CLASSIFICATION OF MICROARRAY DATASETS USING RANDOM FOREST

NG EE LING

UNIVERSITI SAINS MALAYSIA
2009

CLASSIFICATION OF MICROARRAY DATASETS USING RANDOM FOREST

By

NG EE LING

Thesis submitted in fulfillment of the requirements
for the degree of
Master of Science

June 2009

Acknowledgement

Many individuals have contributed to the success of this research. These individuals consist of my supervisor, Dr. Yahya Abu Hasan, my family and friends.

I would like to express my deepest gratitude to Dr. Yahya Abu Hasan, who is the supervisor of this master's research for his continuous support. His bright advice and constructive ideas have become the main factor towards the success of this research. I also want to thank him for sharing with me the important writing techniques and for being so patient with me throughout the whole process.

Besides that, I would like to express gratitude to the Institute of Graduate Studies and School of Mathematical Sciences for granting me the entitlement of fellowship for a total of three semesters. The financial support obtained had been of great help in my studies.

I also want to thank my fellow post-graduate mates and friends who have shared their ideas with me.

Not to forget, I am grateful to my family who have consistently supported me. Their support has indeed boosted my confidence.

Lastly, I thank God for making everything possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Page

KLASIFIKASI SET DATA TATASUSUNAN MIKRO MENGGUNAKAN RANDOM FOREST

ABSTRAK

Teknologi DNA tatasusunan mempunyai keupayaan untuk memerhati lebih daripada ribuan nilai ekspresi gen dalam satu chip. Ia juga mendatangkan kebaikan dalam bidang perubatan kerana ia dapat membantu dalam pengesanan mutasi genetik dan penyakit. Kewujudan satu model yang baik dapat meramalkan kelas penyakit yang tidak diketahui sebelumnya. Untuk mendapatkan satu model yang baik, kita mesti terlebih dahulu mmperolehi keputusan klasifikasi yang baik. Namun, kebanyakan data tatasusunan mempunyai bilangan gen yang melebihi bilangan sampel. Oleh itu, untuk mendapatkan keputusan klasifikasi yang baik, bukan sahaja pemilihan jenis klasifikasi penting tetapi juga ciri penting dalam gen yang dipilih. Dalam penyelidikan ini, kita telah mencadangkan satu cara dinamakan 'stair-line' dalam pemilihan gen yang penting untuk mengurangkan kesan kurtosis yang wujud. Klasifikasi yang digunakan ialah 'Random Forest'. Lima set data tatasusunan dengan bilangan gen dan sampel yang berlainan telah digunakan untuk mempamerkan keupayaan cara 'stair-line' yang dicadangkan. Cadangan ini telah memperbaiki peratusan kebetulan dalam keputusan klasifikasi dan pada masa yang sama telah mengurangkan kesan kurtosis yang wujud dalam gen. Selain itu, pengklasifikasi yang lain juga telah dipertimbangkan dan keputusan yang diperolehi telah dibandingkan dengan keputusan yang diperolehi dengan menggunakan 'Random Forest'. Secara kesuluruhan, keputusan yang diperolehi dengan menggunakan Random Forest adalah lebih baik jika dibandingkan dengan keputusan yang diperolehi dengan menggunakan klasifikasi lain.

CLASSIFICATION OF MICROARRAY DATASETS USING RANDOM FOREST

ABSTRACT

DNA microarray technology has enabled the capability to monitor the expressions of tens of thousands of genes in a biological sample on a single chip. Medical fields can benefit from microarray data mining as it helps in early detection of genes mutation and diagnosis of disease. A well built model can be used to predict unknown disease classes in a test case. Prior to a well built model is to achieve good classification results which rely very much on the classifiers that are being used. However, in most microarray data, the number of genes usually outnumbers the number of samples. Thus, it is often not just selecting the type of classifier that is essential but also the features looked in selecting significant genes that will contribute to good classification results. Genes selection also varies from study scope and depends on the criteria researchers are looking at. In this study, we propose a stair-line method to select significant genes to reduce the effect of kurtosis found among the genes. Classification is then done using Random Forest. Five microarray datasets with different number of genes and samples are used to demonstrate the effectiveness of this method. This method improves the percentages of correct classification and at the same time reduces the effect of kurtosis in the genes expression values. Other conventional classification schemes are also looked at as a comparison to Random Forest and it is shown that the latter is one classifier that is more superior to the others. In short, Random Forest managed to give a competitive result in classifying genes correctly as Random Forest performed consistently well on all datasets.

CHAPTER 1

INTRODUCTION

1.1 Knowledge Discovery in Database

Knowledge discovery in databases (KDD) is the analysis of data. It is the practice of sorting through data to identify pattern and to establish relationship. The main reason in doing so is to discover previously unknown information that might be potentially useful in the future. With the availability of advanced mining tools which use artificial intelligence, statistical methods or pattern recognition plus the availability of the abundance of data, people are able to perform data mining on various sequences to achieve various outcomes.

There are various methods in which one can adopt to perform data mining. These methods are often recognized as the data mining parameters. Some of the often used methods which include the following:

Association - looks for patterns where one event is connected to another event

Sequence or path analysis – looks for patterns where one event leads to another later event

Classification – finding a model that describes data classes so as to use the model for future prediction

Clustering – finds and visually documents groups of fact that is not previously known

Predictions or Forecasting – discovers patterns that can lead to predictions about the future (Olson and Delen, 2008)

1.2 Microarray Data Mining

Genomic study has been of great interest over the past years. Genomic study involves gene analysis tasks which are carried out to identify and learn characteristics of genes which can lead to many hidden potential information. One of the potential information that has been looked into by the bioinformatics community is the identification of diseases. In the past, genomic study was done by looking at one gene at a time. This technique is not only tedious but also has a potential of lack of information because it is only capable of generating limited results and at a time. Now, with the advancement of microarray technology, this can be done very easily.

Microarray technology has given researchers the opportunity to perform genomic study by looking at thousands of genes simultaneously instead of just one gene at a time. This technology enables the measurement of tens and thousands of gene expressions of a biological sample in just one single chip (Samb, 2005).

Microarray data usually consists of two sections, the samples and variables or genes. Measuring gene expression using microarray is relevant to many areas of biology and medicine. The uses of microarray in the field of medicine vary and they include DNA microarray, tissue microarray, protein microarray, plant microarray and many more which adds to the reasons why microarray data is mined so widely since its existence.

Microarray data mining is indeed a very useful study as it helps in early detection of genes mutation and diagnosis of disease of which, if diagnosed early can help prevent death. Hence, microarray data mining which uses the combination of both mathematical modeling and biological technology is certainly a comprehensive way not only to classify disease but also to examine disease outcome and discover new cancer subtypes. Some recognize this field as the field of Bioinformatics.

Cancer classification has been a popular study over the past few years. Just like any other data, cancer too come in different subtypes. In classification problems, these subtypes are known as classes. Classification can therefore be done onto cancer data to build a model that can describe the classes. Previously, cancer classification is done using the most traditional method which is based on combinations of few clinical techniques. These techniques include looking at the differences of the cell shapes and detecting enzymes that are not normally produced by certain cells. The former are the clinical methods that are carried out to help diagnose cancer disease. However, studies show that not one of those tests are 100% accurate and are always inconclusive (Twyman, 2002).

Just like when mining other types of data, many challenges are faced when mining microarray data. Microarray data is one data which contains the expression levels of thousands of genes, thus increasing the difficulty level when it comes to mining the data. Secondly, microarray data usually has a very large number of variables as compared to the observed samples. And therefore efforts to achieve good results very much depend on the study scope of the researcher. Some researchers might classify good

results as obtaining good models whereby they obtain high percentage of correct classification while some chose to look at the error rates of models obtained instead. For example, Ng and Breiman (2005) decided to use Random Forest to select their first 20 important genes before they used their proposed bivariate selection method to see the interaction among genes and further reduce the number of genes to obtain better results. Nevertheless, although mining microarray data might be a wearisome task, the result obtained is often worth the effort.

1.3 Objective

Classification, which allows us to find a model that describes data classes, is the main mining method in this research. Classification not only allows us to classify genes but also to see hidden patterns especially among significant genes in order to obtain better results. Most classification schemes rely very much on selecting useful or important genes which can contribute significantly to the classification results and thus creating a good model.

The main objective of this research is to come out with good classification accuracy (high percentage of correct classification of genes) by identifying smaller set of genes. We propose a stair-line method to select significant genes. Basically, our stair-line method involves three steps which consist of first selecting significant genes using Random Forest, second eliminating genes with odd platykurtic behaviour and third re-select top 20 significant genes with highest t-values. Here, we define significant genes as those genes which are well differentially expressed. Lastly, classification is then done

using Random Forest classifier of which its error function has been modified to reduce the effect of kurtosis.

1.4 Methodology

Data mining tasks vary from one study to another. The fundamental stages that are involved in data mining include pre-processing, processing and post-processing. The initial data mining task in our research involves selecting significant genes to reduce the effect of kurtosis found among the genes. While many other researchers have chosen to work at specific algorithms, we have chosen to look at the effect of the statistical measurement kurtosis instead as this is an area which has not much been emphasized on. Teschendorff et. al. (2006) used kurtosis behaviour found among genes as a clustering method.

In our paper, we have proposed the stair-method which consists of a total of three steps in selecting significant genes before classification is done to reduce the kurtosis effect found among genes. While we could have only used Random Forest classifier to select important genes, we want to bring in the importance of distribution in genes and show that the selection of significant genes does not necessarily involve only one or two steps but three as shown in our research. However, we have also proven that the three steps chosen synchronized well with each other, giving good and reasonable results.

Once a raw data has been obtained, it must go through certain steps of data cleaning before it can be processed. Thus, the data cleaning process, often referred to as

pre-processing stage is absolutely vital as it is the initial stage to start the whole data mining processes. As microarray is normally a large dimensioned data, pre-processing is usually not an easy task. In our study, we have introduced a stair-line method to choose the significant genes. This stair-line method will be done using scripts written in Mathematica, a computer algebraic system.

Once a raw data has been pre-processed or cleaned, mining methods can be applied onto it. As mentioned earlier, the mining method used for this research is classification. Besides using the Random Forest classifier, the use of a powerful data mining software, WEKA (Waikato Environment for Knowledge Analysis), and the readily available several classification algorithms in WEKA will also be used to build our classification models. These algorithms include J48, ZeroR, k-nearest neighbour, Naïve Bayes, support vector machine and neural network and will be used as a comparison to the Random Forest classifier.

1.5 Summary of Contribution

In this research, we have proposed an alternative method to select significant genes, which is by looking at the genes' distribution. Normal procedure of selecting significant genes usually involves only one or two steps. Tibshirani et. al. (2002) who has created an approach known as nearest shrunken centroids to identify subsets of genes that best characterize each class in classifying the blue-cell tumor. However, their research was only validated by the blue-cell tumor and leukemia dataset which could possibly mean that the method might not work well for the other cancer datasets. An

example on research on the usage of two algorithms was done by Li et. al. (2002). In their paper, a Bayesian method which performed similarly to that of support vector machine's algorithm was used. Nevertheless, they also limited their research to only three datasets and have also not clearly proven their Bayesian method's superiority over the other methods.

Our proposed stair-line method has three steps and all these three steps synchronize well with each other. We have also opted to use five datasets instead to show the superiority of our proposed method. Our methods deviate from the conventional way of selecting significant genes. Our combination of steps looked at both genes' distribution as well as how the genes are differentially expressed. Initial experiment showed that these genes generally have a negative kurtosis value or are of platykurtic distribution although there are some outliers. After omitting the outliers and selecting genes which are differentially expressed using a t-test, we reduce the effect of kurtosis by modifying the error function in the Random Forest classifier. Our study has also successfully proven that Random Forest is a versatile classifier yet robust enough to handle highly dimensioned data such as the microarray data.

1.6 Thesis Summary

This thesis has six chapters. It starts with the introduction chapter which gives a brief but precise explanation on microarray data mining and its issues that motivate this research. The introduction also tells the study scope and the layout of our research.

In chapter two, we highlight the importance of data mining and different methods of data mining. Besides, the statistical measurement and the classification techniques used in this study are also explained. We also discuss about the other classification methods which are the J48, ZeroR, k-nearest neighbour, Naïve Bayes, support vector machine and neural network that are being used in this study as a comparison to our main classifier, Random Forest.

Chapter three introduces the technology of microarray. This chapter focuses on introducing the fundamental of microarray including the process and its connection with human genes. We also highlight the common issues faced in microarray data mining and past researches that have been done in this field.

The following chapter which is chapter four discusses the implementation of our experiments. Here, we introduce our datasets in detail and explain how our data is being prepared using our proposed method which is the stair-line method before classification is done.

Chapter five presents and discusses our results. Results are discussed in details and graphs and tables are used to show a better representation of the results.

The last chapter is the conclusion of the thesis. In this chapter, we recapture the purpose of this study as well as our objective and motivation.

CHAPTER 2

DATA MINING AND TECHNIQUES OF CLASSIFICATION

2.1 Data

2.1.1 Collection of Data

Data and information of different forms are created and stored each day. These data are collected for a variety of reasons. We are indeed overwhelmed with the amount of data in the world, and this amount seems to be increasing with no end in sight. Some data are so huge that it requires computers with larger memory capacity to handle them. Hence, it is almost impossible to imagine having such data handled manually without the help of computer technology.

The phenomenon of data-handling is actually closely related to the development of the computer technology. Computers have now made it easy to save and store information. There are a lot of advanced tools that are available to store data. Examples of such tools that are available are Structured Query Language (SQL) and Oracle or Microsoft Access. With the availability of these tools, we can store whatever data we want in a clearer form with the additional benefit that this data can be retrieved anytime and anywhere and in a much convenient way.

However, while having to store data efficiently is important, good human skills are essential when it comes to understanding the data that is being stored. Most of the time, people tend to lack the skill in understanding the data collected and might eventually not be able to interpret the collected data properly. As there might be hidden information in the data that can be potentially useful, the former is

definitely a serious problem. Therefore, knowledge discovery is introduced in the hope to solve this and other matters arising that are connected closely to data-understanding.

2.1.2 Data and Its Quality

There are many forms of data and they usually come in different dimensions. While some expects a large data to contribute to more new findings, it also requires far more complicated methods to handle the data.

Microarray data for example, is one data that is very large in dimension and contains more number of variables (genes) as compared to the number of samples or observations. Hence, carrying out analysis on the data is definitely going to require more time and effort.

Besides, it is vital to have an overview of the type of data we are mining. To do this, the data's pattern must be evaluated so as to obtain a clearer picture of the data which can then enhance the process of data mining.

Looking at the data's pattern involves the usage of statistics. Spiegel (1999) mentioned that statistics is a scientific method relating to the collection, analysis, summarization, and explanation of data. There are many ways to look at the pattern of a data which include investigating on the measures of central tendency which involve the calculation of mean, median and mode and measures of dispersion which involve the calculation of first quartile, third quartile, variance and standard deviation. Some also consider the data's maximum and minimum values to help

locate any outliers in the data. Missing values is another problem that is commonly faced especially when handling real-life data. Depending on which variables these values are missing on, researchers will conduct necessary steps, either to substitute the missing values with a certain average point or to disregard the whole variable This again depends on the researchers' scope of study.

Unlike the commonly used statistical measurement like the measure of central tendency and measure of dispersion, the measurement of kurtosis is one criterion we looked at in this study.

Kurtosis is the degree of peakedness of a distribution. Mathematically, kurtosis is the normalized form of the fourth order central moment of a distribution. A high kurtosis value means a higher variance which is due to the infrequent extreme deviations, as opposed to the frequent modestly sized deviation. Kurtosis is useful to characterize the characteristics of a distribution (Pearson, 2005). Unlike skewness which can be easily seen from a box plot, kurtosis is often not as easily detected. Nevertheless, kurtosis can be calculated by using $\frac{\sum (x-\mu)^4}{N\sigma^4} - 3$ whereby, x is the value of a point, μ represents average and σ represents standard deviation of the data. An approximate standard error to compensate the existence of non-zero kurtosis is given by $e = \sqrt{\frac{24}{n}}$ (Crawley, 2005).

2.2 Data Mining

Data mining has become a powerful technology in different fields. The term data mining was used to describe the component of the Knowledge Discovery in Databases (KDD) process where the learning algorithms were applied to the data and can be defined as the process of selection, exploration and modeling of large quantities of data to discover models and unknown patterns (Giudici, 2003).

Data mining is a whole process of data extraction and analysis to achieve specified goals. Data mining is different from data retrieval because it looks for relations between phenomena that are not known beforehand. So, in short, data mining is about solving problems by analyzing data that is already present in the databases (Olson and Delen, 2008).

Data mining uses techniques such as artificial intelligence, statistics and pattern recognition. Data mining methodologies include:

Association - looking for patterns where one event is connected to another event

Sequence or path analysis - looking for patterns where one event leads to another later event

Classification - looking for new patterns

Clustering - finding and visually documenting groups of facts not previously known

Forecasting - discovering patterns in data that can lead to reasonable predictions about the future.

A complete data mining process comes in three steps which are the pre-processing, processing and the post-processing. The pre-processing step is often

known as the feature selection step whereby researchers reduce the number of variables by getting rid of noisy and irrelevant ones. Clustering and classification are types of processing method where the former is unsupervised and the latter is supervised. Forecasting on the other hand is a post-processing task.

While there are many data mining methodologies available, classification is probably the oldest and most widely-used of all when it comes to mining microarray data and will be used throughout our study. There are a few classification techniques which will be used in this study apart from our main concern which is the Random Forest. Those classification techniques are ZeroR, J48, Naïve Bayes (NB), k-nearest neighbour (KNN), support vector machine (SMO) and neural network (MLP).

2.3 Classification and Its Techniques

2.3.1 Random Forest

Random Forest is an algorithm that is able to compute a collection of single classification trees. Random Forest is a classification algorithm developed by the late Leo Breiman in 2001.

Random Forest creates a forest-like classification. The basic algorithm in Random Forest works in such a way that each tree is constructed using a different bootstrap sample built from the original data. The each tree that is built is grown to the fullest without any pruning. The bootstrap data points is a random sample of size n drawn with replacement from the sample (x_1, ..., x_n). This means that the bootstrap data set consists of members of the original data set, some appearing zero times, some appearing once twice, etc (Efron and Tibshirani, 1997). The whole bootstrap

procedure is repeated several times, with different replacement samples for the training set and the result is then averaged.

The bootstrap sample usually consists of about two-thirds of the data. The other one-third or out-of-bag (oob) case will then be used as the 'test' set to get the classification result. Classification is done by getting the majority vote (particular class) of each 'test' set in a certain collection (Breiman, 2001).

Random Forest has its own variable (genes) selection procedure. The number of votes cast for the correct class is counted after each out-of-bag case is put down in each tree grown in the forest. The values of the mth variable in the oob cases are then permuted and put down the trees. The difference between the correct votes cast for the variable-permuted data and the untouched data is calculated by subtracting the former from the latter. The raw importance score for the mth variable is the average over all trees in the forest.

Random Forest is a good classifier because it gives competitive results in accuracy among current algorithms. Besides, it has the capacity to run efficiently on large data which means it can handle thousands of input variables and this is definitely an important feature in our study as we dealt with microarray data which contains thousands of variables.

2.3.2 Decision Tree (J48)

Decision tree is derived from the simple divide-and-conquer algorithm. The most common algorithms of the decision trees are C4.5 and ID3. The attractiveness

of decision tree is its easy and convenient representation whether in visualization or in rules that can readily be expressed so that human can understand them (Gamberger et. al., 2001).

J48 is one classifier that is implemented based on the concept of decision tree and uses the C4.5 algorithm. It generates pruned and un-pruned C4.5 algorithm decision tree. C4.5 allows pruning of the resulting decision tree. Although pruning tends to increase the error rates on the training data, more importantly, it can decrease the error rates on the unseen test cases (Witten and Frank, 2000).

The decision tree algorithm works by first selecting an attribute to be the root node and make a branch for each possible value. So, this will split the instances into subsets. When all instances at a node have the same classification, the tree will stop splitting. In short, the decision tree is a classifier that works in the form of a tree structure (Gamberger et. al., 2001). Figure 2.1 shows a visualization of a tree structure classifier.

On the whole, J48 can be considered as a good classifier as it is able to deal with numeric attributes, missing values and noisy data. Nevertheless, the drawback is that only one attribute is used to split the data into subsets at each node of the tree. Besides that, J48 usually only performs better with binary-class data as compared to multi-class data.

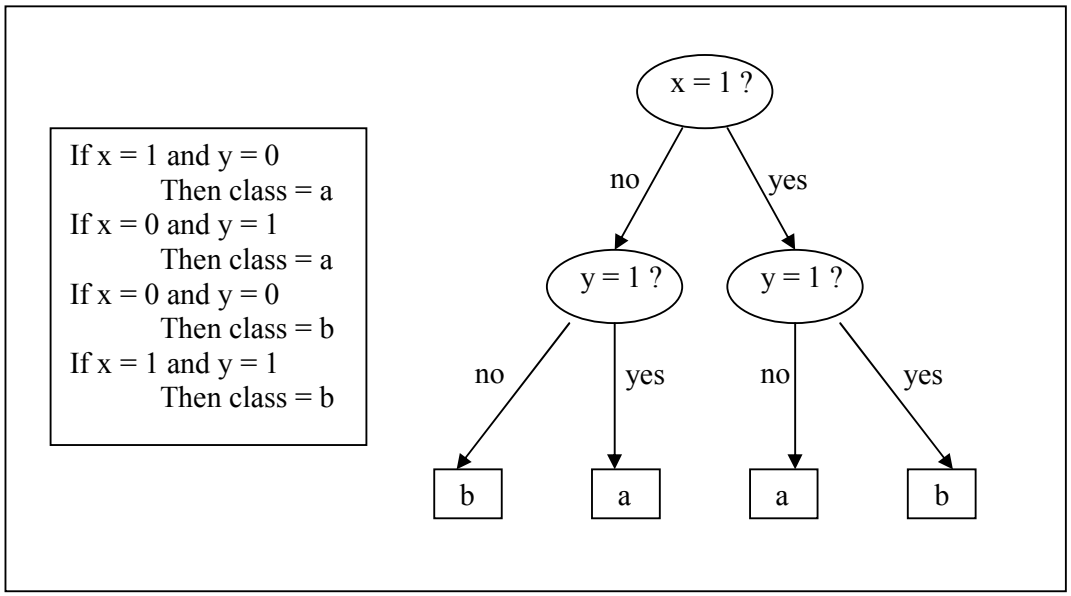

Figure 2.1: Visualization of a Tree

2.3.3 Bayesian Theorem - Naïve Bayes

The Naïve Bayes Classifier technique is based on Bayesian theorem. The Naïve Bayes classifier has been successfully applied in a number of machine learning applications. It is constructed by using the training data to estimate the probability of each class given the gene expression of the new sample. The Naïve Bayes model makes additional assumption that the values for each attributes are independent (Aas, 2001).

Naïve Bayes is particularly appropriate when the dimensionality of the independent space i.e., number of input variables is high. For the reasons given above, Naïve Bayes can often outperform other more sophisticated classification methods (The Statistics Homepage, 2003). The Bayesian Theorem is given

by $P(H \mid D) = \frac{P(D \mid H)P(H)}{P(D)}$. Generally the problem is to find the hypothesis H that best explains the data D.

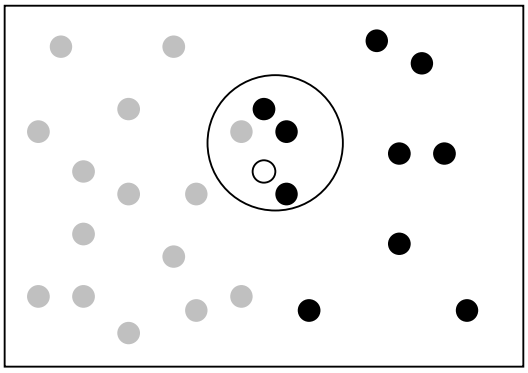

Figure 2.2: Classifying a New Object

In order for us to demonstrate the concept of the Naïve Bayes classification, consider the example shown in Figure 2.2. There are both grey and black objects. Our task is to classify the new object which is the white object (namely X). Since there are 15 grey objects and only 10 black objects in the figure, it is logical to believe that the new object is likely to have membership of grey rather than black. In the Bayesian analysis, this belief is known as the prior probability (The Statistics Homepage, 2003). So, the prior probabilities for grey circle and black object are:

$$\text{Prior Probability for grey object} = \frac{\text{Number of grey objects}}{\text{Total number of objects}} = \frac{15}{25}$$

$$\text{Prior Probability for black object} = \frac{\text{Number of black objects}}{\text{Total number of objects}} = \frac{10}{25}$$

We assume that the more grey (or black) object around X, the more likely that X belongs to that particular colour. So, in order for us to measure that, we draw a circle around X which encompasses a number of points irrespective of their colour labels. Then we calculate the number of objects in the circle belonging to each class label. From this we calculate the likelihood:

$$\text{Likelihood of X given grey} = \frac{\text{Number of grey objects in the circle}}{\text{Total number of grey objects}} = \frac{1}{15}$$

$$\text{Likelihood of X given black} = \frac{\text{Number of black objects in the circle}}{\text{Total number of black objects}} = \frac{3}{10}$$

Although the prior probability indicates that X may belong to grey but the likelihood indicates otherwise. In the Bayesian analysis, the final classification is produced by combining both sources of information, i.e., the prior and the likelihood, to form a posterior probability using the so-called Bayes' rule (The Statistics Homepage, 2003).

Posterior probability of X being grey

= Prior Probability for grey object × Likelihood of X given grey

$$= \frac{15}{25} \times \frac{1}{15} = \frac{1}{25}$$

Posterior probability of X being black

= Prior Probability for black object × Likelihood of X given black

$$= \frac{10}{25} \times \frac{3}{10} = \frac{3}{25}$$

Finally, we classify the X as black because it achieves the highest posterior probability.

From the above visualization, we can conclude that it is one classifier that is easy to comprehend. Naïve Bayes also easily handles missing values by simply omitting single attribute probabilities for each class. However, as the attributes of most of the datasets available are usually not all independent, this contradicts with Naïve Bayes' assumption and might affect the performance of this classifier.

2.3.4 K-Nearest Neighbours (KNN)

In this classification technique, a new variable with an unknown label is assigned the label of the variable in the training set which is nearest and similar. The nearest neighbour algorithm is extremely simple and is used in many applications. The similarity may be measured using distance measures which include Euclidean distance, Euclidean squared distance, Manhattan distance (also known as City-block distance or taxi-cab distance), and Chebychev distance.

While nearest neighbour refers to the nearest neighbour or 1 nearest neighbour, *k*-nearest neighbour or KNN refers to the *k*th nearest neighbour. Apart from that, KNN is a more robust method that classifies data points by looking at more than just the nearest neighbour. KNN is a memory-based method. That, in contrast to other statistical methods, requires no training. It functions on the intuitive idea that close objects are more likely to be in the same category. Thus, in KNN, predictions are based on a set of prototype examples that are used to predict new or unseen data based on the majority vote (The Statistics Homepage, 2003).

The KNN classifier in WEKA uses the Euclidean distance, D which is the distance measured between the sample with the gene values $a_1^{(1)}$, $a_2^{(1)}$, ..., $a_k^{(1)}$ (where k is the number of attributes) and one with values $a_1^{(2)}$, $a_2^{(2)}$, ..., $a_k^{(2)}$.

The formula is given by $D = \sqrt{(a_1^{(1)} - a_1^{(2)})^2 + (a_2^{(1)} - a_2^{(2)})^2 + ... + (a_k^{(1)} - a_k^{(2)})^2}$

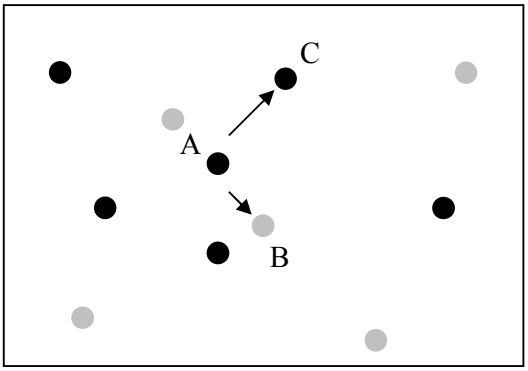

Figure 2.3: The Distance Between A-B and A-C

On the other hand, when nominal variable are present as shown in Figure 2.3, it is necessary to come up with a "distance" between different values of that variable. In this case, we have to calculate the distance between the black dots and the grey dots as seen in Figure 2.3. Usually a distance of 0 is assigned if the values are identical, otherwise the distance is 1. Thus, the distance between black and black is 0 and that between black and grey is 1.

If the value of k becomes very large, then the classification will become all the same – simply classify each attribute as the most numerous class. For this study, we will use k=4.

The KNN classifier is user-friendly and gives optimal results by numeric data. However, the weakness of this classifier is its large computing power

requirement, since the distance to all the objects in the dataset has to be calculated in order to do classification and the database also can be easily corrupted by noisy exemplars, which are the already-seen instances that are used for classification (Ye, 2004).

2.3.5 Support Vector Machine - Sequential Minimal Optimization (SMO)

Support vector machine (SVM) is a linear modeling that is used for classification and instance-based learning. It is based on the maximum margin hyperplane. SVM selects a small number of critical boundaries called support vector from each class and builds a linear discriminate function that separates them as widely as possible. Support vector is a set of points in the feature space that determines the boundary between objects of different class memberships. It transforms the instance space into a new space. With a nonlinear mapping, a straight line in the new space does not look straight in the original instance space. A linear model constructed in the new space can represent a nonlinear decision boundary in the original space (Witten and Frank, 2000).

If there is a two-class dataset whose classes are linearly separable; that is, if there is a hyperplane in instance space that classifies all training samples correctly then the maximum margin hyperplane is the one that gives the greatest separation between the classes.

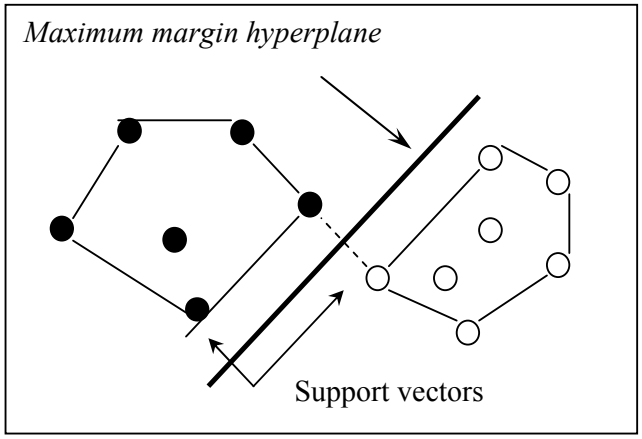

Figure 2.4: A Maximum Margin Hyperplane

In Figure 2.4, two classes are represented by open and filled circle. We connect each circle of the same class and two polygons are created. Since we assumed that the two classes are linearly separable, it cannot overlap each other. Among all hyperplanes that separate the classes, the maximum hyperplane is considered to be the one that is as far as possible from both the polygons that are built. The equation of the hyperplane separating the two classes can be written as $x = w_0 + w_1 a_1 + w_2 a_2$ with a_1 and a_2 as the variable values and w as weights to be learned.

The instance that is closest to the maximum margin hyperplane is the one with minimum distance and it is called the support vector. There is always at least one or more support vector for each class (Witten and Frank, 2000).

There are many methods to train SVM. One particularly simple method is Sequential Minimal Optimization (SMO) which is what we will be using in WEKA. Nevertheless, SMO is often slow to converge to a solution, particularly when the data

is not linearly separable in the space spanned by the nonlinear mapping. This situation increases with noisy data (Witten and Frank, 2000).

2.3.6 Neural Network - Multi Layer Perceptrons (MLP)

Neural network has been successfully applied in many areas. Indeed, neural network can be seen anywhere especially when it comes to problems like prediction, classification or control. Basically, neural network is so popular because of its powerful algorithm and the fact that it is easy to use. In addition, neural network is nonlinear and is also a very sophisticated modeling technique which is able to model an extremely complex function. However, the algorithm is also not as easily comprehensible as the others and is often called the black box.

The basic neural network consists of neurons. A neuron receives a number of inputs either from the original data or from the output of other neurons in the neural network and each of the input comes via a connection that has a strength or weight. Each neuron also has a single threshold value. The weighted sum of the inputs is formed, and the threshold is subtracted to compose the activation of the neuron. The activation signal is then passed through an activation function to produce the output of the neuron (The Statistics Homepage, 2003).

A simple network, as shown in Figure 2.5, has a feedforward structure: signals flow from inputs, forwards through any hidden units, eventually reaching the output units. A typical feedforward network has neurons arranged in a distinct layered topology. The input layer is not really neural: these units simply serve to introduce the values of the input variables. The hidden and output layer neurons are

each connected to all of the units in the preceding layer (The Statistics Homepage, 2003).

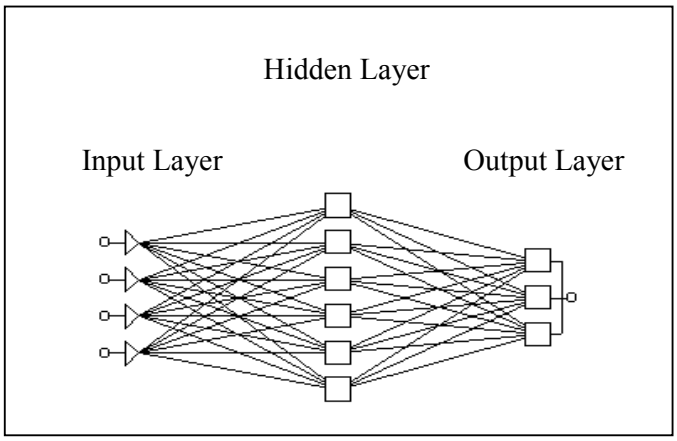

Figure 2.5: A Basic Artificial Model

When the network is used, the input variable values are placed in the input units, and then the hidden and output layer units are progressively executed. Each of them calculates its activation value by taking the weighted sum of the outputs of the units in the preceding layer, and subtracting the threshold. The activation value is passed through the activation function to produce the output of the neuron. When the entire network has been executed, the output of the output layer acts as the output of the entire network (The Statistics Homepage, 2003).

The neural network is trained using one of the supervised learning algorithms. The supervised learning networks are the Multi Layer Perceptron (MLP), the Cascade Correlation learning architecture, and Radial Basis Function networks (Michie et. al. 1994). However, the most popular network architecture in use is the MLP and will be used for this study. It also uses the concept and the algorithm that we discussed in the previous part. The number of input and output units are defined

by the problem and one hidden layer with the number of hidden units set to half the sum of the number of input and output units (The Statistics Homepage, 2003).

The strength of this classifier is that it can deal with missing values and is also noise-tolerant. However, there is a limit for this tolerance. If there are outliers far outside the range of the normal values for the variables, they may bias the training. The disadvantage of this classifier is that it does not perform well with nominal attributes. Moreover, the time taken to construct the hidden layer can be very lengthy with the increase of the number of samples.

CHAPTER 3

MICROARRAY AND ISSUES ON MICROARRAY

3.1 Genes and Its Significance

Genes are responsible for establishing some set of properties in all living organisms which are made up of smallest units, cells. All essential functions of organisms are controlled by cells which carry very useful information to be passed from generation to generation of a living organism. As genes are somehow known as the discrete hereditary units which do this job and are made of deoxyribonucleic acid or better known as DNA (Dale and Schantz, 2007), the interaction among genes results in the appearance of different characteristics in organisms. The inherited information could be of characteristics such as the physical appearance of an organism which could be the colour of the hair, eye and other physical characteristics. Nevertheless, not all genes play the roles of the hereditary of these physical traits. Dominant genes for example will take over recessive genes. This means that the physical appearance is a result of the dominating genes taking place over the recessive ones.

Many genes are situated along each long DNA molecule. Chromosomes are attributes of DNA which sub-divide the activities of genes into the coding and non-coding sequences. The complete set of genes in an organism is called genome. Gene expression is obtained when the process of transmission of DNA to protein takes place. Gene expressions are important as they reveal the inheritable information in an organism.

Genomics is the study of structure and function of genes and have become a popular study over the years. The field of medicine for example has benefited from this branch of study. Genetic diseases which are caused by genetic disorders are a result of genes not properly expressed or not expressed at all. Cancer happens because these complications occur. Previously, studies to deal with genetic diseases are done by carrying out analysis of one gene at a time. This however is very time consuming and might not always be 100% accurate. Nevertheless, with the availability of microarray which enables a set of thousands of genes to be looked at simultaneously, the study of genes has become easier.

The expression values obtained through the transcription of genes can carry very significant meanings especially when it comes to the studies of genes in genetic diseases. In microarray experiments, scientists aimed to look at genes which are differentially expressed because these are said to be genes that are contributing to the occurrence of the diseases. These genes are also called the significant genes. A cancer tumor growth for example, might be due to a mutated gene or genes that are inappropriately expressed.

3.2 Microarray Technology

The advancement of microarray technology has created never-ending efforts in mining the data. Mining these data include performing regression, classification and even gene-analysis of the genes expression levels.

A technology that is widely adapted by most biologists to perform genomic analysis, microarray is a study on the interaction among a large number of genes and how a cell's regulatory network control vast batteries of genes simultaneously. Basically, microarray is so-named because the sizes of the sample spots obtained in its experiments are typically less than 200 microns (10^{-6}) in diameter. These spots are rather recognized as microscopic. Besides that, these arrays usually contain thousands of spots (Samb, 2005).

There are several different types of microarray namely Short oligonucleotide arrays (made by Affymetrix), cDNA or spotted arrays (originated by Pat Brown lab at Stanford), Long oligonucleotide arrays (Agilent Inkjet) and Fiber-optic arrays. However, the first and the second type are the most common ones. In addition, different types of microarray use different technologies for measuring the Ribonucleic acid (RNA) expression levels. As of 2002, the Affymetrix U133 2-chip set, can measure expression of over 30,000 genes which is almost the entire human genome (Piatetsky-Shapiro, 2003).

The uses of microarray in the field of medicine include DNA microarray, tissue microarray, protein microarray, plant microarray etc. Microarray may be used to assay gene expression within a single sample or to compare gene expression in two different cell types or tissues samples, such as in healthy and diseased tissue (Samb, 2005). Microarray also has many potential applications. These applications include:

1. More accurate disease diagnosis from gene expression levels;

2. Predicting treatment outcome;
3. Tailoring drug therapy based on gene expression levels (pharmacogenomics);
4. Drug discovery and toxicology studies;
5. Assisting fundamental biological discovery.

In the study of genetic diseases for example, microarray technology has been proven to be of help. Microarray enables scientists to look at numerous genes at a time which saves time and cost. Genes which are differentially expressed could be found out easily instead of the usual analysis of gene by gene. Thus, drugs could be made to directly aim at treating these diseased cells.

Microarray data usually has many variables (genes) and few samples, making the process of correctly analyzing such data difficult to formulate and prone to common mistakes. For this reason, it contributes to the rise of microarray data mining.

3.2.1 Process of DNA Microarray

Living cells contain chromosomes while deoxyribonucleic acid or better known as DNA contains thousands of genes. Each of those genes has specific composition and structure of the single protein. Every cell has the same sets of chromosomes, but they have very distinct properties. This is due to the differences caused by the abundance, state and the distribution of the cells. The changes of the protein is determined by the changes in the level of messenger ribonucleic acid (mRNAs), which are the nucleid acid polymers carrying information from chromosomes to the cellular machines that

synthesize new protein. Thus, gene expression is the process of transcribing the gene's DNA sequence into mRNA (Aas, 2001).

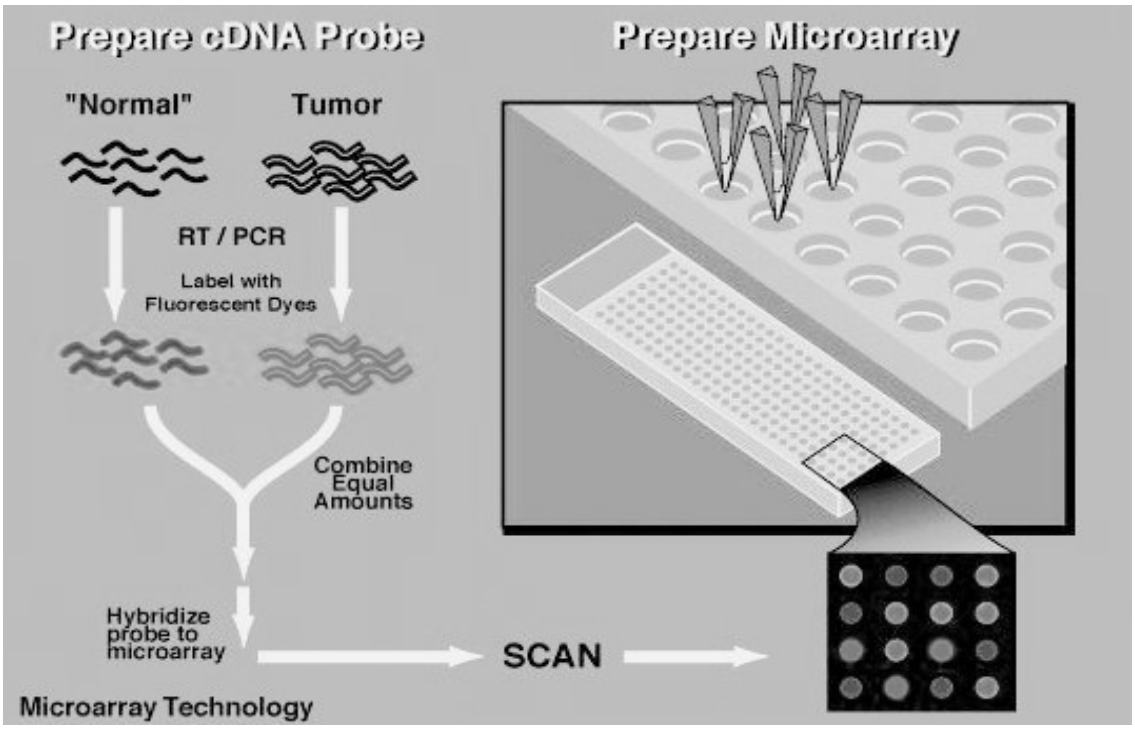

Figure 3.1: Process of Microarray (Aas, 2001)

A DNA microarray experiment consists of the measurement of the relative representation of each mRNA in a set of biological samples. This is done using principles of base-pairing or hybridization (Schena et. al., 1995). The collection of DNA spots is usually done on a solid surface, such as glass, plastic or silicon chip to form an array (Piatetsky-Shapiro, 2003).

In detail, the DNA samples are fixed to a solid surface with their position known in the array. Red and green dyes are used to label target or tumor sample and reference or normal samples as shown in Figure 3.1. The ratio (green/red) which is also the

intensity of mRNA are measured using a fluorescent microscope. The result is the ratio of the relative abundance of genes in the experimental sample and the common reference sample as the experimental sample is compared to the common reference sample. Therefore, the ratio obtained can consist of positive and negative values. The positive values indicate a higher expression in the target versus the reference and vice versa for the negative values (Aas, 2001).

Result obtained is in the form of a table, whereby the rows represent the genes, and the columns represent the samples. Each cell is then changed to the log of base 2 – transformed expression ratio of the appropriate gene in the appropriate sample (Aas, 2001).

3.3 Review of Microarray Studies

The development of microarray has attracted many researchers to conduct studies in the hope to obtain a reliable research result that can contribute to the significance of microarray development. Studies comprise from analysis and interpretation of microarray data to clustering and classifying it varying from the usage of statistical methods to a combination of mathematical modeling and biological techniques to using machine learning artificial intelligence tools.

One very vital study on the microarray data is on the feature selection method. Many researchers had created different methods for performing feature selection, which is to pre-select important genes before conducting any classification. Feature selection is

indeed an important step as it eliminates thousands of other genes which are somehow not contributing much to the data analysis. Nevertheless, significant genes have different definition among researchers, and because of this belief, there are various ways in deciding which genes are important and which are not. Microarray dataset is a highly-dimensioned data and often contains a lot of noisy genes and outliers too. Thus, researchers are keen to come out with the most reliable gene selection method. Li et. al. (2005) had combined the powerful genetic algorithm tool together with support vector machine to extract optimal subset of gene. Xiao et. al. (2004) suggested that one needs a pertinent distance between two random vectors in order to compare gene expression signals in two different experimental conditions. Kim and Park (2004) used regularized t-tests to improve the identification of differentially expressed genes in microarray data. Jirapech-Umpai and Aitken (2005) have used genetic algorithm to pre-select their best genes and then classify them using nearest neighbour classification scheme. In Datta and DePadilla's (2006) study, they demonstrated the usefulness of a feature selection step prior to applying a machine learning tool. A natural and common choice of a feature selection tool is the collection of marginal p-values obtained from t-tests for testing the intensity differences at each m/z ratio in the cancer versus non-cancer samples. Silva el. al. (2005) investigated how to use feature selection techniques to speed up the process of finding significant genes. Ng and Breiman (2005) proposed a bivariate method using Random Forest to select significant genes. In their method, they select pairs of genes which are relevant to one another. While others use just one single algorithm to carry out the feature selection, our stair-line method involves a few steps carried out using Mathematica as well as Random Forest.

While selecting significant genes is essential, it is also important to have reliable computational tool to carry out the classification process. Khan et. al. (2001) used artificial neural network to classify four classes of round blue-cell tumor. They combined genetic algorithm and support vector machine to classify multi-class tumor. Another paper by Yeung et. al. (2005) had created a Bayesian model averaging tool to pre-select genes and classify them for multi-class microarray data. Other studies include hybridization of two or more machine learning tools for classification purposes as what Penga et. al. (2003) had done.

Some researchers even swerve away from either the invention of new classification scheme and feature selection method to just simply conducting a robust comparison of few classification algorithms and feature selection to select the ones that perform best. Wu et. al. (2003) did a comparison of the statistical methods just for the classification of ovarian cancer using mass spectrometry. Another paper by Li et. al. (2004) which conducted a comparative study on feature selection and multiclass classification methods for tissue classification based on gene expression. In their paper, they used eight feature selection methods, while training nine datasets on seven classification algorithms. A recent paper done by Lee et. al. (2005) is even more intensive as they compared among 21 different classification methods in seven microarray datasets after undergoing three different gene selection methods or pre-processing stage.

While selecting important genes has always been considered an important step, there are not many studies in which analysis of the importance of genes that have been selected are carried out. One paper by Pang et. al. (2006) conducted a pathway analysis using the Random Forest classification and regression to better understand significant genes in their biological manner. In their paper, they described a pathway-based method to rank important genes and to distinguish outliers. However, their method is based more on the biological approach rather than the mathematical approach.

Other papers include focusing on using Random Forest as a strong tool for machine learning. Chen et. al. (2004) came up with two ways to deal with imbalanced data which is often a common problem faced by all researchers when mining microarray data. They have used the approach of cost sensitive learning and the other is based on a sampling technique. Nevertheless they have proven that while the error rate of a particular minority class can be reduced, the overall error rate is still considered more essential as it is what researchers often look at.

Issues on microarray are just too many to tackle. This has led to never-ending researches on problems which often revolved around obtaining more accurate classification results and acquiring significant genes or genes that are differentially expressed. While many chose to focus only on one section of the data mining parameter, we have chosen to further prove the strength of Random Forest as a classifier with the help of simple filtering and statistical measuring techniques. We have borrowed the idea of Teschendorff et. al. (2006) of looking at one of the statistical measurement known as kurtosis to rank the genes. In their paper, they have used the statistical measurement to

rank the genes and cluster them. However, Random Forest was not used in their paper as their main concern was to do clustering.

CHAPTER 4

THE STAIR-LINE METHOD

The experiments run for this study involved three basic stages which are pre-processing, processing and post-processing. The pre-processing stage, also known as the data-cleaning task, is one of the major stages in this research. It comprises of four steps of eliminating noisy genes and selecting important genes. The processing stage in this study involves the building of classification model with our proposed method of reducing the effect of kurtosis by changing the error function in Random Forest whereas, post-processing is the stage where we will view the results and comment on them. The post-processing stage will be discussed further in chapter 5.

4.1 Pre-processing Data

Microarray data always comes in as a large dimensioned matrix. Hence, it is inefficient to just process the raw data without any data-cleaning. In fact, data-cleaning is an essential step as it prepares the data well enough for processing tasks. It is also almost impossible to process a data without first cleaning it. What adds on to the problem is that data preparation is a very tedious task as it involves more than one step and one method. Therefore, data preparation has to be carried out carefully considering relevant factors as closely as possible.

In this part, we will explain the data preparation methods we use to prepare our data. In short, data preparation involves thresholding and filtering. Thus, we have

written several Mathematica scripts (see Appendix B-E) to complete the above tasks. All five datasets are prepared in similar manner.

4.1.1 Removing Irrelevant Information

Original data have attributes in rows and samples in columns. Besides, it also contains other unimportant information such as control genes and genes description as well. Thus, the first step in data preparation is to remove any control features. As for an experiment that is obtained using Affymetrix gene chip, it is often called the Affymetrix control or in short, Affy-control. Next, we also remove the gene description as it is not needed in this study.

4.1.2 Threshold and Filter

Note that because the expression values can vary very drastically while some expression values might not be well expressed due to weak signal strengths. Therefore, thresholding is essential. A standard minimum value of 20 and maximum value of 16000 have been introduced and biologists consider any values out of this range to be unreliable (Piatetsky-Shapiro et. al., 2003).

Filtering is done by calculating the fold difference values of the genes. Fold difference is the maximum value across samples divided by minimum value. Fold difference is important as it is frequently used by biologists to assess the changes of genes. Filtering is done by omitting genes with values of fold difference less than 2.

Carrying out this filtering process is vital as some genes are not well expressed and do not vary sufficiently to be useful (Samb, 2005).

4.1.3 Finding Significant Genes

DNA microarray data typically has many attributes (genes) and few examples (samples), making the process of correctly analyzing such data difficult to formulate and prone to common mistakes. Thus having the data threshold and filtered are not enough if we want to come out with good classifications (Samb, 2005). To deal with this problem, we look for genes that can distinguish well among themselves between classes and eliminate any unclear feature. In other words, we reduce the number of genes by performing several steps in finding most significant genes. Finding significant genes means selecting genes that vary distinctively from one class to another or are differentially expressed. On top of that, it is vital to only collect genes that are not noisy and genes that are distinctively different, that is to say genes that show clear boundary among classes. Besides, with reduced number of genes versus samples, more precise classification can also be acquired.

To do this, we propose a stair-line method which basically involves three steps. The stair-line method starts off by first selecting the top 50 genes as selected by Random Forest. Kurtosis values are then calculated for these 50 genes. Initial analysis found out that these genes have a negative kurtosis value of -2 or very close to -2. Thus, we narrow down our study scope to only look at genes with kurtosis value of -2. Hence, we have

eliminated genes which have kurtosis value not equal to -2 and these genes can also be considered as the outlier genes.

The remaining genes are now filtered and the top 20 genes with the highest t-values are selected and arranged in ascending order. From the mathematical point of view, a t-value is a statistical measurement that represents the distance or deviation between two classes in units of standard deviation. It shows how genes in different classes are differentially expressed. Therefore, a high t-value shows significant difference between genes in different classes. We have also chosen to use a variation of the original t-value formula which is the t-LIMMA formula. T-LIMMA formula is the moderated t-statistic and is shown to follow a t-distribution with augmented degrees of freedom (Smyth, 2004).

Formulae used are as follows.

$$\mu = \frac{\sum x}{N}$$

$$\sigma_1 = \sqrt{\frac{N_1 * \sum x_1^{\,2} - \left(\sum x_1^{\,2}\right)^2}{N_1 * (N_1 - 1)}}$$

$$T - Value1 = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^{\,2}}{N_1} - \frac{\sigma_2^{\,2}}{N_2}}}$$

$$Kurtosis = \frac{\sum (x - \mu)^4}{N\sigma^4} - 3$$

To ease the understanding of the formulas, we shall take the brain tumor dataset as an example to explain the symbols used.

The classes available in the brain tumor dataset are MED, EPD, MGL, RHB and JPA.

x is the expression value of a gene

N_1 is the number of samples in class MED

N_2 is the number of samples in the remaining class

μ_1 is the average expression values for each gene in all samples with class MED

μ_2 is the average expression values for each gene in all samples with remaining class

σ_1 is the standard deviation of the expression values for each gene in all samples with class MED.

σ_2 is the standard deviation of the expression values for each gene in all samples with remaining class (EPD, MGL, RHB and JPA)

T-Value1 is the t-value for each gene in all samples with class MED

4.2 Processing Data

There are numerous classification tools available that are well-established. However, the main tool used in this work is Random Forest which is initially implemented in Fortran programming language.

While Random Forest computes a forest like classification, it means that models built are based on the creation of more than one tree. Therefore, the number of trees opted for is one very essential parameter. To see how this parameter actually affects the classification results, we have run the classifier using 100, 1000, 5000 and 10000 trees.

In the processing stage, genes that are selected from section 4.1 are now being classified using the modified Random Forest classifier. As our main objective is to deal with the abnormality of distribution of the genes which have a kurtosis value of non-zero, we have created an error function to compensate this non-normal distribution. An approximate standard error which is used to handle kurtosis is $e = \sqrt{\frac{24}{n}}$ whereby the n represents the number of samples observed (Crawley, 2005). In our method, we have proposed to modify the error function in Random Forest into $e = \sqrt{\frac{(node \times trees) - (average)^2}{n}}$, whereby node is the value of node on the built tree, trees are the number of trees built, average is the average of all nodes on the built tree ion a chosen sample and *n* represents the number of samples observed.

Besides Random Forest, we also chose to use WEKA, Waikato Environment for Knowledge Analysis which is an open source software by the University of Waikato. WEKA has a wide selection of other classifiers and evaluation methods which give the prime reason for its selection of our data mining tool. The classifiers that will be used in WEKA are ZeroR, J48, Naïve Bayes (NB), k-nearest neighbour (KNN), support vector machine (SMO) and neural network (MLP). Figure 4.1 shows the flow of experiments done in this study from pre-processing to post-processing and the shape of a stair-line can be seen in this figure.

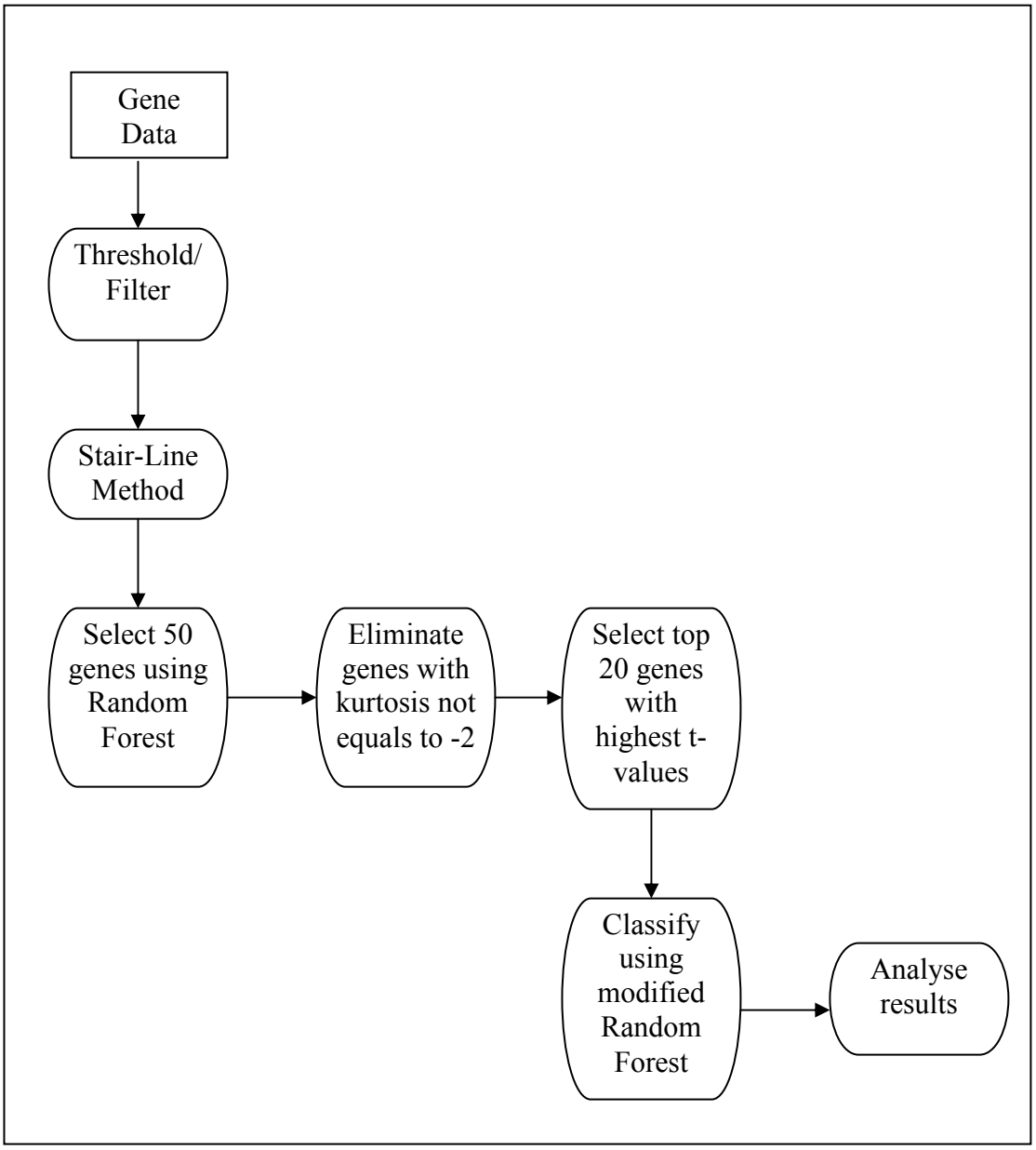

Figure 4.1: Flow of Experiment

4.3 Datasets and Their Descriptions

There will altogether be five datasets which will be used in this study. These five datasets have their own uniqueness of number of genes as well as number of samples and class distributions. To demonstrate the superiority of our stair-line method, we have chosen to look at both binary and multi-class datasets. Besides, we have also widen our horizon by looking at datasets with as low as 57 samples to as high as 203 samples. Table 4.1 shows the full descriptions of the datasets used in this study.

Table 4.1: Descriptions of Datasets Used

Dataset	Attributes / Number of Genes	Number of Classes	Number of Samples	Class Distribution
Brain Tumor (BRAIN) (Pomeroy et. al., 2002)	7070	5	69	MED - 39 samples EPD - 10 samples MGL - 7 samples RHB - 7 samples JPA - 6 samples
Central Nervous System (CNS) (Pomeroy et al., 2002)	7070	2	60	Survivors – 21 samples Failures – 39 samples
Diffuse Large B-Cell Lymphoma (DLCBL) (Shipp et al., 2002)	6817	2	77	DLBCL – 58 samples FL – 19 samples
Leukaemia (LEU) (Armstrong et al., 2002)	12584	3	57	ALL - 20 samples MLL - 17 samples AML - 20 samples
Lung Cancer (LUNG) (Bhattacharjee et al., 2001)	12600	5	203	ADEN - 139 samples SQUA - 21 samples COID - 20 samples SCLC - 6 samples NORMAL-17samples

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Selecting Optimum Number of Trees

While selecting significant genes to improve classification results is the main objective in this study, we also need to first decide on the number of trees to grow in our forest. The number of trees used in our trials is limited from 100 trees to 10000 trees. We do not want to grow more than 10000 as we want to avoid over-fitting problems (Díaz-Uriarte and Alvarez de Andrés, 2006).

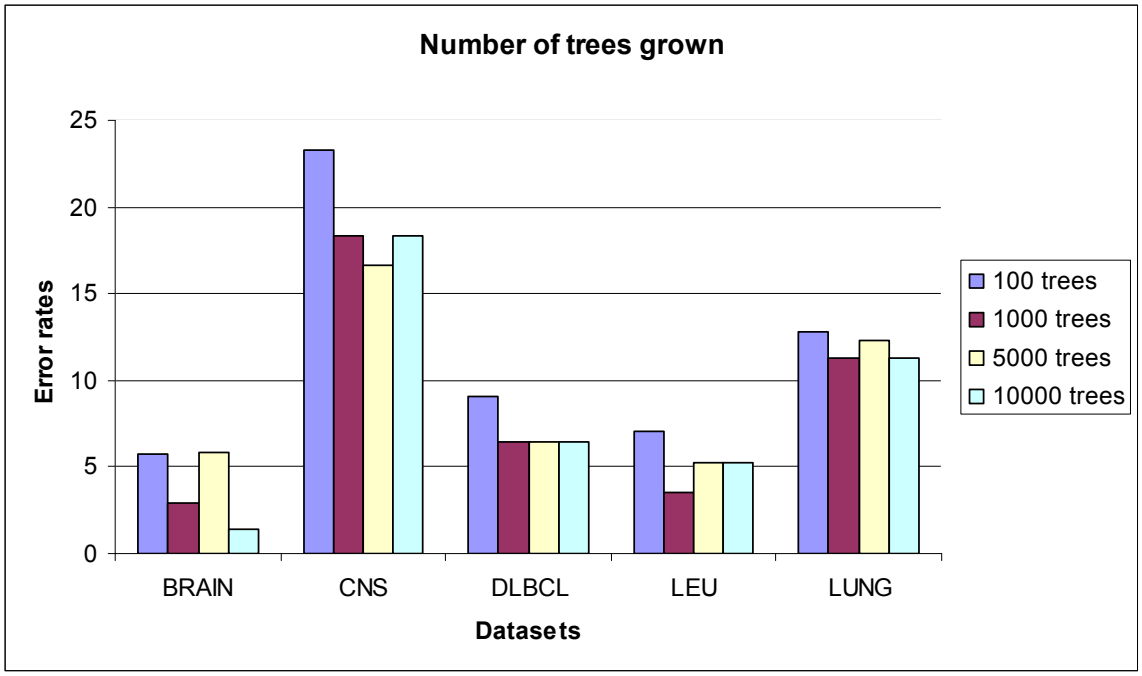

Figure 5.1: Number of Trees Grown for Each Dataset

Figure 5.1 shows that the error rates for each dataset decreased as the number of trees built increases. 10000 trees can be seen works best for all datasets except for the

LEU dataset. This could be due to the vast difference between the number of genes (12533) and number of samples (57) in LEU dataset. 100 trees on the other hand give very poor results. It is therefore obvious that computing 100 trees is too few to produce good results.

Although 1000 trees also give just as good results as compared to 10000 trees for all the datasets, we have chosen to use 10000 trees instead. This is because we want to accommodate the vast number of genes which exceeds 6000 genes in all datasets used in this study. Recall how the bootstrap sample is taken in the Random Forest algorithm, growing 1000 might give us good results but as there are more than five times of genes as compared to the number of trees built, some genes might not be involved in the computation of trees and thus results given might not be accurate although the error rate is low. Besides that, we also want to keep uniformity in the experiment and have therefore chosen to grow 10000 trees for all datasets.

5.2 Results of Threshold and Filtering

As mentioned in chapter 4, threshold and filtering are two very important steps in data pre-processing as it helps us to identify genes that are differentially expressed or really useful in the classification. This is because some genes that are collected might not be well expressed or are expressed only in very few samples. Threshold is essential as we want to standardize the genes expression values and thus have kept to using 20 for minimum and 16000 for maximum. Our main reason in doing so is also to alleviate any noise level. Table 5.1 shows the percentage of genes reduction from the original dataset.

These filtered datasets showed a clearer picture on how the genes are differentially expressed. When genes are differentially expressed, it will be easier for us to discover which are the noisy genes are and which are not.

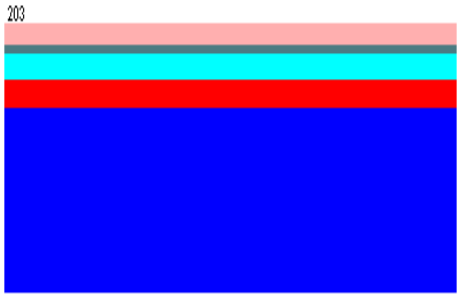

Figure 5.2: Unfiltered Data

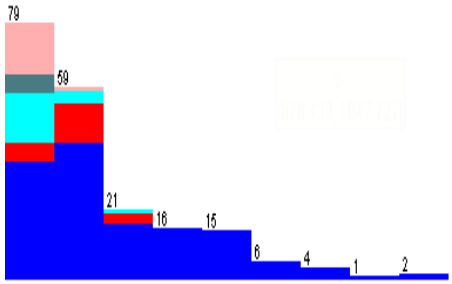

Figure 5.3: Filtered Data

Figure 5.2 shows unfiltered lung data with 203 samples and 12600 genes and we can see clearly that there is no variation among the genes for all the five classes. In other words, the genes in the unfiltered data do not show us which genes are differentially expressed and which are not. Hence, it is not likely to select good predictive genes. On the other hand, Figure 5.3 shows a good variation with 203 samples and now 12232 genes distributed among the five classes. The filtered data is also definitely going to make a better model (Twyman, 2002).

Table 5.1: Percentage of Genes Reduction After Threshold and Filtering

Dataset	Number of genes in original dataset	Number of genes left	Percentage of reduction in the number of genes (%)
BRAIN	7070	6413	9.3
CNS	7070	6921	2.1
DLBCL	6817	6679	2.0
LEU	12584	11842	5.9
LUNG	12600	12232	2.9

From table 5.1, we can see that there are a total of 657 genes which are not well expressed and thus have been eliminated in the BRAIN dataset. While for the CNS dataset, there is a total of 149 genes which have been eliminated whereas, 138, 742 and 368 genes are eliminated for the DLBCL, LEU and LUNG datasets respectively.

Of the five datasets, the BRAIN dataset has the highest percentage of genes reduction. This means that the BRAIN dataset has a lot of genes which are not well expressed and do not vary well from each other and are thus considered not able to give good classification model. Besides, the reason for this dataset to be having such high number of not well expressed genes could also be due to human error when collecting the data. Human error is often something that we cannot avoid in real life experiments.

5.3 Results of Stair-Line Method

5.3.1 Selecting Top 50 Genes Using Random Forest

50 top genes were first selected from the threshold and filtered set of genes for all datasets. Table 5.2 shows the results of percentage of correct classifications obtained as compared to the classification done on the original set of genes.

Table 5.2: Results for Top 50 Genes Obtained from Random Forest

Dataset	Percentage of correct classification (%)
BRAIN	89.9
CNS	81.7
DLBCL	92.2
LEU	94.7
LUNG	82.3

5.3.2 Results For Top 20 Genes Selected From Highest T-value After Eliminating Genes With Odd Kurtosis Value

Kurtosis values are calculated for the 50 top genes selected in section 5.3.1. Initial analysis found out that these genes are found to have a negative kurtosis value of -2 or close to -2. Those genes with negative kurtosis value are also considered to have a platykurtic distribution. Nevertheless, while most genes have negative kurtosis values, there are some genes with very high negative values and some even with positive kurtosis values. Thus, we considered these genes as the outlier genes and have eliminated these outliers.

The t-values for those genes are then calculated according to their classes and are ranked in ascending order whereby the top 20 genes for each class are selected. Overlapping genes are taken into account as only one gene. Table 5.3 shows the number of genes left for each dataset after selecting top 20 genes from each of the classes according to their highest t-values.

Table 5.3: Number of Genes Left After Being Ranked According to Highest T-Values

Dataset	Number of genes left
BRAIN	43
CNS	24
DLBCL	28
LEU	41
LUNG	48

While many other researchers chose to look at the algorithm per se to select important genes, we analyze the distribution of the genes in our dataset using the statistical measurement, kurtosis in our research.

Chen et. al. (2004) described the imbalance of microarray data by looking at the contributing classes. In their paper, they stated two methods to reduce the effect imbalance caused by one dominating class. The methods are cost sensitive learning and sampling technique. As the number of samples for a particular class can never be changed, they introduced the two methods to reduce the influence of the dominating class in classification. While the imbalance of data is looked from the class point of view, the former researcher has not looked at the distribution of the genes per se unlike that of our research.

Recall that the main goal in this study is to reduce the effect of kurtosis found among genes. Our proposed method which uses the stair-line method to select important genes and finally running it through a modified error function in our Random Forest algorithm has shown improvement in the percentage of correct classification.

Kurtosis which measures the peakedness of a distribution has effect on the classification results. Our experiment showed that the datasets that we mined are platykurtic. On the other hand, they have a negative kurtosis value or have a flat top distribution. A platykurtic distribution also means that the distribution is not normal which could be due to variances which are due to infrequent deviations (Kline, 2008). Kurtosis is not easily seen with bare eyes and can only usually be distinguished by

calculations. Therefore, we have used a formula to calculate the kurtosis level of our genes. Any values of kurtosis that are non zero are considered non-normal distribution and therefore have their own level of kurtosis depending on the value obtained from the formula.

For example, Figure 5.4 and Figure 5.5 below show gene M31303_rna1_at extracted from the BRAIN dataset, having a negative kurtosis value. The five different colours shown in the graph represent five different classes as are in the BRAIN dataset.

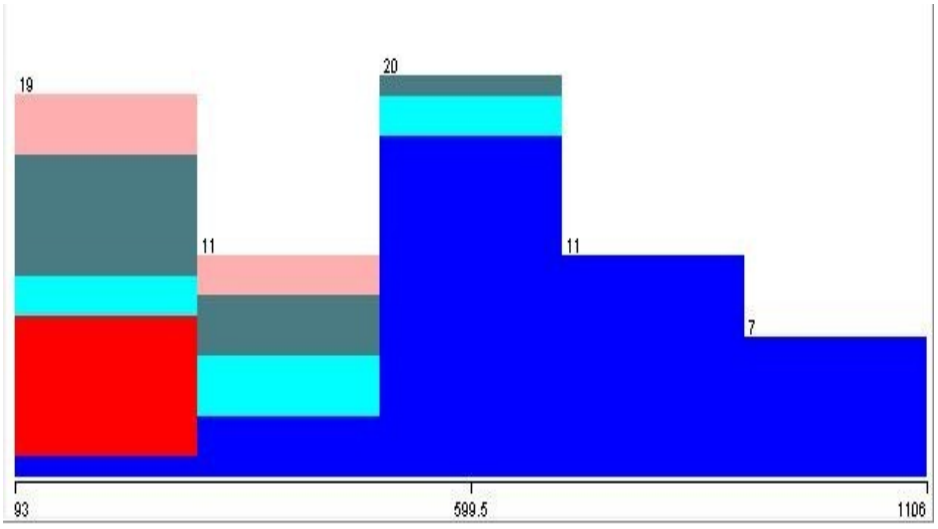

Figure 5.4: Graph of Gene M31303_rna1_at

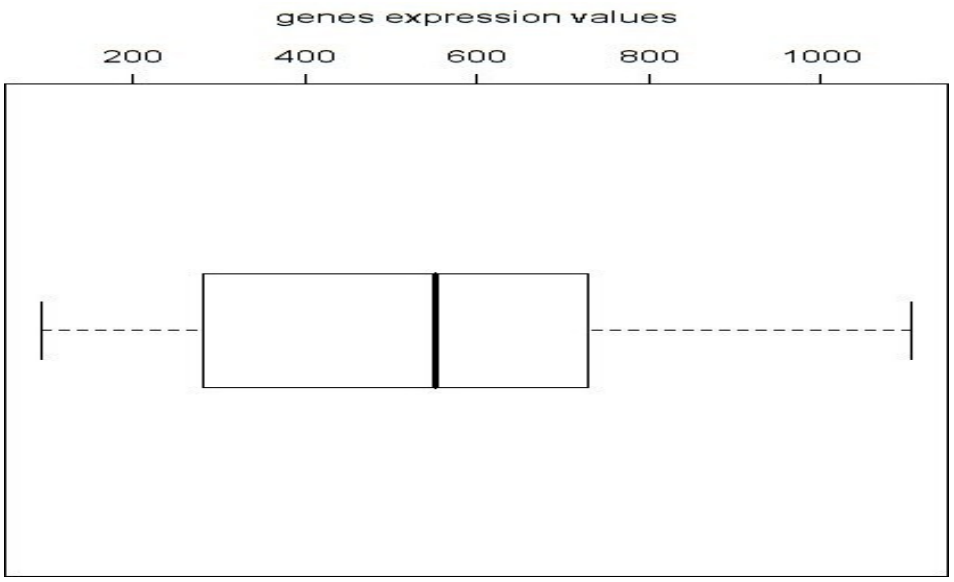

Figure 5.5: Box Plot for Gene M31303_rna1_at

5.3.3 Comparison Results With Other Classifier

Table 5.4 shows the results of classifications obtained from using the modified Random Forest as compared to the other classifiers. A graph representation of the results is also shown in Figure 5.6.

Table 5.4: Percentage of Correct Classification among Classifiers

	RF	J48	KNN4	ZeroR	SMO	MLP	NB
BRAIN	97.1	89.9	94.2	56.5	95.7	94.2	95.7
CNS	84.7	71.2	76.3	64.4	88.1	76.3	84.3
DLBCL	94.8	79.2	94.8	75.3	93.5	92.2	90.9
LEU	98.2	89.5	93.0	35.1	94.7	96.5	94.7
LUNG	89.7	90.6	87.2	68.5	89.7	89.2	89.2

Legend for Table 5.4 and Figure 5.6:

RF: Random Forest

J48: Decision trees

KNN4: 4th Nearest Neighbour

ZeroR: Zero R

SMO: Support Vector Machine

MLP: Neural Network

NB: Naïve Bayes

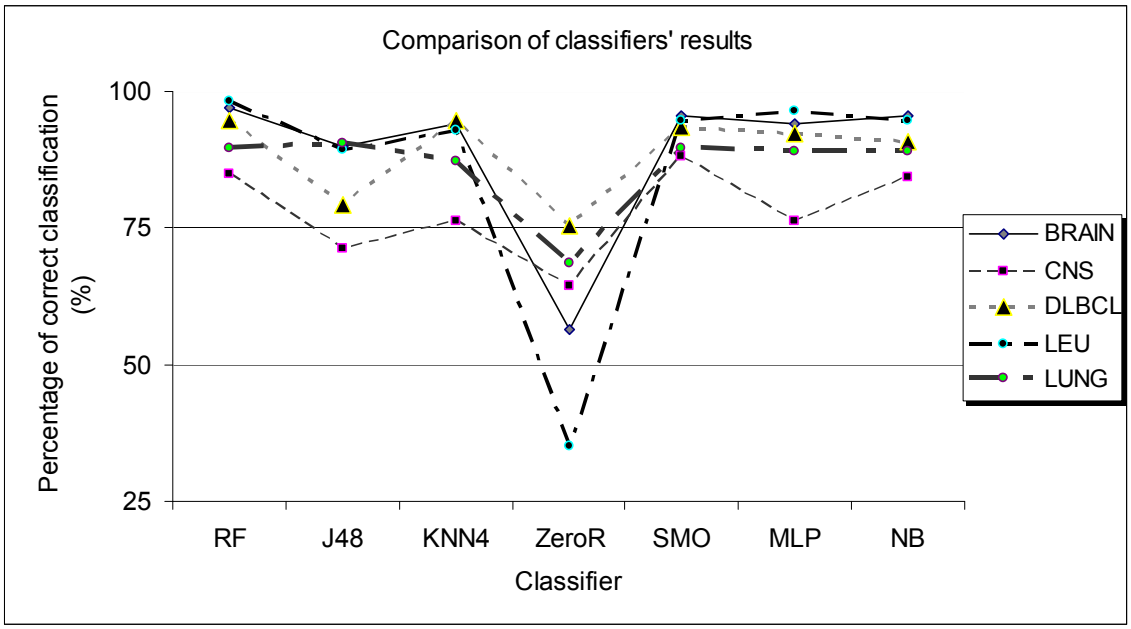

Figure 5.6: Comparison of Classifiers' Results

Figure 5.6 shows the resulst of the stair-line method in reducing the effect of kurtosis. As can be seen in the Figure 5.6, Random Forest (RF) has shown a consistently competitive result as compared to the other classifiers. The final top 20 genes that are chosen from the modified Random Forest algorithm has seen giving better results than before.

While the results obtained from ZeroR classifier is worst among all, it is also one classifier which does not use any function and simply predicts the majority class in the

data (Witten and Frank, 2000). This classifier actually serves well as a baseline comparison for the other classification methods.

As can be seen, the J48 results are also not as promising. This explains the reason for using Random Forest which computes a collection of trees instead of just one tree as computed in the J48 algorithm. The ability of computing more trees in Random Forest clearly gives a higher percentage in correct classification. It is also shown that the J48 which computes only one single tree does not have the ability to perform well when given a larger data such as the microarray data which usually comes in huge dimension.

The KNN4 classifier here means the fourth nearest neighbour is used to classify the unknown object. The reason for this selection is done after Ng and Abu Hasan's (2007) study. In that paper, it is shown that KNN4 works best for the datasets as compared to the other neighbour values ranging from one to ten. The reason to keep it below ten is also to avoid similar classification problem. Recall that the nearest neighbour classifier uses the distance function and thus unknown object will take the class label that is nearest to it. Therefore, using a k that is too high will cause the classification to be all the same. In other words, the unknown object will take class-value of the class that occurs most times or the dominating class.

Apart from Random Forest, three other classifiers which performed rather well are SMO and MLP and NB. The SMO classifier used was done using the default polynomial kernel. The result was also proven to be of better performance in the

polynomial kernel compared to the radial kernel as shown in the paper by Ng and Abu Hasan (2007).

The MLP classifier that is done here used *a* number of hidden layers. Also presented by Ng and Abu Hasan (2007) is that the percentage of correct classification improves with the increment of the number of hidden layers. *a* is equal to ½ (number of genes + number of samples). Nevertheless, while the higher number of hidden layers improves the classification results, it also takes longer time to compute especially with the high number of samples such as 203 as in the LUNG dataset. However, we still chose to use *a* number of hidden layers in this research as our proposed stair-line method had reduced the original thousands of genes to 20. And thus, our computation time is still manageable and at the same time, we obtained optimum results for this classifier.

While the other classifiers chosen have had their parameters tuned to obtain optimum results, the Naïve Bayes classifier simply uses its unique posterior probability of classifying and does not require any fine tuning. Yet, results obtained are almost as good as the ones obtained by Random Forest.

5.3.4 Evaluation Method

While the parameters are tuned for the classifiers to obtain optimum results, it is also important to select the best evaluation method. Evaluation methods here refer to methods that are used to evaluate the performance of each of the classifiers. Ng and Abu Hasan (2008) had also determined that the usage of 9:1 cross validation or better known

as the 10-fold cross validation gives optimum results. This evaluation uses 90% of data to be trained leaving the other 10% to be tested and the procedure is repeated ten times until every sample has been used exactly once for testing.

5.4 Main Contribution

Looking at the distribution of genes is not something new yet not many researchers are working towards this context. In this research, we have highlighted the idea of stair-line method in selecting important and significant genes. The reason it is called stair-line is because it involves three steps and not just one unlike the other classification schemes. While good classification accuracy is subjected to different definitions, in this study, a high percentage of correct classification is considered as good classification accuracy.

Many studies also chose to classify imbalanced data by defining the dominant class as what has been done by Chen et. al. (2004). Instead, our proposed stair-line method defines imbalanced data by looking at the genes distribution. The method not only reduced the effect of kurtosis, but has been proven to integrate well with Random Forest classifier.

This method of looking at how genes are distributed shows how important it is to consider genes distribution before selecting important genes. Our method of reduction of the dimension of microarray datasets step by step, although tedious, has proven its results. Using Random Forest as a classifier and a tool to pre-select the genes while

looking at the genes distribution shows that genes selection does not necessarily need to involve one algorithm per se.

CHAPTER 6

CONCLUSION

Microarray is a field that is being studied widely and its research purpose has grown from time to time. The technology of microarray has eased the task of analyzing genes, as it enables tens of thousands of genes to be looked at simultaneously instead of the conventional way of which one gene is looked at a time. Measuring gene expression using microarray is relevant to many areas of biology and medicine. The uses of microarray in the field of medicine vary and it includes DNA microarray, tissue microarray, protein microarray, plant microarray and many more which add to the reasons why microarray data is mined so widely over the past few years.

Cancer, for instance, is one of the health diseases which has benefited from the existence of microarray technology. Over the past few years, classifying cancer using microarray technology has been widely researched and results are shown to be optimistic. Hence, microarray data mining which uses the combination of both mathematical modeling and biological technology is certainly a comprehensive way not only to classify disease but also to examine disease outcome and discover new cancer subtypes.

However, just like when mining other types of data, many challenges are faced when mining microarray data. First, microarray data is one data which contains the expression levels of tens of thousands of genes, thus increasing the difficulty level when

it comes to mining the data. Secondly, microarray data usually has a very large number of variables as compared to the observed samples. Thus, these challenges have led to data-cleaning which involved a few steps such as threshold, filtering and feature selection. All the mentioned steps play an important role in selecting only significant or useful genes to give competitive results.

Classification, which allows us to look for new patterns, is the main mining method in this research. Classification not only allows us to classify genes but also to see hidden patterns especially among significant genes in order to obtain better results. Most classification schemes rely very much on selecting the important genes or better known as genes which can contribute significantly to our classification results.

The very main concern in most research in classification of microarray data is to select important genes. Selecting important genes is important because it helps us reduce the dimension of the usually highly-dimensioned microarray data. Important genes are those with relevant information towards good analysis of the data. This is a step to be carried out as unclean microarray datasets contain too many genes that are noisy, outliers or irrelevant. There are however various ways of selecting significant genes. Available ones are the usual univariate and multivariate methods for selecting significant genes according to a certain criteria before performing any classification technique on the data. Nevertheless, dimension-reduction should not cause the loss of its original information as this can cause significant loss of information from our datasets.

In this research, we have proposed a stair-line method which involves three steps to determine important genes. We look at the statistical measurement known as kurtosis for sets of genes initially selected using Random Forest. Random Forest which grows a collection of trees has given us the option to pre-select the number of trees we choose to grow instead of just one in the conventional tree algorithm. Our initial test with the raw datasets shows that growing 10000 trees give best results for all datasets. Besides, the error function in the original Random Forest classifier is modified to reduce the effect of kurtosis. The results obtained show the effectiveness of our method as it successfully gave a better classification accuracy in almost all datasets used.

We have also proven, in our method, that it is not impossible to manage the vast dimensions of microarray which contain thousands of genes by hand computations as these computations can be transcribed into computer algebraic system such as Mathematica scripts.

We have stressed the importance of good classification accuracy. Good classification accuracies are important for building a good model which can then be used as a prediction model. As mentioned, good classification accuracy also depends on the criteria that are being looked at when selecting significant genes. The main criterion that is being looked at in this study is to reduce the effect of kurtosis on genes. As such, the mining method in this research meets the objectives of our study. We are confident that this research will shed a different light in feature selection as well as classification. Previously, very few researchers look at the distributions of genes but rather their

dominant class or values per se. Our research has pointed out the importance of considering the genes distribution while selecting significant genes.

Nevertheless, we shall not deny that every proposed method has not only strengths but also flaws. In most cases in the real world, obtaining perfect classification accuracies are almost impossible. And even if there are perfect results, doubts are there as to whether the researches were carried out properly. As these researches are going to be the pioneers of the field of medicine in cancer curing, doctors themselves have to validate the results obtained from all the researchers' works. This might give rise to another problem such as the definition of good classification accuracies. As long as there are different definitions of good classification accuracies, it is going to take some time before everyone can see eye-to-eye as to which definition works best and finally apply it into the field of medicine in real life.

For future research purpose, more cases of microarray data can be applied to test the effectiveness of our method. As our research has also proven that it is essential to consider genes' distributions, the kurtosis, future research work can involve the prediction of cancer classes on test sets with unknown class with the same type of distribution as well. Such predictions can also eventually link to a brighter future of cancer treatment. Besides that, our stair-line method can serve as a strategy to researchers who choose to deal with feature reduction in microarray data mining as it involves not just one simple step but a few steps. Moreover, this stair-line idea can also be used even if the researcher choose not to use Random Forest as their main classifier.

REFERENCES

Aas, K. (2001). *Microarray Data Mining: A Survey*. Norsk Regnesentral SAMBA/02/01.

Armstrong, S. A., Staunton, J. E., Silverman, L. B., Pieters, R., Boer, M. L. D., Minden, M. D., Sallan, S. E., Lander, E. S., Golub, T. R. and Korsmeyer, S. J. (2002). *MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia*. Nature Genetics **30**, 41-47.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E. J., Lander, E. S., Wong, W., Johnson, B. E., Golub, T. R., Sugarbaker, D. J. and Meyerson, M. (2001). *Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses*. PNAS **98**, 13790-13795.

Breiman, L. (2001). *Random forests*. Machine Learning **45**, 5-32.

Chen, C., Liaw, A., and Breiman, L. (2004). *Using Random Forest to Learn Imbalanced Data*. Technical Report 666, Department of Statistics, University of California, Berkeley. Available access: http://www.stat.berkeley.edu/tech-reports/666.pdf

Crawley, M. J. (2005). *Statistics: An Introduction Using R*. John Wiley and Sons. Available access: http://books.google.com.my/books?id=czbzO5iD1Z0C

Dale, J. W. and Schantz, M. (2007). *From Genes to Genomes: Concepts and Applications of DNA Technology*. Wiley-Interscience. Available access: http://books.google.com.my/books?id=LDcEOpOj6MsC

Datta, S. and DePadilla, L. M. (2006). *Feature selection and machine learning with mass spectrometry data for distinguishing cancer and non-cancer samples*. Statistical Methodology **3***,* 79-92.

Díaz-Uriarte, R. and Alvarez de Andrés, S. (2006). *Gene selection and classification of microarray data using random forest*. BMC Bioinformatics **7(3)**.

Efron, B. and Tibshirani, R. (1997). *Improvements on cross-validation: the .632+ bootstrap method*. J American Statistical Association **92,** 548-560.

Gamberger, D., Marić, I. and Šmuc, T. (2001). *DMS Tutorial, Decision Trees*. Rudjer Boskovic Institute. Available access: http://dms.irb.hr/tutorial/tut_dtrees.php

Giudici, P. (2003). *Applied Data Mining, Statistical Methods for Business and Industry*. Wiley, West Sussex, England.

Jirapech-Umpai, T. and Aitken, S. (2005). *Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes*. BMC Bioinformatics **6(148)**.

Khan, J., Wei, J. S., Ringnér, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C. and Meltzer, P. S. (2001). *Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks*. National Institute of Health (NIH) **7(6),** 673-679.

Kim, R. D. and Park, P. J. (2004). *Improving identification of differentially expressed genes in microarray studies using information from public databases*. Genome Biology **5(9),** 70.1-70.10.

Kline, R. B. (2008). *Becoming a Behavioral Science Researcher: A Guide to Producing Research That Matters*. Guilford Press 237-240. Available access: http://books.google.com.my/books?id=ppJg1QIncRYC

Lee, J. W., Lee, J. B., Park, M. and Song, S. H. (2005). *An extensive comparison of recent classification tools applied to microarray data*. Computational Statistics & Data Analysis **48(4)**, 869-885.

Lee, Y. and Lee, C. K. (2003). *Classification of multiple cancer types by multicategory support vector machines using gene expression data*. Bioinformatics **19,** 1132-1139.

Li, L., Jiang, W., Li, X., Moser, K. L., Guo, Z., Du, L., Wang, Q., Topol, E. J., Wang, Q., Rao, S. (2005). *A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset*. Genomics **85,** 16-23.

Li, T., Zhang, C. and Ogihara, M. (2004). *A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression*. Bioinformatics **20(15)**, 2429-2437.

Li, Y., Campbell, C. and Tipping, M. (2002). *Bayesian automatic relevance determination algorithms for classifying gene expression data*. Bioinformatics **18(10)**, 1332-1339.

Michie, D., Spiegelhalter, D.J. and Taylor, C.C. (1994). *Machine Learning, Neural and Statistical Classification*. Available access: http://www.sicyon.com/classifion/References/whole.pdf

Ng, E. L. and Abu Hasan, Y. (2007). *Classification on Microarray Data*. Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and Applications, Vol. IV, Yahya Abu Hasan, Adli Mustafa and Zarita Zainuddin (edi), ISBN 963-3391-89-3.

Ng, E. L. and Abu Hasan, Y. (2008). *Evaluation Method in Random Forest as Applied to Microarray Data*. Malaysian Journal of Mathematical Sciences **2(2)**, 73-81.

Ng, V. W. and Breiman, L. (2005). *Bivariate variable selection for classification problem*. Technical Report 692, Department of Statistics, University of California, Berkeley. Available access: http://oz.berkeley.edu/tech-reports/692.pdf

Olson, D. L. and Delen, D. (2008). *Advanced Data Mining Techniques*. Springer, New York.

Pang, H., Lin, A., Holford, M., Enerson, B. E., Lu, B., Lawton, M. P., Floyd, E., and Zhao, H. (2006). *Pathway analysis using random forests classification and regression*. Bioinformatics **22(16),** 2028-2036.

Pearson, R. K. (2005). *Mining Imperfect Data: Dealing with Contamination and Incomplete Records*. SIAM. Available access: http://books.google.com.my/books?id=4FH1QJFMRzEC

Penga, S., Xub, Q., Lingc, X. B., Pengd, X., Dua, W., Chen, L. (2003). *Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines*. Federation of European Chemical Studies **555,** 358-362.

Piatetsky-Shapiro, G., Ramaswamy, S. and Khabaza, T. (2003). *Capturing Best Practice for Microarray Gene Expression Data Analysis*. Available access: http://www.kdnuggets.com/dmcourse/data_mining_course/microarray-best-practice.pdf

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., Kim, J. Y. H., Goumnerova, L. C., Black, P. M., Lau, C., Allen, J. C., Zagzag, D., Olson, J. M., Curran, T., Wetmore, C., Biegel, J. A., Poggio, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, G., Louis, D. N., Mesirov, J. P., Lander, E. S. and Golub, T. R. (2002). *Prediction of Central Nervous System Embryonal Tumour Outcome Based on Gene Expression*. Nature **415**, 436-442.

Samb, A. (2005). Array Based Cancer Diagnostics: *Gene Expression Profiling of DNA Microarray Data*. DPS.

Schena, M., Shalon, D., Davis, R. W. and Brown, P. O. (1995). *Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray*. Science **270**, 467-470.

Shipp, M. A., et al. (2002). *Diffuse Large B-cell Lymphoma Outcome Prediction by Gene expression Profiling and Supervised Machine Learning*. Nature Medicine, **8(1)**, 68-74.

Silva, P. J. S., Hashimoto, R F., Kim, S., Barrera, J., Brandao, L. O., Suh, E., Dougherty, E. R. (2005). *Feature selection algorithms to find strong genes*. Pattern Recognition Letters **26**, 1444–1453.

Smyth, G. K. (2004). *Linear models and empirical Bayes methods for assessing differential expression in microarray experiments*. Statistical Applications in Genetics and Molecular Biology, **3(1)**, Article 3.

Spiegel, M. R. and Larry J. Stephens. L. J. (1999). *Schaum's Outline of Theory and Problems of Statistics: Theory and Problems of Statistics*. McGraw-Hill Professional. Available access: http://books.google.com.my/books?id=a6m_I4a2fmsC

Teschendorff, A. E., Ali Naderi, A., Nuno L. Barbosa-Morais, N. L. and Carlos Caldas, C. (2006). *PACK: Profile Analysis using Clustering and Kurtosis to find molecular classifiers in cancer*. Bioinformatics **22**, 2269-2275.

The Statistics Homepage (2003). Electronics Textbook StatSoft, © Copyright StatSoft, Inc. Available access: http://www.statsoft.com/textbook/stathome.html

Tibshirani .R., Hastie, T., Narasimhan, B. and Chu, G. (2002). *Diagnosis of multiple cancer types by shrunken centroids of gene expression*. PNAS **99,** 6567-6572.

Twyman, R. (2002). Wellcome Trust, *DNA arrays and cancer classification*. Available access: http://www.wellcome.ac.uk/en/genome/tacklingdisease/hg10f002.html

Witten, I. H. and Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Academic Press.

Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K. and Zhao, H. (2003). *Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data*. Bioinformatics **19**, 1636-1643.

Xiao, Y., Frisina, R., Gordon, A., Klebanov, L. and Yakovlev, A. (2004). *Multivariate search for differentially expressed gene combinations*. BMC Bioinformatics*,* **5(164).**

Ye, N. (2004). *The Handbook of Data Mining*. Routledge. Available access: http://books.google.com.my/books?id=tABuaqVTf3MC

Yeung, K. Y., Bumgarner, R. E. and Raftery, A. E. (2005). *Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data*. Bioinformatics **21(10),** 2394-2402.

APPENDIX A

Complete names of classes in each dataset.

Brain tumor dataset:

Class	Complete Name
MED	Medulloblastoma
EPD	Normal Cerebellum
MGL	Malignant Glioblastoma
RHB	AT/RT (Rhabdoid)
JPA	PNET

Central nervous system dataset:

Class	Descriptions
Survivor	Patients who survived treatment
Failure	Patients who succumbed to treatment

Diffuse Large B-Cell Lymphoma dataset:

Class	Complete Name
DLBCL	Diffuse Large B-Cell Lymphoma
FL	Follicular Lymphoma

Leukemia dataset:

Class	Complete Name
ALL	Acute Lymphoblastic Leukemia
MLL	Myeloid/Lymphoid or Mixed-Lineage leukemia
AML	Acute Myelogenous Leukemia

Lung cancer dataset:

Class	Complete Name
ADEN	Lung Adenocarcinomas
SQUA	Squamous Cell Lung Carcinomas
COID	Pulmonary Carcinoids
SCLC	Small-Cell Lung Carcinomas
NORMAL	Normal Lung

APPENDIX B

Threshold data

```
Clear["@"]
Directory[];
SetDirectory["D:\\Brain_data"];
DataPts=Import["pp5i_train.gr.txt","Table"];

m=7071;
n=70;
R=Table[0,{m},{n}];
i=1;
j=1;
Do[R[[i,j]]=DataPts[[i,j]],{i,1,m},{j,1,n}];

R//MatrixForm;

i=2;
j=2;
Do[
        If[
        R[[i,j]]\[LessEqual]20,R[[i,j]]=20,
        If[R[[i,j]]\[GreaterEqual]16000,R[[i,j]]=16000,R[[i,j]]=R[[i,j]]
        ]
        ]
  ,{i,2,m},{j,2,n}
  ];

R//MatrixForm;

RT=Transpose[R];

Export["pp5i_train.dat",RT,"Table"];
RT//MatrixForm;
```

APPENDIX C

Filter data

```
Clear["@"]
Directory[];
SetDirectory["D:"];
DataPts=Import["Brain_data\pp5i_train.norm.dat","Table"];
m=7071;
n=70;
i=1;j=1;
R=Table[0,{m},{n}];
i=1;
j=1;
Do[R[[i,j]]=DataPts[[i,j]],{i,1,m},{j,1,n}];
R//MatrixForm;
Taking submatrix to perform calculation
Data1=Take[DataPts,{2,m},{2,n}];
Taking individual rows of genes
k=1;
S=Table[0,{k,1,m-1}];
Do[S[[k]]=Data1[[k,All]],{k,1,m-1}];
Calculating the fold difference, FD
T=Table[0,{7070}];
T=N[Table[Max[S[[k]]]/Min[S[[k]]],{k,1,m-1}]];
Eliminating genes with FD less than 2
U=Position[T,_?(#<2 &)];
V=Table[0,{m-657},{n}];
V=Delete[R,U];
Length[V]
Export["pp5i_train.cnorm.dat",V,"Table"];
6414
Directory[];
SetDirectory["D:\\Brain_data"];
Adding class
V=Import["Brain_data\pp5i_train.cnorm.dat","Table"];
Length[V]
6413
ClassData=Import["Brain_classnames.txt","Table"];
W=Transpose[V];
W1=Table[0,{70},{Length[V]+1}];
Do[W1[[i,j]]=W[[i,j]],{i,1,70},{j,1,Length[V]}];
Do[W1[[i,Length[V]+1]]=ClassData[[i,1]],{i,1,70}];
Export["brainormfilcno.dat",W1,"Table"];
Length[W]
```

APPENDIX D

Counting kurtosis

```
Clear["@"]
Directory[];
SetDirectory["D:\\Brain_data"];
DataPts=Import["brainormfilcno.dat","Table"];
Classno=Import["Brain_classno.txt","Table"];
<<Statistics`DescriptiveStatistics`
<<Statistics`ContinuousDistributions`
top50genesnew=
   Column[DataPts,{1434,3402,2928,5232,1324,2110,4138,1565,159,2103,4778,
      1997,3745,2818,2716,2910,4082,5601,2672,5819,490,3155,3897,5866,5865,
      2063,489,5136,4763,413,3148,5383,1697,3681,4917,1430,5793,2707,4475,
      5856,2532,3396,2069,4868,2093,1240,1534,4718,3844,2814}];
Export["top50genesnew.dat",top50genesnew,"Table"];
R=Import["top50genesnew.dat","Table"];
Dimensions[top50genesnew]
{69,50}
m=69;n=50;
avg=N[Table[Mean[R]]]
{42.9275,90.4348,264.246,299.493,95.6377,214.884,156.014,261.42,216.957,289.\
377,189.696,100.928,161.855,101.449,112.478,279.667,699.377,1149.54,103.696,\
141.957,123.812,86.3043,127.145,58.2029,51.1449,143.203,188.362,239.391,110.\
42,226.043,111.058,1480.59,532.855,135.362,129.145,143.928,359.406,192.333,\
109.652,484.159,586.696,151.768,69.3768,90.0435,54.6087,247.594,313.348,122.\
565,248.188,207.391}
stdev=StandardDeviation[R];
\!\(K = \(valuekurtosis =
     N[Sum[\((R[\([m, n]\)] - avg[\([n]\)])\)\^4]/\((69*stdev\^4)\) - 3]\)\)
{4.75232,-2.06966,-2.98463,-2.99733,-2.26446,-2.99739,-2.99357,-2.98643,-2.\
94426,-2.99832,-2.99597,4.03317,-2.90731,-2.96994,-2.81032,-2.99855,-2.99992,-\
2.99994,-2.93064,-2.84424,-2.4713,-2.94417,-2.97692,9.63261,15.3555,-2.95651,-\
2.93581,-2.99863,-2.56875,-2.98302,-2.66422,-2.99993,-2.99596,-2.9654,-2.\
96146,-2.96181,-3.,-2.99675,-2.89065,-2.99704,-2.99908,-1.92594,5.4995,-2.\
7302,-0.772509,-2.97185,-2.9977,-2.25015,-2.99791,-2.99672}
Omitting genes with kurtosis more than -2
U=Position[K,_?(-2<#&)]
L=Length[U]
V=Table[0,{m},{n-L}];
Dimensions[V];
V=Delete[RT,U];
W=Transpose[V];
Dimensions[W]
Export["brainclearkurtosisnew.dat",W,"Table"];
{{1},{12},{24},{25},{42},{43},{45}}
7
{69,43}
Clear["@"]
```

```
m=69;n=43;
R=Import["brainclearkurtosisnew.dat","Table"];
Rnew=Table[0,{m},{n+1}];
Do[Rnew[[i,j]]=R[[i,j]],{i,1,m},{j,1,n}];
Do[Rnew[[i,n+1]]=Classno[[i,1]],{i,1,m}];
Export["brainclearkurtosiscno1.dat",Rnew,"Table"];
```

APPENDIX E

Selecting top 50 genes with highest t-value

```
Clear["@"]
Directory[];
SetDirectory["D:\\Brain_data"];
R=Import["brainclearkurtosiswgenesnew.dat","Table"];
ClassData=Import["brain_ClassData.dat","Table"];
DataPts=Transpose[R];
Dimensions[DataPts]
{43,70}
m=43;n=70;
Data1=Table[Take[DataPts,{1,m},{2,40}]];   (*MED*)
Data2=Table[Take[DataPts,{1,m},{41,47}]]; (*MGL*)
Data3=Table[Take[DataPts,{1,m},{48,54}]]; (*RHB*)
Data4=Table[Take[DataPts,{1,m},{55,64}]]; (*EPD*)
Data5=Table[Take[DataPts,{1,m},{65,70}]]; (*JPA*)
Genesnames=Table[Take[DataPts,{1,m},{1}]];
\!\(NN1 = 39; NN2 = 7; NN3 = 7; NN4 = 10;
 NN5 = 6;\[IndentingNewLine]\[IndentingNewLine]
 \(NN11 = Plus @@ {NN2, NN3, NN4, NN5};\)\[IndentingNewLine]
 \(NN22 = Plus @@ {NN1, NN3, NN4, NN5};\)\[IndentingNewLine]
 \(NN33 = Plus @@ {NN1, NN2, NN4, NN5};\)\[IndentingNewLine]
 \(NN44 = Plus @@ {NN1, NN2, NN3, NN5};\)\[IndentingNewLine]
 \(\(NN55 = Plus @@ {NN1, NN2, NN3, NN4};\)\(\[IndentingNewLine]\)
 \)\[IndentingNewLine]
 \(Sumval1 =
    Table[Apply[Plus, Data1[\([k]\)]], {k, 1, m}];\)\[IndentingNewLine]
 \(Sumval2 =
    Table[Apply[Plus, Data2[\([k]\)]], {k, 1, m}];\)\[IndentingNewLine]
 \(Sumval3 =
    Table[Apply[Plus, Data3[\([k]\)]], {k, 1, m}];\)\[IndentingNewLine]
 \(Sumval4 =
    Table[Apply[Plus, Data4[\([k]\)]], {k, 1, m}];\)\[IndentingNewLine]
 \(\(Sumval5 =
    Table[Apply[Plus, Data5[\([k]\)]], {k, 1, m}];\)\(\[IndentingNewLine]\)
 \)\[IndentingNewLine]
 \(Sumval11 =
    Table[Plus @@ {Sumval2[\([k]\)], Sumval3[\([k]\)], Sumval4[\([k]\)],
        Sumval5[\([k]\)]}, {k, 1, m}];\)\[IndentingNewLine]
 \(Sumval22 =
    Table[Plus @@ {Sumval2[\([k]\)], Sumval3[\([k]\)], Sumval4[\([k]\)],
        Sumval5[\([k]\)]}, {k, 1, m}];\)\[IndentingNewLine]
 \(Sumval33 =
    Table[Plus @@ {Sumval2[\([k]\)], Sumval3[\([k]\)], Sumval4[\([k]\)],
        Sumval5[\([k]\)]}, {k, 1, m}];\)\[IndentingNewLine]
 \(Sumval44 =
    Table[Plus @@ {Sumval2[\([k]\)], Sumval3[\([k]\)], Sumval4[\([k]\)],
        Sumval5[\([k]\)]}, {k, 1, m}];\)\[IndentingNewLine]
```

```
\(\(Sumval55 =
   Table[Plus @@ {Sumval2[\([k]\)], Sumval3[\([k]\)], Sumval4[\([k]\)],
       Sumval5[\([k]\)]}, {k, 1, m}];\)\(\[IndentingNewLine]\)
\)\[IndentingNewLine]
\(Sumsq1 =
   Table[Apply[Plus, Data1[\([k]\)]\^2], {k, 1, m}];\)\[IndentingNewLine]
\(Sumsq2 =
   Table[Apply[Plus, Data2[\([k]\)]\^2], {k, 1, m}];\)\[IndentingNewLine]
\(Sumsq3 =
   Table[Apply[Plus, Data3[\([k]\)]\^2], {k, 1, m}];\)\[IndentingNewLine]
\(Sumsq4 =
   Table[Apply[Plus, Data4[\([k]\)]\^2], {k, 1, m}];\)\[IndentingNewLine]
\(\(Sumsq5 =
   Table[Apply[Plus, Data5[\([k]\)]\^2], {k, 1,
     m}];\)\(\[IndentingNewLine]\)
\)\[IndentingNewLine]
\(Sumsq11 =
   Table[Plus @@ {Sumsq2[\([k]\)], Sumsq3[\([k]\)], Sumsq4[\([k]\)],
       Sumsq5[\([k]\)]}, {k, 1, m}];\)\[IndentingNewLine]
\(Sumsq22 =
   Table[Plus @@ {Sumsq1[\([k]\)], Sumsq3[\([k]\)], Sumsq4[\([k]\)],
       Sumsq5[\([k]\)]}, {k, 1, m}];\)\[IndentingNewLine]
\(Sumsq33 =
   Table[Plus @@ {Sumsq1[\([k]\)], Sumsq2[\([k]\)], Sumsq4[\([k]\)],
       Sumsq5[\([k]\)]}, {k, 1, m}];\)\[IndentingNewLine]
\(Sumsq44 =
   Table[Plus @@ {Sumsq1[\([k]\)], Sumsq2[\([k]\)], Sumsq3[\([k]\)],
       Sumsq5[\([k]\)]}, {k, 1, m}];\)\[IndentingNewLine]
\(\(Sumsq55 =
   Table[Plus @@ {Sumsq1[\([k]\)], Sumsq2[\([k]\)], Sumsq3[\([k]\)],
       Sumsq4[\([k]\)]}, {k, 1, m}];\)\(\[IndentingNewLine]\)
\)\[IndentingNewLine]
\(Avg1 = N[Table[Sumval1[\([k]\)]/NN1, {k, 1, m}]];\)\[IndentingNewLine]
\(Avg2 = N[Table[Sumval2[\([k]\)]/NN2, {k, 1, m}]];\)\[IndentingNewLine]
\(Avg3 = N[Table[Sumval3[\([k]\)]/NN3, {k, 1, m}]];\)\[IndentingNewLine]
\(Avg4 = N[Table[Sumval4[\([k]\)]/NN4, {k, 1, m}]];\)\[IndentingNewLine]
\(\(Avg5 =
   N[Table[Sumval5[\([k]\)]/NN5, {k, 1, m}]];\)\(\[IndentingNewLine]\)
\)\[IndentingNewLine]
\(Avg11 = N[Table[Sumval11[\([k]\)]/NN11, {k, 1, m}]];\)\[IndentingNewLine]
\(Avg22 = N[Table[Sumval22[\([k]\)]/NN22, {k, 1, m}]];\)\[IndentingNewLine]
\(Avg33 = N[Table[Sumval33[\([k]\)]/NN33, {k, 1, m}]];\)\[IndentingNewLine]
\(Avg44 = N[Table[Sumval44[\([k]\)]/NN44, {k, 1, m}]];\)\[IndentingNewLine]
\(\(Avg55 =
   N[Table[Sumval55[\([k]\)]/NN55, {k, 1, m}]];\)\(\[IndentingNewLine]\)
\)\[IndentingNewLine]
\(Stdev1 =
   N[Table[\((\((NN1*Sumsq1[\([k]\)] - \
Sumval1[\([k]\)]*Sumval1[\([k]\)])\)/\((NN1*\((NN1 - 1)\))\))\)\^\(1/2\), {k,
       1, m}]\ ;\)\[IndentingNewLine]
```

```
 \(Stdev2 =
    N[Table[\((\((NN2*Sumsq2[\([k]\)] - \
Sumval2[\([k]\)]*Sumval2[\([k]\)])\)/\((NN2*\((NN2 - 1)\))\)\)\^\(1/2\), {k,
         1, m}]]\ ;\)\[IndentingNewLine]
 \(Stdev3 =
    N[Table[\((\((NN3*Sumsq3[\([k]\)] - \
Sumval3[\([k]\)]*Sumval3[\([k]\)])\)/\((NN3*\((NN3 - 1)\))\)\)\^\(1/2\), {k,
         1, m}]]\ ;\)\[IndentingNewLine]
 \(Stdev4 =
    N[Table[\((\((NN4*Sumsq4[\([k]\)] - \
Sumval4[\([k]\)]*Sumval4[\([k]\)])\)/\((NN4*\((NN4 - 1)\))\)\)\^\(1/2\), {k,
         1, m}]]\ ;\)\[IndentingNewLine]
 \(\(Stdev5 =
    N[Table[\((\((NN5*Sumsq5[\([k]\)] - \
Sumval5[\([k]\)]*Sumval5[\([k]\)])\)/\((NN5*\((NN5 - 1)\))\)\)\^\(1/2\), {k,
         1, m}]]\ ;\)\(\[IndentingNewLine]\)
 \)\[IndentingNewLine]
 \(Stdev11 =
    N[Table[\((\((NN11*Sumsq11[\([k]\)] - \
Sumval11[\([k]\)]*Sumval11[\([k]\)])\)/\((NN11*\((NN11 - 1)\))\)\)\^\(1/2\), \
{k, 1, m}]]\ ;\)\[IndentingNewLine]
 \(Stdev22 =
    N[Table[\((\((NN22*Sumsq22[\([k]\)] - \
Sumval22[\([k]\)]*Sumval22[\([k]\)])\)/\((NN22*\((NN22 - 1)\))\)\)\^\(1/2\), \
{k, 1, m}]]\ ;\)\[IndentingNewLine]
 \(Stdev33 =
    N[Table[\((\((NN33*Sumsq33[\([k]\)] - \
Sumval33[\([k]\)]*Sumval33[\([k]\)])\)/\((NN33*\((NN33 - 1)\))\)\)\^\(1/2\), \
{k, 1, m}]]\ ;\)\[IndentingNewLine]
 \(Stdev44 =
    N[Table[\((\((NN44*Sumsq44[\([k]\)] - \
Sumval44[\([k]\)]*Sumval44[\([k]\)])\)/\((NN44*\((NN44 - 1)\))\)\)\^\(1/2\), \
{k, 1, m}]]\ ;\)\[IndentingNewLine]
 \(Stdev55 =
    N[Table[\((\((NN11*Sumsq55[\([k]\)] - \
Sumval55[\([k]\)]*Sumval55[\([k]\)])\)/\((NN55*\((NN11 - 1)\))\)\)\^\(1/2\), \
{k, 1, m}]]\ ;\)\[IndentingNewLine]
 \)
T-Value
For Class MED :
\!\(\(TValue1 = Table[0, {m}];\)\[IndentingNewLine]
 \(N[Table[
      Do[TValue1[\([k]\)] =
         If[Stdev1[\([k]\)] \[Equal] 0 &&
            Stdev11[\([k]\)] \[Equal]
             0, \(-1\), \((Avg1[\([k]\)] -
               Avg11[\([k]\)])\)/\((Stdev1[\([k]\)]*Stdev1[\([k]\)]/NN1 \
+ Stdev11[\([k]\)]*Stdev11[\([k]\)]/NN11)\)\^\(1/2\)], {k, 1,
         m}]]];\)\[IndentingNewLine]
 \(TValue1;\)\)
```

```
For Class MGL :
\!\(\(TValue2 = Table[0, {m}];\)\[IndentingNewLine]
  \(N[Table[
       Do[TValue2[\([k]\)] =
          If[Stdev2[\([k]\)] \[Equal] 0 &&
             Stdev22[\([k]\)] \[Equal]
               0, \(-1\), \((Avg2[\([k]\)] -
                 Avg22[\([k]\)])\)/\((Stdev2[\([k]\)]*Stdev2[\([k]\)]/NN2 \
+ Stdev22[\([k]\)]*Stdev22[\([k]\)]/NN22)\)\^\(1/2\)], {k, 1,
          m}]]];\)\[IndentingNewLine]
  \(TValue2;\)\)
For Class RHB :
\!\(\(TValue3 = Table[0, {m}];\)\[IndentingNewLine]
  \(N[Table[
       Do[TValue3[\([k]\)] =
          If[Stdev3[\([k]\)] \[Equal] 0 &&
             Stdev33[\([k]\)] \[Equal]
               0, \(-1\), \((Avg3[\([k]\)] -
                 Avg33[\([k]\)])\)/\((Stdev3[\([k]\)]*Stdev3[\([k]\)]/NN3 \
+ Stdev33[\([k]\)]*Stdev33[\([k]\)]/NN33)\)\^\(1/2\)], {k, 1,
          m}]]];\)\[IndentingNewLine]
  \(TValue3;\)\)
For Class EPD :
\!\(\(TValue4 = Table[0, {m}];\)\[IndentingNewLine]
  \(N[Table[
       Do[TValue4[\([k]\)] =
          If[Stdev4[\([k]\)] \[Equal] 0 &&
             Stdev44[\([k]\)] \[Equal]
               0, \(-1\), \((Avg4[\([k]\)] -
                 Avg44[\([k]\)])\)/\((Stdev4[\([k]\)]*Stdev4[\([k]\)]/NN4 \
+ Stdev44[\([k]\)]*Stdev44[\([k]\)]/NN44)\)\^\(1/2\)], {k, 1,
          m}]]];\)\[IndentingNewLine]
  \(TValue4;\)\)
For Class JPA :
\!\(\(TValue5 = Table[0, {m}];\)\[IndentingNewLine]
  \(N[Table[
       Do[TValue5[\([k]\)] =
          If[Stdev5[\([k]\)] \[Equal] 0 &&
             Stdev55[\([k]\)] \[Equal]
               0, \(-1\), \((Avg5[\([k]\)] -
                 Avg55[\([k]\)])\)/\((Stdev5[\([k]\)]*Stdev5[\([k]\)]/NN5 \
+ Stdev55[\([k]\)]*Stdev55[\([k]\)]/NN55)\)\^\(1/2\)], {k, 1,
          m}]]];\)\[IndentingNewLine]
  \(TValue5;\)\)
Class MED
(*Top 20 TValue*)
highest20TVal1=Ordering[TValue1,-20];
genes20TVal1=Table[Genesnames[[highest20TVal1]]];
valtop20TVal1=Table[TValue1[[highest20TVal1]]];
```

```
top20TVal1=Table[0,{20},{2}];
Do[top20TVal1[[i,1]]=genes20TVal1[[i,1]],{i,1,20}];
Do[top20TVal1[[i,2]]=valtop20TVal1[[i]],{i,1,20}];
top20TVal1;

Class MGL
(*Top 20 TValue*)
highest20TVal2=Ordering[TValue2,-20];
genes20TVal2=Table[Genesnames[[highest20TVal2]]];
valtop20TVal2=Table[TValue2[[highest20TVal2]]];

top20TVal2=Table[0,{20},{2}];
Do[top20TVal2[[i,1]]=genes20TVal2[[i,1]],{i,1,20}];
Do[top20TVal2[[i,2]]=valtop20TVal2[[i]],{i,1,20}];
top20TVal2;
Class RHB
(*Top 20 TValue*)
highest20TVal3=Ordering[TValue3,-20];
genes20TVal3=Table[Genesnames[[highest20TVal3]]];
valtop20TVal3=Table[TValue3[[highest20TVal3]]];

top20TVal3=Table[0,{20},{2}];
Do[top20TVal3[[i,1]]=genes20TVal3[[i,1]],{i,1,20}];
Do[top20TVal3[[i,2]]=valtop20TVal3[[i]],{i,1,20}];
top20TVal3;
Class EPD
(*Top 20 TValue*)
highest20TVal4=Ordering[TValue4,-20];
genes20TVal4=Table[Genesnames[[highest20TVal4]]];
valtop20TVal4=Table[TValue4[[highest20TVal4]]];

top20TVal4=Table[0,{20},{2}];
Do[top20TVal4[[i,1]]=genes20TVal4[[i,1]],{i,1,20}];
Do[top20TVal4[[i,2]]=valtop20TVal4[[i]],{i,1,20}];
top20TVal4;
Class JPA
(*Top 20 TValue*)
highest20TVal5=Ordering[TValue5,-20];
genes20TVal5=Table[Genesnames[[highest20TVal5]]];
valtop20TVal5=Table[TValue5[[highest20TVal5]]];

top20TVal5=Table[0,{20},{2}];
Do[top20TVal5[[i,1]]=genes20TVal5[[i,1]],{i,1,20}];
Do[top20TVal5[[i,2]]=valtop20TVal5[[i]],{i,1,20}];
top20TVal5;
ALL TOP 20 GENES
Combinetop20genes=

Union[genes20TVal1,genes20TVal2,genes20TVal3,genes20TVal4,genes20TVal5];
Length20=Length[Combinetop20genes]
```

```
ALL20=Table[1,{Length20}];
Do[ALL20[[x]]=Position[DataPts,Combinetop20genes[[x,1]]],{x,1,Length20}];
top20genes=Table[1,{Length20}];
Do[top20genes[[x]]=First[First[ALL20[[x]]]],{x,1,Length20}]
top20=Table[1,{Length20}];
Do[top20[[x]]=Table[Extract[DataPts,{top20genes[[x]]}]],{x,1,Length20}]
pp5itop20=Transpose[top20] ;

pp5itop20gcol=Table[0,{n},{Length20+1}];
Do[pp5itop20gcol[[i,j]]=pp5itop20[[i,j]],{i,1,n},{j,1,Length20}];
Do[pp5itop20gcol[[i,Length20+1]]=ClassData[[i,1]],{i,1,n}];
Export["braintop20new.gcol.dat",pp5itop20gcol,"Table"];
Export["braintop20new.gcol.csv",pp5itop20gcol,"CSV"];
```
43

List of Publications

Ng, E. L. and Abu Hasan, Y. (2007). Classification on Microarray Data. Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and Applications, Vol. IV, Yahya Abu Hasan, Adli Mustafa and Zarita Zainuddin (edi), ISBN 963-3391-89-3.

Ng, E. L. and Abu Hasan, Y. (2008). Evaluation Method in Random Forest as Applied to Microarray Data. Malaysian Journal of Mathematical Sciences **2(2)**, 73-81.