similar testing was conducted for the remaining 11 datasets and the results are shown in Table 4.4 through 4.25.

Table 4.4 Result set for dataset 2

ROUND 1 DATASET 2	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	2	2	2	2	G1 $K_{(3,2)}$
G 2	10/10	5	4	4	5	G2 $K_{(5,3)}$
G 3	10/10	3	2	2	2	G3 $K_{(7,4)}$
G 4	10/10	4	5	4	3	G4 $K_{(9,5)}$
G 5	10/10	4	2	3	2	G5 $K_{(5,3)}$

Table 4.5 Final validation of committee and result

DATASET 2	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	1(10)	84.6% (2 Misclassification)
G 2	C 2	3(1,5,10)	
G 3	C 3	2(5,10)	
G 4	C 4	5(1,2,5,7,10)	
G 5	C 5	2(5,10)	

Table 4.6 Result set for dataset 3

ROUND 1 DATASET 3	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	3	3	3	3	G1 $K_{(3,2)}$
G 2	10/10	0	0	0	0	G2 $K_{(3,2)}$
G 3	10/10	2	1	1	1	G3 $K_{(5,3)}$
G 4	10/10	2	4	4	4	G4 $K_{(3,2)}$
G 5	10/10	1	1	1	1	G5 $K_{(3,2)}$

Table 4.7 Final validation of committee and result

DATASET 3	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	1(5)	100% (0 Misclassification)
G 2	C 2	2(10,13)	
G 3	C 3	2(5,6)	
G 4	C 4	2(6,7)	
G 5	C 5	0	

Table 4.8 Result set for dataset 4

ROUND 1 DATASET 4	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	2	1	1	2	G1 $K_{(5,3)}$
G 2	10/10	2	2	3	3	G2 $K_{(3,2)}$
G 3	10/10	0	0	0	0	G3 $K_{(3,2)}$
G 4	10/10	3	3	2	2	G4 $K_{(7,4)}$
G 5	10/10	3	3	3	3	G5 $K_{(5,3)}$

Table 4.9 Final validation of committee and result

DATASET 4	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	1(10)	92.3% (1 Misclassification)
G 2	C 2	3(1,5,10)	
G 3	C 3	0	
G 4	C 4	1(10)	
G 5	C 5	2(5,10)	

Table 4.10 Result set for dataset 5

ROUND 2 DATASET 5	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	1	1	1	2	G1 $K_{(3,2)}$
G 2	10/10	1	0	1	0	G2 $K_{(5,3)}$
G 3	10/10	1	0	1	1	G3 $K_{(5,3)}$
G 4	10/10	1	1	1	2	G4 $K_{(7,4)}$
G 5	10/10	0	1	3	2	G5 $K_{(3,2)}$

Table 4.11 Final validation of committee and result

DATASET 5	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	0	100% (0 Misclassification)
G 2	C 2	1(13)	
G 3	C 3	0	
G 4	C 4	1(6)	
G 5	C 5	2(6,10)	

Table 4.12 Result set for dataset 6

ROUND 2 DATASET 6	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	2	2	1	1	G1 $K_{(3,2)}$
G 2	10/10	1	1	1	0	G2 $K_{(3,2)}$
G 3	10/10	1	2	1	1	G3 $K_{(5,3)}$
G 4	10/10	1	0	0	0	G4 $K_{(7,4)}$
G 5	10/10	4	4	4	4	G5 $K_{(7,4)}$

Table 4.13 Final validation of committee and result

DATASET 6	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	1(6)	92.3% (1 Misclassification)
G 2	C 2	1(6)	
G 3	C 3	1(6)	
G 4	C 4	1(10)	
G 5	C 5	1(6)	

Table 4.14 Result set for dataset 7

ROUND 2 DATASET 7	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	3	3	3	2	G1 $K_{(9,5)}$
G 2	10/10	2	2	2	2	G2 $K_{(9,5)}$
G 3	10/10	2	2	1	1	G3 $K_{(7,4)}$
G 4	10/10	0	0	0	0	G4 $K_{(3,2)}$
G 5	10/10	2	2	3	4	G5 $K_{(5,3)}$

Table 4.15 Final validation of committee and result

DATASET 7	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	1(6)	100% (0 Misclassification)
G 2	C 2	1(3)	
G 3	C 3	0	
G 4	C 4	1(6)	
G 5	C 5	2(4,5)	

Table 4.16 Result set for dataset 8

ROUND 2 DATASET 8	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	3	4	5	5	G1 $K_{(3,2)}$
G 2	10/10	1	3	4	5	G2 $K_{(3,2)}$
G 3	10/10	5	6	6	6	G3 $K_{(3,2)}$
G 4	10/10	2	2	2	3	G4 $K_{(7,4)}$
G 5	10/10	4	5	5	5	G5 $K_{(5,3)}$

Table 4.17 Final validation of committee and result

DATASET 8	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	1(10)	92.3% (1 Misclassification)
G 2	C 2	1(4)	
G 3	C 3	1(10)	
G 4	C 4	0	
G 5	C 5	1(10)	

Table 4.18 Result set for dataset 9

ROUND 3 DATASET 9	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	2	2	2	2	G1 $K_{(7,4)}$
G 2	10/10	1	1	1	2	G2 $K_{(3,2)}$
G 3	10/10	2	1	2	2	G3 $K_{(5,3)}$
G 4	10/10	1	2	3	1	G4 $K_{(9,5)}$
G 5	10/10	3	2	2	3	G5 $K_{(7,4)}$

Table 4.19 Final validation of committee and result

DATASET 9	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	0	100% (0 Misclassification)
G 2	C 2	0	
G 3	C 3	1(2)	
G 4	C 4	0	
G 5	C 5	1(2)	

Table 4.20 Result set for dataset 10

ROUND 3 DATASET 10	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	4	4	4	3	G1 $K_{(9,5)}$
G 2	10/10	1	1	1	1	G2 $K_{(3,2)}$
G 3	10/10	4	4	3	2	G3 $K_{(9,5)}$
G 4	10/10	3	4	5	3	G4 $K_{(3,2)}$
G 5	10/10	3	3	4	4	G5 $K_{(5,3)}$

Table 4.21 Final validation of committee and result

DATASET 10	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	0	100% (0 Misclassification)
G 2	C 2	0	
G 3	C 3	0	
G 4	C 4	1(2)	
G 5	C 5	1(2)	

Table 4.22 Result set for dataset 11

ROUND 3 DATASET 11	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	1	1	1	1	G1 $K_{(3,2)}$
G 2	10/10	2	2	2	2	G2 $K_{(5,3)}$
G 3	10/10	6	5	5	5	G3 $K_{(5,3)}$
G 4	10/10	3	3	3	1	G4 $K_{(9,5)}$
G 5	10/10	3	0	0	1	G5 $K_{(7,4)}$

Table 4.23 Final validation of committee and result

DATASET 11	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	0	100% (0 Misclassification)
G 2	C 2	1(4)	
G 3	C 3	0	
G 4	C 4	0	
G 5	C 5	0	

Table 4.24 Result set for dataset 12

ROUND 3 DATASET 12	NUMBER OF SAMPLES (ALL/AML)	Misclassifications for different k-NN classifiers				Classifier selected in committee
		$K_{(3,2)}$ (ALL/AML)	$K_{(5,3)}$ (ALL/AML)	$K_{(7,4)}$ (ALL/AML)	$K_{(9,5)}$ (ALL/AML)	
G 1	10/10	1	1	3	3	G1 $K_{(5,3)}$
G 2	10/10	2	3	4	4	G2 $K_{(3,2)}$
G 3	10/10	1	1	1	1	G3 $K_{(9,5)}$
G 4	10/10	3	4	3	2	G4 $K_{(9,5)}$
G 5	10/10	4	5	4	4	G5 $K_{(7,4)}$

Table 4.25 Final validation of committee and result

DATASET 12	COMMITTEE	MISCLASSIFICATIONS	FINAL RESULT
G 1	C 1	0	100% (0 Misclassification)
G 2	C 2	0	
G 3	C 3	0	
G 4	C 4	0	
G 5	C 5	0	

4.2 Discussion

The present study represents the first application of committee k-NN classifiers for cancer classification using microarray gene expression data. The results obtained from this approach prove that a k-NN classifier committee is far more stable than individual k-NN classifiers. If we consider the heuristic nature of machine learning algorithms, the committee approach provides a highly stable result with more accuracy.

4.2.1 k-NN classifier committee members

The best performing k-NN classifiers from the initial validation were recruited to form the committee. To analyze the occurrence of repeating classifiers we have summarized the details of the committee members of all 12 datasets in Table 4.26.

Table 4.26 Overview of recruited committee members for all datasets

ROUND	DATASET	CLASSIFIERS USED TO FORM COMMITTEE				
		C 1	C2	C3	C4	C5
1	D 1	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(11,6)}$
1	D 2	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(7,4)}$	$\mathbf{K}_{(9,5)}$	$\mathbf{K}_{(5,3)}$
1	D 3	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(3,2)}$
1	D 4	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(7,4)}$	$\mathbf{K}_{(5,3)}$
2	D 5	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(7,4)}$	$\mathbf{K}_{(3,2)}$
2	D 6	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(7,4)}$	$\mathbf{K}_{(7,4)}$
2	D 7	$\mathbf{K}_{(9,5)}$	$\mathbf{K}_{(9,5)}$	$\mathbf{K}_{(7,4)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(5,3)}$
2	D 8	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(7,4)}$	$\mathbf{K}_{(5,3)}$
3	D 9	$\mathbf{K}_{(7,4)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(9,5)}$	$\mathbf{K}_{(7,4)}$
3	D 10	$\mathbf{K}_{(9,5)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(9,5)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(5,3)}$
3	D 11	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(9,5)}$	$\mathbf{K}_{(7,4)}$
3	D 12	$\mathbf{K}_{(5,3)}$	$\mathbf{K}_{(3,2)}$	$\mathbf{K}_{(9,5)}$	$\mathbf{K}_{(9,5)}$	$\mathbf{K}_{(7,4)}$

We can easily see that the most commonly occurring classifiers recruited in the committee are $K_{(3,2)}$ and $K_{(5,3).}$ The reason for this occurrence is that the most informative genes, i.e. the top 50, were used to feed the classifier $K_{(3,2)}$ and the next 50 informative genes were used to feed the classifier $K_{(5,3).}$ This is the reason we are getting consistently good classification results from these classifiers. This gives our committee a stable platform to perform consistently and makes it more reliable than any other method.

Table 4.27 Committee results for all the datasets

DATASET	C 1	C 2	C 3	C 4	C 5	FINAL COMMITTEE
1	2	1	2	1	3	1
2	1	3	2	5	2	2
3	1	2	2	2	0	0
4	1	3	0	1	2	1
5	0	1	0	1	2	0
6	1	1	1	1	1	1
7	1	1	0	1	2	0
8	1	1	1	0	1	1
9	0	0	1	0	1	0
10	0	0	0	1	1	0
11	0	1	0	0	0	0
12	0	0	0	0	0	0

4.2.2 Significance of the study

The present study was successful in demonstrating that a committee of k-NN classifiers could carry out a classification using gene expressions as input parameters. The study was the first of its kind to make use of a committee with individual Classifiers trained with different sets of input parameters. The committee decision could be considered to be more reliable as it was obtained from members with different backgrounds participating in a majority-voting scheme.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

The study demonstrates that a committee of k-NN classifiers can reform classification considering the gene expressions as input parameters. The committee decision provided a highly reliable result with more confidence than the individual classifiers.

5.1 Conclusions

Based on the results, we derive the following conclusions:

1. Original gene expression profiles with more than 7000 genes each were successfully processed to identify the 250 most informative genes which could classify one class from the other.
2. The genes were successfully utilized to train the k-NN classifiers and classify the leukemia data into two subsets.
3. The top performing classifiers were put together to form a committee and hence, achieved higher classification accuracies.
4. The k-NN classifier approach was able to classify all samples and give 100% accuracy in 7 out of 12 datasets. This shows the stability of this approach.

5. In conclusion, we can state that a committee of k-NN classifiers can help us achieve more accurate and consistent results as compared to individual k-NN classifiers.

5.2 Future work

The results of the current study give us a clear picture of the stability of the committee k-NN classifier approach for the classification of a leukemia dataset. The committee of k-NN classifier approach can be applied to several other datasets for classification purposes and the stability of the method analyzed. Results attained in a classification problem are very much dependent on the preprocessing method used for the dataset. It might be possible to increase the stability of the classification model by trying different data preprocessing techniques. We can also apply other popular methods to the same dataset and then do a comparative study with the committee k-NN classifier approach. Applying the above variations we can get more results and hence analyze the variations in the classification of samples.

REFERENCES

[1] Saifur Rahman; Bioinformatics Lab Pursues Personalized Drug Treatments; HPC wire, November 03, 2006, http://www.hpcwire.com/topic/applications/17888864.html

[2] David S. Roos; "Computational Biology: Bioinformatics--Trying to Swim in a Sea of Data", *Science* 16 February 2001:Vol. 291. no. 5507, pp. 1260 – 1261, DOI: 10.1126/science.291.5507.1260

[3] "Genetic Engineering and Biotechnology News; Spread of Cancer May Be Triggered Early in Disease Progression"; http://www.genengnews.com/news/bnitem.aspx?name=41039092; 28 Aug 2008.

[4] Definition of microarray; http://www.medterms.com/script/main/art.asp?articlekey=30712

[5] Leming Shi; "DNA Microarray (Genome Chip) --- Monitoring the Genome on a Chip", January 7,2002, http://www.gene-chips.com/

[6] Eva Paszek; "Affymetrix Chip-Basic Concepts", http://cnx.org/content/m12387/latest/oligo.gif

[7] "DNA Microarray Methodology- Flash Animation.", Department of Biology, Davidson College, Davidson, http://www.bio.davidson.edu/courses/genomics/chip/chip.html

[8] G. J. Babu & E. D. Feigelson; "Statistical Challenges in Modern Astronomy II", (New York: Springer) 1997, pp. 135-148; http://sundog.stsci.edu/rick/SCMA/node2.html

[9] Broad Institute, http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi

[10] Mihir Sewak; “Application of Committee Neural Networks for Gene Expression Based Leukemia Classification”, Master Thesis, University of Akron, 2008.

[11] Huilin Xiong, Xue-wen Chen; “Kernel-based distance metric learning for microarray data”; BMC Bioinformatics 2006, **7:**299 doi:10.1186/1471-2105-7-299**,** http://www.biomedcentral.com/1471-2105/7/299

[12] S. Dudoit, J. Fridlyand, T. P. Speed; "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data"; Journal of the American Statistical Association, 97:77-87; 2002

[13] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. BloomÞeld, E. S. Lander; “Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring”; Science; 286:531-537; 1999.

[14] Usama Fayyad; Decision Theory & Adaptive Systems
- KDD Definition and Concepts, Microsoft Research,
http://www.research.microsoft.com/~fayyad

[15] Swami Yedida; “Protein Function Prediction Using Decision Tree Technique”, Master Thesis, University of Akron, 2008.

[16] Usama Fayyad, Gregory Piatetsky-Shapiro and Padhraic Smyth; *“*The KDD Process for Extracting Useful Knowledge from Volumes of Data”, November 1996/Vol. 39, No. 11 COMMUNICATIONS OF THE ACM.

[17] Graham J. Williams, Zhexue Huang: “Modeling the KDD Process, A four stage process and four element model”, *CSIRO Division of Information Technology* February 1996.

[18] J. N. McClintick, H. J. Edenberg; “Effects of filtering by Present call on analysis of microarray experiments”; BMC Bioinformatics; 7:49; 2006

[19] E. F. Schuster, E. Blanc, L. Partridge, J. M. Thornton; “Correcting for sequence biases in present/absent calls”; Genome Biology, 8:R125; 2007

[20] K. Yang, Z. Cai, J. Li, G. Lin; “A stable gene selection in microarray data analysis”; BMC Bioinformatics; 7:228; 2006

[21] A. A. Antipova, P. Tamayo, T. Golub; “A strategy for Oligonucleotide microarray probe reduction”; Genome Biology; 3:12; 2002

APPENDICES

APPENDIX A

PERL SCRIPT USED FOR THE PREPROCESSING OF TRAINING DATASET

```
open (INPUT, "Leuk_train.res");
@train = <INPUT>;
close INPUT;
shift @train;
shift @train;
shift @train;
chomp @train;

open (INPUT2, "Leuk_train_set.cls");
@train2 = <INPUT2>;
close INPUT2;
shift @train2;

@AML_VS_ALL = split /\s+/,$train2[0]; #Array of 0's and 1's.

#prints ALL OR AML accordingly
for ($x=0;$x<$#AML_VS_ALL+1;$x++){
#if($AML_VS_ALL[$x] eq (0)){print "ALL"."\n";}
#if($AML_VS_ALL[$x] eq (1)){print "AML"."\n";}
}

print "experiments:".($#AML_VS_ALL+1)."\n";
@nonendoTrain; #creates blank array for storing non-endogenous control lines

for ($y=0;$y<$#train+1;$y++){
if (index($train[$y],"endogenous control") eq (-1) ){
@nonendoTrain=(@nonendoTrain,$train[$y]);
}

}
```

```
print "number of non endogenous lines:". ($#nonendoTrain+1)."\n"; #prints count of nonendogenous elements
@updatedTrain; #creates a blank array for storing non all-A lines


for ($d=0;$d<$#nonendoTrain+1;$d++){

@lineDataO=split /\t/,$nonendoTrain[$d];

@testArray=@lineDataO;
shift @testArray;
shift @testArray;

$testString = join '',@testArray;

$ACount= ($testString=~tr/A//);

if($ACount < ($#Tum_VS_Nor+1)) {
@updatedTrain=(@updatedTrain,$nonendoTrain[$d]);

}


}


print "number of elements w/o all a's:". ($#updatedTrain+1)." \n"; #prints count of elements without all a's

@thresholdTrain; #creates a blank array for correcting threshold values

for($z=0;$z<$#updatedTrain+1;$z++){

@lineData=split /\t/,$updatedTrain[$z];

@testArray=@lineData;
$name=$lineData[0];
$aInfo=$lineData[1];

shift @testArray;
shift @testArray;

for($r=0;$r<$#testArray+1;$r++){
$numberCount=($testArray[$r]=~tr/1234567890//);
if($numberCount ne 0){
if($testArray[$r]<20){
```

```
$testArray[$r]=20;
}
}
}

#this recomposes the data into a single element before adding it to the new array
@newLineData=($name,$aInfo,@testArray);
$newLineToString=join "\t",@newLineData;
#print $newLineToString;
@thresholdTrain=(@thresholdTrain,$newLineToString);

}


#print $thresholdTrain[$#thresholdTrain]; #last element
print "number of elements with corrected threshold: ".($#thresholdTrain+1)."\n";
#print $thresholdTrain[3]; #shows that some values have been changed to 20 from
negatives.



open OUTFILE, "> results.txt";

print OUTFILE "\t";

for ($x=0;$x<$#Tum_VS_Nor+1;$x++){
print OUTFILE "Tst",$x;
print OUTFILE "\t";

}

print OUTFILE "\n\t";

for ($x=0;$x<$#AML_VS_ALL+1;$x++){

if($AML_VS_ALL[$x] eq (0)){
print OUTFILE "(AML)";
print OUTFILE "\t"; }
if($Tum_VS_Nor[$x] eq (1)){
print OUTFILE "(ALL)";
print OUTFILE "\t"; }

}


print OUTFILE "\n";
```

```
#Final dataset eliminating the genes with less than two fold change across the
experiments
@FinalDataset;

for($z=0;$z<$#thresholdTrain+1;$z++){

@NewLineData=split /\t/,$thresholdTrain[$z];

@NewTestArray=@NewLineData;

$Newname=$NewLineData[0];
$NewaInfo=$NewLineData[1];

shift @NewTestArray;
shift @NewTestArray;

$min = $NewTestArray[0];
$max = $NewTestArray[0];

for($l=1;$l<$#NewTestArray+1;$l++)
{
$NewnumberCount=($NewTestArray[$l]=~tr/1234567890//);
if($NewnumberCount ne 0)
{
if($NewTestArray[$l]< $min){ $min = $NewTestArray[$l]}
if($NewTestArray[$l]> $max){ $max = $NewTestArray[$l]}

}
}


if($min>0){
if(($max/$min) >= 2.5){
@FinalData=($Newname,$NewaInfo,@NewTestArray);
$string=join "\t",@FinalData;
@FinalDataset=(@FinalDataset,$string);

print OUTFILE $NewaInfo,"\t";

for($k=0;$k<($#NewTestArray+1);$k++)
{
$NewnumberCount=($NewTestArray[$k]=~tr/1234567890//);

if($NewnumberCount ne 0)
{
print OUTFILE $NewTestArray[$k];
print OUTFILE "\t";

}
}
print OUTFILE "\n";

}

}



#$max = 0;
#$min = 0;

}


close (OUTFILE);

#Write to textfile
```

APPENDIX B

PERL SCRIPT USED FOR THE PREPROCESSING OF TESTING DATASET

```
open (INPUT, "Leuk.test.res");
@train = <INPUT>;
close INPUT;
shift @train;
shift @train;
shift @train;
chomp @train;


open (INPUT2, "top250.txt");
@top250 = <INPUT2>;
chomp @top250;
close INPUT2;


open OUTFILE, "> test_preprocess.txt";
print OUTFILE "\t\t";

for($t=0;$t<40;$t++)
{
print OUTFILE "TE$t","\t";
}

print OUTFILE "\n";


for ($d=0;$d<$#train+1;$d++){
@lineData = split /\t/,$train[$d];
for($r=2;$r<$#lineData+1;$r++){
$numberCount=($lineData[$r]=~tr/1234567890//);
if($numberCount ne 0){
if($lineData[$r]<20){
$lineData[$r]=20;
}
}
}


$name = $lineData[0];
shift @lineData;
$String=join "\t",@lineData;;
for ($k=0;$k<$#top250+1;$k++){
if($lineData[0] eq $top250[$k]){

print OUTFILE $lineData[0],"\t";
for($i=1;$i<$#lineData+1;$i++){
if(($lineData[$i]=~tr/1234567890//) ne 0){
print OUTFILE $lineData[$i],"\t";

}
}
print OUTFILE "\n";

}

}

}

close (OUTFILE);
```

APPENDIX C

R-CODE USED FOR IMPLEMENTING k-NN CLASSIFIERS

```
library(class)
#file is read and stored in train
train <-read.table("train1.txt" ,header=T);
#this command displays the contents in train
#train

#test file is read
test <- read.table("test1.txt" ,header=T);
#displays the test file in console
#test

cl <- c(c(rep("ALL",10), rep("AML",10)));

knn(t(train), t(test), cl, k =3, l =2 , prob = TRUE, use.all = TRUE);
```

APPENDIX D

SCHEMA AND SQL SCRIPT TO EXTRACT TOP 250 GENES FROM TRAINING DATASET

SCHEMA

```
[d1test.txt]
Format=TabDelimited
CharacterSet=OEM
ColNameHeader=True
Col1=probe_name Text width 50
Col2=ALL1 Text Width 50
Col3=ALL2 Text width 10
Col4=ALL3 Text width 10
Col5=ALL4 Text width 10
Col6=ALL5 Text width 10
col7=ALL6 Text width 10
col8=ALL7 Text width 10
col9=ALL8 Text width 10
col10=ALL9 Text width 10
col11=ALL10 Text width 10
col12=AML1 Text width 10
col13=AML2 Text width 10
col14=AML3 Text width 10
col15=AML4 Text width 10
col16=AML5 Text width 10
col17=AML6 Text width 10
col18=AML7 Text width 10
col19=AML8 Text width 10
col20=AML9 Text width 10
col21=AML10 Text width 10


[d1train.txt]
Format=TabDelimited
```

```
CharacterSet=OEM
ColNameHeader=True
Col1=probe_name Text width 50
Col2=ALL1 Text Width 50
Col3=ALL2 Text width 10
Col4=ALL3 Text width 10
Col5=ALL4 Text width 10
Col6=ALL5 Text width 10
col7=ALL6 Text width 10
col8=ALL7 Text width 10
col9=ALL8 Text width 10
col10=ALL9 Text width 10
col11=ALL10 Text width 10
col12=AML1 Text width 10
col13=AML2 Text width 10
col14=AML3 Text width 10
col15=AML4 Text width 10
col16=AML5 Text width 10
col17=AML6 Text width 10
col18=AML7 Text width 10
col19=AML8 Text width 10
col20=AML9 Text width 10
col21=AML10 Text width 10
```

SCRIPT

```
create table train
(
seqno int identity(1,1),
probe_name varchar(100),
ALL1 varchar(25),
ALL2 varchar(25),
ALL3 varchar(25),
ALL4 varchar(25),
ALL5 varchar(25),
ALL6 varchar(25),
ALL7 varchar(25),
ALL8 varchar(25),
ALL9 varchar(25),
ALL10 varchar(25),
AML1 varchar(25),
AML2 varchar(25),
AML3 varchar(25),
AML4 varchar(25),
AML5 varchar(25),
AML6 varchar(25),
AML7 varchar(25),
AML8 varchar(25),
AML9 varchar(25),
AML10 varchar(25)
)


insert into train(probe_name,ALL1,ALL2,ALL3,ALL4,ALL5,ALL6,ALL7,ALL8,ALL9,
ALL10,AML1,AML2,AML3,AML4,AML5,AML6,AML7,AML8,AML9,AML10)
SELECT * from OPENDATASOURCE
('Microsoft.Jet.OLEDB.4.0',
'Data Source=C:\thesis\;Extended
Properties="Text;HDR=Yes;FMT=Delimited"')...d1train#TXT

SELECT * into #test from OPENDATASOURCE
('Microsoft.Jet.OLEDB.4.0',
'Data Source=C:\thesis\;Extended
Properties="Text;HDR=Yes;FMT=Delimited"')...d1test#TXT


select * from train a
join #test b on a.probe_name = b.probe_name order by a.seqno
```