COLON CANCER CLASSIFICATION USING MICROARRAY DATA.

by
Saima Tariq Khan

A thesis submitted in partial
fulfillment of the requirements for the
degree of M.Sc. Informatics

The British University in Dubai

Institute of Informatics

September, 2009

THE BRITISH UNIVERSITY IN DUBAI

ABSTRACT

Colon Cancer Classification using Microarray Data

by Saima Tariq Khan

Chairperson of the Supervisory Committee: **Dr. Saad Ali Amin**
Department of Informatics

A thesis presented on the classification of cancerous and normal tissue samples using microarray data. In treating cancer time is of the essence and early detection can dramatically increase the chances of survival. Imaging techniques, which are the prevalent method of detection and diagnosis, are only useful once the cancerous growth has become visible.

However, if techniques that detect cancerous processes at a genetic level are utilized then the cancerous tissues could be identified, and the disease diagnosed much earlier, thus giving a far better prognosis.

Therefore, the aim of this thesis is to evaluate the performance of a variety of different classification methods with a particular dataset containing genetic samples of both normal and cancerous biopsies of the colon tissue.

A classifier will be recommended which is able to learn the patterns within the microarray data that best determines the classification of the samples.

TABLE OF CONTENTS

L I S T o f F I G U R E S

L I S T o f T A B L E S

CHAPTER 1: INTRODUCTION

Cancer is, without a doubt, a major cause of death throughout the World: whether in richer, more developed states, where early detection is common, or those lesser-developed countries where such early warning systems are not routinely in place. According to the World Health Organization (WHO, 2009) cancer, "...accounted for 7.4 million deaths (or around 13% of all deaths worldwide) in 2004." WHO's own data projects a dramatic increase in these figures in the next few years (WHO 2007: Figure 1).

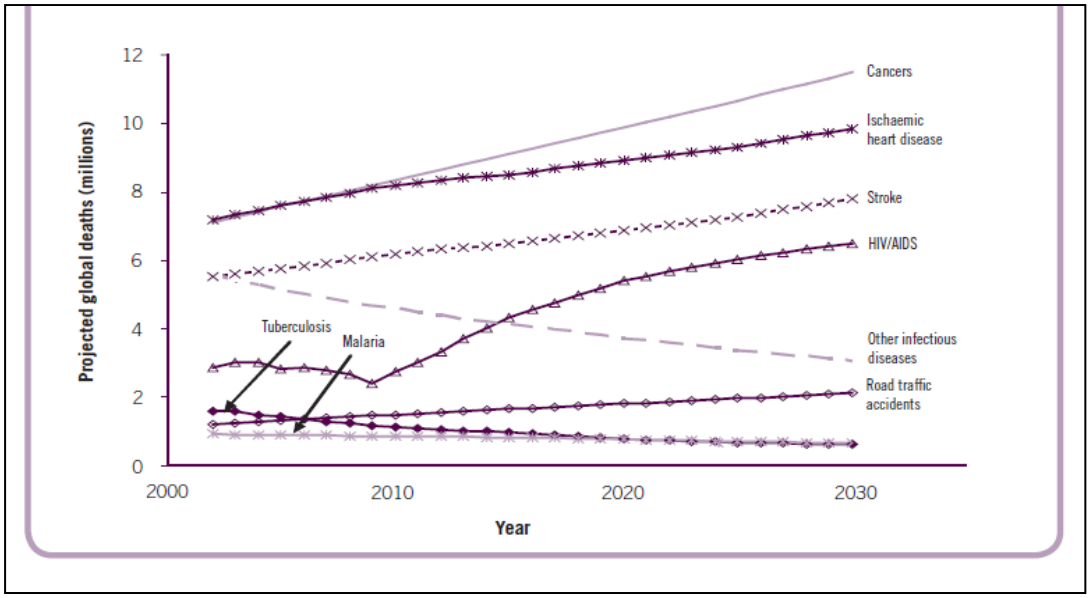


Figure 1: Projected Deaths for Selected Causes - 2030

The United Arab Emirates (U.A.E.) is no exception to this rule – statistical projections indicate that by the year 2030 more than 16% of deaths in the UAE will be due to cancer (Figure 2).

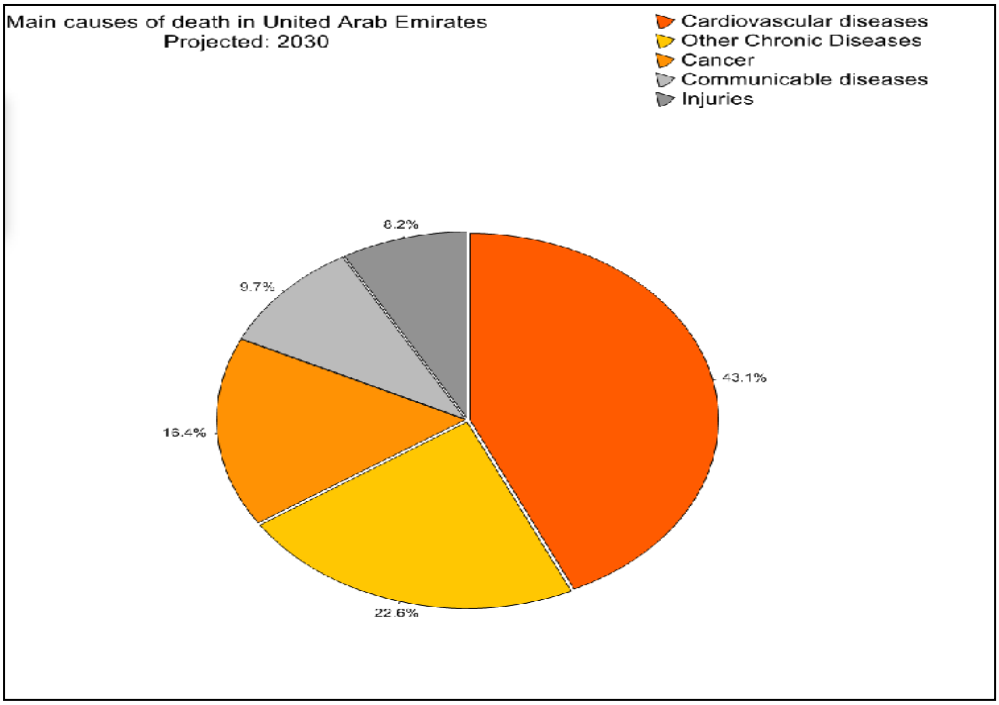


Figure 2: Main Causes of Death in the UAE - 2030 (WHO Global InfoBase).

Data from 2005 showed that of the 1200 deaths caused by cancer in the UAE that year, the large majority (1000) were under the age of seventy (WHO Global InfoBase).

Cancer is a disease that affects not only affluent countries but is also prevalent in the developing world. According to WHO (2009, p. 1), "(A)bout 72% of all cancer deaths in 2007 occurred in low and middle-income countries." Often these deaths occur in younger patients. Howard et al. (2008, p. 1), noted that, "80% of the world's children live in middle- and low-income countries (MIC and LIC), where poverty,

lack of public health infrastructure...(under 5-year mortality rates), and low childhood cancer cure rates are pervasive."

This often indicates a lack of resources for effective detection and treatment. Such a situation is particularly worrying as, according to WHO (2009, p. 2), "(A)bout one-third of the cancer burden could be decreased if cases were detected and treated early." Early detection is paramount in the battle against cancer, and has been shown to dramatically reduce mortality rates from colon, rectal, breast and uterine cervix cancers in particular (ACS, 2008; Danaei et al., 2005). Thus, the need for a more efficient and accurate diagnostic system becomes apparent.

Cancer itself is a generic term referring to a group of more than one hundred chronic diseases, which can affect any given part of the body. Cancerous cells differ from normal cells at the intra-cellular level. These abnormalities are clearly visible at this level, where cancerous tissues vary from normal tissue in a number of ways including texture, spatial arrangement, and colour, amongst other aspects (Xu et al., 2003).

With such reasons in mind, a number of image processing algorithms have been designed in order to create automatic classifiers that are able to differentiate between normal and cancerous tissues. Yuan et al. (2009) have suggested a skin lesion segmentation algorithm that utilises feature differences between cancerous and non-cancerous regions for early diagnostic purposes. Skin cancer has also been closely studied by Tang (2009), who proposed the use of multi-direction gradient vector flow (GVF) for the segmentation of skin cancer images. Often CAD (Computer-Aided Detection), or diagnosis techniques, are used as 'second readers' to aid radiologists' diagnoses. Such techniques typically make use of the fact that there are visual differences between cancerous and non-cancerous tissues. Wei et al. (2009) proposed an image-retrieval based approach to CAD, where images similar to the one being examined are used to create a classifier, which yields a malignancy

measure for the tissue under examination. Yao et al. (2009) recommended a new technique to detect colon cancer polyps (these are pre-cursors to colon cancer growths) using CT Colonography, which utilizes concepts from geographic information systems. Additionally, Hafner et al. (2009) proposed a method of classification and assessment of colonic polyps using a colour wavelet cross occurrence matrix to extract texture features. Although many of these visual techniques are currently being used to research and refine improved methods for cancer detection and classification, most of the features used for classification only become visible once the cancer has started to develop: if the disease is detected at an early stage; the prognosis for the patient is, for obvious reasons, far better.

One other method for classifying cancerous tissues occurs at a deeper level – at the level of changes in cellular expression – (i.e. in the form of genes). Genes are particles made of DNA (deoxyribonucleic acid). DNA contains protein-building instructions. It is these proteins that control the structure and function of every cell in the human body. Genes are found on chromosomes – most human cells contain two copies of each chromosome and therefore of each gene – one from each parent. If there are 'errors' in the DNA building these genes it can lead to abnormal cell growth and thus produce cancerous tissues. There are 100,000 genes that encode the human genome, which are, in turn, expressed as proteins in a two-step process.

DNA sequences are first transcribed into mRNA sequences within cells and then translated into amino acid sequences, which are used to build proteins that perform a number of different cellular functions. Different cell types express different subsets of genes and this ensures proper cell function. Normal cells can change into cancerous cells through gene mutation. The analysis of gene expression data can help identify a classification or diagnosis platform for different cancers. Such analyses can also improve our understanding of the response of cell tissues to certain drugs. In the last 2-3 decades, vast amounts of biological data have been collected

about the human genome and its functionality. This extremely large collection of data has necessitated careful storage, sequencing and indexing methods (see Appendix One (1) for details of some public domain databases).

As early as 1999, it was discovered that oligonucleotide arrays could be used to gather a snapshot of a cell's current state by monitoring the gene expression levels of thousands of genes at any given time. Research was also conducted into how to extract meaningful information from these large collections of data. One of the most useful methods discovered was the clustering of genes according to similarity in their temporal expression. This method identified functionally related groups of genes, which helped to reduce the high dimensionality of the gene array dataset (Alon et al., 1999). This study also noted the importance of developing, "the ability to process and extract useful information from large gene expression data sets." There have also been a number of different approaches towards creating an automatic classifier for the early detection of cancer. This paper will utilise a public dataset looking exclusively at colon cancer.

Researchers have also looked at using statistical techniques to predict the risk posed to any given individual (Whiteman et al., 2005). This approach is unfortunately of limited value when looking at colorectal cancer because, "75% of cases occur in people without these risk factors." (ACS, 2009, p. 4) This particular cancer is the third most commonly diagnosed cancer, and is also the, "third leading cause of cancer death in both men and women in the US." (ACS, 2008, p. 3)

Colon cancer, if detected at an early stage, can be treated with comparatively high rates of success (the five-year survival rate for patients is 90%). However, to date only 40% of patients are diagnosed at a suitably early, or preliminary, stage. Of the 49,920 people expected to die of colorectal cancers in 2009, approximately half could be prevented by early screening processes. (ACS, 2009) Colorectal cancer is one of the few cancers that can also be prevented through screening because

precancerous polyps, from which colon cancers often develop, can be identified and removed.

As such, this paper will review some of the most important approaches and evaluate the more pertinent techniques required to build a suitable classifier for colon cancer. Therefore, the aim of this study will be to analyse the aforementioned dataset, and then create a classifier based on that data. The comparative accuracy for each type of classification method will be measured and evaluated before the most suitable classifier is recommended.

This study is being carried out in the field of BioInformatics, and will involve the use of Data Mining techniques including Exploratory Data Analysis and Predictive Classification. The original dataset consists of gene samples of patients who have colon cancer and those who do not. This particular dataset contains expression levels for 2000 genes taken over 62 different samples (1.9 MB data, 529 KB names, 207 bytes labels). The samples indicate whether they were taken from tumour biopsies i.e. they carry metadata. The first task (as above) was to perform exploratory data analysis on the dataset. As new genomic data is not easily, or freely, obtained, the same dataset is often used by a number of different researchers: this particular dataset has been explored in previous studies.

In order to do so, this research will utilise Exploratory Data Analysis (EDA) including data visualisation. For data that has greater dimensions than three this becomes difficult and so the dimensions must first be decreased using projection methods. One of the most common methods to use for visualisation is the scatter plot, and as this dataset has high dimensionality, a projection technique will be used – Principal Component Analysis (PCA). The dataset will be converted to different formats in order to facilitate exploration using various software packages, such as Matlab and WEKA, for the initial exploration. These software packages are readily available, making any further research and verification of results easily replicable.

Once this step has been completed, the next stage will be to build a model that will learn from the dataset and can use this knowledge to predict values – predictive modelling, where the concept is to build a classifier which can correctly predict labels for sample data. In classification the predicted variable is categorical, whilst in regression it is quantitative. A number of different classification algorithms will be used, including, amongst others, the Nearest Neighbour Method, the Naive Bayes Model and Logical Discriminant Analysis. Once this has been completed, the performance of the various classifiers, and the results they yield, can be usefully compared and analysed.

This thesis is structured so that Chapter 2 contains a review of the salient literature (Literature Review), whilst Chapter 3 focuses on the methodology adopted. The Data Analysis and Classification techniques are featured in Chapter 4, while the results, and a discussion of the implications arising from these results, are contained in the 5th Chapter, followed by a conclusion (Chapter 6).

CHAPTER 2: LITERATURE REVIEW

2.1 MICROARRAYS

Cancer is a widespread disease affecting a large number of disparate people. As such, there is an urgent need to understand the underlying mechanism and characteristics of this potentially virulent disease in order to more efficiently detect (at as early a stage as possible) and treat this ubiquitous affliction. Most previous diagnostic methodologies have relied on human interpretation of imaging data in order to study areas of diseased tissue. These images are, more often than not, obtained by using personally invasive methods and, perhaps more significantly; before such cancerous growths manifest themselves in an easily observable way, the cancer will often have been present for a considerable length of time. Yet, this is still often the primary method available to, and therefore used by, most doctors when diagnosing any form of cancer.

However, over the past few years research into cancer genetics has indicted a link between some forms of cancer and specific 'marker' genes (Memo to the Media, 2009: Ferracin et al., 2008; Polakis et al., 2007; Chung et al., 2007; De Soto et al., 2006). These observations have led to the establishment of genetics clinics, such as those set up by the National Health Service in the U.K., the European Directory of DNA Diagnostic Laboratories and the clinics listed in the databank of the National Centre for Biotechnology Information in the US. The National Cancer Institute in the United States has also dedicated an entire section of its website to cancer genetics.

Winawer (2007), would remind us that colorectal cancer is a global disease with a steadily increasing number of cases reported every year – probably due to an ever increasing global population and a lack of effective screening facilities. This is also

echoed by (Malikzadeh et al., 2009), in their study of colorectal cancer incidence in Iran. They reported an accelerated growth rate for colon cancer in Iran's comparatively young population, and that the incidence was usually observed in close relatives of other colon cancer patients – clearly indicating a genetic link. One method for decreasing the incidence of death in colorectal cancer patients would be to devise screening technology that is, as proposed by Rennert (2009, p.1), "as simple as possible, easy to perform, cheap, and, most importantly, acceptable to the population, that is, noninvasive and with an overall balance of more benefit than harm."

Recently a new technique for diagnosing colorectal cancer has been introduced (Rennert 2009b, p. 1), which uses genetic biomarkers to identify patients. The same paper states that:

> Genetic diagnosis of colorectal cancers and meaningful adenomas has now reached a new phase that, when further fine-tuned, may carry the promise of becoming a suitable and affordable means of prevention and early detection ofcolorectal cancer in the general population.

The concept of using microarrays to detect genetic data, that can help to predict the probability of cancer, holds out the hope of many potential breakthroughs in the fight against this major disease. DNA microarrays are, according to (Eisen et al., 1999, p. 1), "valuable tools in areas of research that require the identification or (quantification) of many specific DNA sequences in complex nucleic acid samples." These arrays have been used extensively in the last few years in a wide range of studies including mutational analysis (Salvado et al., 2007; Giordano et al., 2005), genetic mapping (Drost et al., 2009; Altshuler et al., 2008), and genome wide monitoring of gene expression (Takata et al., 2005; Tamura et al., 2007).

Since the publication of the aforementioned work technical advances have made possible the generation of arrays with high densities of DNA, which means thousands of genes can now be represented in minute areas. This in turn means more data can be captured and analysed in order to give a clearer understanding of cellular activity and to assist in diagnosing and preparing a better classification system for the various forms of cancer (Lora et al., 2007; Goldstraw et al., 2007).

2.2 QUALITY OF COLON CANCER MICROARRAY DATASET

Although much of the process for creating microarrays is fully automated, there is still scope for error – e.g. some dots have 'comet tails', arrays where individual dots appear as 'donut holes,' or abnormally high fluorescent backgrounds. The various steps involved in creating these microarrays unfortunately leaves room for error, in which case the quality of the data may well be compromised to an extent. Indeed, (Brazma et al., 2000, p. 18) remind us that, "(I)n any physical experiment it is important to know not only the value of the measurement, but also the standard error or some other indicator of reliability for each data point." Spots are typically based on EST sequences; and linking the EST sequence to a particular gene is not a simple process. (see Appendix 1: EST Sequence Definition)

Wang et al. (2001) described how microarray data often carries noise and other irregularities. They mentioned five of the most common problems, including: spot size (likely due to isolated noise); signal-to-noise ratio (which quantifies how well one can resolve a true signal from system noise); local background variability; excessive high local background and saturation in photo intensity detection. The same study went on to mention that there were many steps leading to the creation of a microarray and each step had scope for error. These errors affect the quality of the final image produced and will then cause variations in the intensity readings on

which most data analysis will be based. For these reasons, their report noted that it was very important that every spot be measured qualitatively on the microarray slide. A standardized quality control measure should therefore be adopted so that reliable data can be generated for the purposes of data mining and to ensure that data can be shared with confidence across laboratories, even worldwide. At the moment the colon cancer dataset which is being studied has not been benchmarked.

2.3 DIMENSION REDUCTION

Pochet, et al. (2004) conducted experiments on microarray data using Least Squares SVMs with linear kernels and RBF kernels. They noted that for both methods dimensionality reduction (using PCA or other methods) is necessary to avoid over fitting of the data by the classifiers.

One useful method for dimension reduction is that of clustering. Eisen et al. (1998, p. 14863) note that a, "natural basis for organizing gene expression data is to group together genes with similar patterns of expression." Of course, this necessitates a mathematical measure of similarity being established, that will help to cluster genes (this is often the Euclidean distance, or the dot product between two gene vectors).

Clustering can be divided into two categories: supervised and unsupervised. In the former, vectors are classified using known reference vectors, whilst in the latter no predefined reference points are used. Supervised clustering methods include neighbourhood analysis, maximum entropy models, SAM (significance analysis of microarrays) and other ranking-based methods (Tang et al., 2002). One of the disadvantages of using supervised methods for clustering, or indeed for any form of data-mining, is the fact that they are limited to hypothesis testing and cannot reveal the, 'unexpected, (they can) never lead to new hypotheses' (Domany 2003, p.1124).

Also, if there are samples that have been misclassified before they are used for training purposes, then the supervised method will not be able to discover this anomaly. For this reason, this study will employ unsupervised clustering methods.

There are a number of different clustering methods including Average Linkage (Sorlie, et al., 2001), K-means and Self-Organising Maps (SOM) and Coupled Two Way Clustering (CTWC), as used by Alon, et al., (1999), for clustering the colon cancer dataset used in this study. Au et al. (2005) introduced the k-modes Attribute Clustering Algorithm and further research (Alon et al., 1999; Au et al., 2005) noted that Clustering can be a tool for reducing the dimensionality of a data-mining algorithm. Bi-clustering algorithms other than CTWC are also outlined by Madeira, et al., in a 2004 survey.

One of the better known types of clustering algorithms is 'hierarchical clustering,' (Perou, et al., 2000; Lonning et al., 2001) the object of which is to create a dendrogram which illustrates the different clusters in a given dataset. This algorithm has been coded by a number of researchers including those mentioned by (Eisen et al., 1998 p. 3). They also noted that there is a, "strong tendency for (genes in the same cluster) to share common roles in cellular processes."

This study will make use of the well-documented hierarchical clustering methods as a visualisation tool and will also investigate the Fuzzy Clustering Toolbox (Abonyi et al., 2004) to determine the optimal number of clusters for this dataset. The methods available in this toolbox have not been used on this particular dataset prior to this study. These methods have been chosen because they are readily available and thus easily replicable, without great expense or highly specific expertise being required.

Projection methods such as Principal Components Analysis (PCA) can also be used to reduce the dimensionality of the dataset. However, this has the disadvantage in that none of the original features can be eliminated. At the same time it is a standard method often used in Data Mining to reduce dataset dimensionality. Other methods use the concept of pruning redundant features – e.g. SVM-RFE: Support Vector Machines – Recursive Feature Elimination (Guyon, et al., 2002; Shen, et al., 2005; Hernandez, et al., 2008 and others). The SVM –RFE method was applied by Mundra, et al. (2007) with the addition of an extra criterion to the existing weight criteria integrated into their method. Additionally, this has been modified by Yousef, et al., (2007) to become SVM-RCE (Recursive Cluster Elimination), which has comparably accurate results, but is reported to be computationally more expensive.

Other schemes that can be, and have been, used for attribute extraction and dimension reduction include Linear Discriminant Analysis (LDA) and Uncorrelated Discriminant Analysis (ULDA) as proposed by Ye, et al., (2004). LDA has also been further developed by Yue, et al., (2007), where they proposed a variation called Null-Space LDA. Bayesian variable selection is mentioned by Yoo, et al. (2004), as is singular value decomposition and PCA as possible methods for dimension reduction. An SVM and GA (Genetic Algorithm) hybrid was proposed by Xiong, et al. (2006), and Li, et al. (2008) amongst others, whilst Nguyen, et al., (2006) proposed the use of Random Forests to select features before classification.

Genetic Algorithms were used with fuzzy clustering by Mukhopadhyay, et al. (2009) and on their own for feature selection. (Chandana, et al., 2009) Mukkamala, et al. (2006) experimented with the use of Regression Splines (MARS), Classification and Regression Trees (CART) and Linear Genetic Programs (LGP). Genetic Programming has also been used by other researchers, (Almal, et al., 2006; Hernandez, et al., 2007) for feature selection. Then there is Mutual Information Rough Set Theory, which is another method of variable selection proposed by Zhou,

et al. (2006). Tandem use of the T-test method and Kernel Partial Least Squares (KPLS), (Li, et al., 2006) also yield promising results for gene extraction.

Peng, et al. (2006) proposed the use of bootstrapping to select genes iteratively. Boosting (one example of which is the AdaBoost algorithm) was also developed to include a level of consistency (Pang, et al., 2007; Lausser, et al., 2008). Kernel methods including Kernel Fisher Discriminant Analysis (KFDA) and Kernel Partial Least Squares (KPLS) were evaluated by Li, et al. (2007), who noted that both these methods reached a high level of classification accuracy on the colon cancer dataset.

Of the various classification systems mentioned thus far, Lee, et al., (2007, p.180) classifies these as either linear or non-linear and found that the non-linear methods, "outperformed the corresponding linear methods". This choice of method also has an impact upon the particular classification model that is used. For instance, Lee (2007, p.180) found that classification accuracy for SVMs and C4.5 Decision Trees were, "consistently higher when using features obtained by nonlinear (Dimension Reduction) methods compared to linear methods." The non-linear methods used were, Graph Embedding, Isometric Mapping and Locally Linear Embedding, whilst the linear methods were, PCA, LDA and Classical Multidimensional Scaling.

Deegalla, et al. (2007, p. 801) also noted that, "the classification accuracy of kNN often decreases with an increase in dimensionality." For such classification algorithms dimension reduction is imperative. These researchers added Random Projection and Information Gain to the methods they compared and discovered that there was no one ideal method for the eight datasets they examined. However, they noted that for a binary class problem (such as the one being considered in this study) Partial Least Squares seemed to perform well. This same method has also been extensively used by Zeng, et al. (2007).

Chang, et al., (2007) proposed a Heuristic Branch-and-Bound Depth First Search. Another approach (Chen, et al., 2007) is to add noise to the data. The assumption, in this case, being that irrelevant features will have little influence on classification performance whilst this performance will be influenced if important features are perturbed. This method, unlike SVM-RFE, can be used with any algorithm. However, it is computationally more expensive

2.3 CHOSEN METHODS FOR DIMENSIONALITY REDUCTION

Although there are many and variable methods that could be used for the purposes of dimensionality reduction, it was decided, in order to fit the scope of the current study, that PCA would be utilised, as this is a standard method used for dimension reduction in data mining. In order to be able to contrast results a second method for dimensionality reduction was also trialed. There were several suitable choices available for attribute selection via the WEKA software package: this has been more fully explored in Chapter 3 (below).

2.4 BUILDING CLASSIFICATION MODELS

The next part of the process of building a classifier involves choosing, from amongst a variety of options, a suitable classification algorithm. Possible algorithms abound with perhaps the following being the most important of these:

Artificial Neural Network Based Classifiers (ANN) (Piatetsky-Shapiro et al., 2003; Alladi et al., 2008) and Support Vector Machines (SVM) (Mukhopadhyay et al., 2009; Zeng et al., 2007) are popular methods for microarray classification.

Decision Trees (DT) are one of the methods used by Wang et al. (2009), whilst Winkler, et al. (2009) and Chandana, et al. (2009) have used kNN (k-Nearest Neighbours) classifiers in addition to some of the aforementioned methods. k-NNs has also been used by an number of other researchers including Xu, et al. (2007); Li et al. (2006); and Rao, et al. (2003). For the colon cancer dataset this study reported that a single Nearest Neighbours (NN) classifier performed better than an ensemble.

Donoho, et al. (2008) in their examination of various feature selection methods also used 'Bagboost' and 'LogitBoost' for classification. Another similar method used for microarray classification is 'AdaBoost,' with variations such as 'MadaBoost' and 'AdaBoost-VC' (Lausser, et al., 2008) also available. Random Forests is a popular variation of decision trees and is in fact an ensemble of such trees (Nguyen et al., 2006).

Prediction Analysis for Microarrays (PAM) (Sorlie, et al., 2003) and Linear Discriminant Analysis (LDA) were explored by Meleth, et al. (2007). SAM (Significance Analysis of Microarrays) was previously used to analyse microarray datasets by Sorlie, et al. in 2001. In addition, Mukkamala, et al. (2006) utilised a number of classification methods including Regression Splines (MARS) and Classification and Regression Trees (CART). Plus, Self Organising Maps (SOM) were used by Tang, et al., in 2002 and Genetic Algorithms (GA) were used for classification by Winkler et al. (2009) and Almal et al. (2006).

Combinations of classification methods have also been investigated (Shen et al., 2005; Peng et al., 2005). The former reported that Partial Least Squares (PLS), Penalized Logistic Regression (PLR) and Singular Value Decomposition (SVD) can be combined to improve both training speed and classification accuracy. In a similar vein (Blanco et al., 2007) experimented with classifier combinations including Bagging with SVM, k-NN and DLDA (Diagonal Linear Discriminant Analysis).

Chen, et al. (2007) also proposed a classifier fusion model consisting of a combination of SVM classifiers using a fuzzy logic system assisted by genetic algorithms. Additionally, Ensemble Classifiers were employed by Zhao, et al. (2007), using an Estimation Distribution Algorithm (EDA). And, a similar method was used by Hernandez et al. (2007) and Nguyen et al. (2006) respectively where an SVM classifier was combined with Genetic Algorithms. Eduardo, et al. (2005, p. 63) combined five classifiers (k-NN, k-Means, SOM, PCA, Parzen Window) and noted that generally, "a combined approach improves the robustness of the overall (classification) decision."

Nguyen, et al. (2005), used kernel functions with SVM. They experimented with radial, neural and inverse multi-quadric kernel functions as well as a combination of all three. For the colon cancer dataset they found the best classification results were obtained with the combined kernel functions and SVM.

Marchiori, et al. (2005), used RFE-SVMs with an ensemble of classifiers in Optimal Bayes and as attributes in Naive Bayes and reported that this combination of techniques led to a more robust classification scheme. Also, Ben-Dor, et al. (2000), compared the performance of NN, Clustering, SVM and Boosting, finding that SVM, Boosting and NN performed well on the colon cancer dataset.

A comparative study by Hong, et al., in 2006 looked at various classifiers including C4.5, Random Forests, Adaboost C4.5, Bagging and LibSVMs. According to this study, the last three methods gave the best classification results for the colon cancer dataset.

Naive Bayes (NB) is another well-known data-mining algorithm, used by a number of researchers over the years including Ruiz et al., in 2006, Au et al. (2005), and Li, et al. (2003), for classification. Shen, et al. (2005), also proposed two new algorithms

– Kernel Partial Least Squares – Minimum Squared Error (KPLS-MSE) and Kernel Singular Value Decomposition – Minimum Squared Error (KSVD-MSE) – both methods are comparable in performance but do not always outperform SVMs; however they do have the advantage of having shorter training times. Radial Basis Functions were also proposed by Wang, et al. (2005) to contrast non-linear (k-NN and Gaussian SVM) and linear classification (Diagonal Linear Discriminant and Linear-SVM).

2.5 CHOSEN METHODS FOR CLASSIFICATION

The classic approach to classification is to use a linear hyperplane to define class boundaries. More complex methods allow higher order polynomial functions yielding polynomial decision boundaries. Flexible non-linear boundaries can be found using classifiers based on neural networks – piece wise linear boundaries are also possible and one example of such an algorithm is Nearest Neighbors.

When building classifiers three basic approaches are possible (Hand et al., 2001):

1. Discriminative Approach: the decision boundary is modeled directly without any calculations of posterior class probability. Examples are Perceptrons and Support Vector Machines.

2. Regression Approach: where posterior class probabilities are modeled. The most common technique used in this category is logistic regression.

3. Class-conditional Approach: The class-conditional distributions are modelled along with estimates of class probability for a given point. Examples of this type of classifier are ‘Bayesian’ classifiers.

Both the discriminative and regression approaches focus on the differences between classes whereas the class-conditional approach focuses on the distribution of the input data vector for the classes. As noted by Hand et al. (2001), in datasets with high dimensional spaces the class-conditional approach may be difficult as with such cases it is difficult to estimate functions. The same paper proposed that in high dimension classification problems discriminative approaches may be better. It is also worth noting that methods based on the discriminative approach need to fit the least parameters, the regression requires more parameter fitting and the class-conditional approach requires the most.

Therefore, it was decided that the best methodological approach was to make use of a number of different methods in combination. Highly prevalent and relevant methods include: Bayesian methods, Tree models, Linear Regression, Nearest Neighbour algorithms and Neural Nets. Therefore, these broad methods will be utilized in combination.

CHAPTER 3: METHODOLOGY

3.1 JUSTIFICATION

There are several public datasets available for analysis and this study will utilise one such dataset <http://microarray.princeton.edu/oncology/> in order to help build a classifier to detect accurately the onset of cancer using genetic information.

There are various types of microarrays, each of which uses different technological techniques to measure RNA expression levels (Piatetsky-Shapiro et al., 2003). The dataset in this study uses Affymetrix microarrays, which contain short oligonucleotide arrays. The raw microarray data is an image where the brightness of the dots in the matrix represents the intensity of gene expression. This is often translated into a numerical value before analysis takes place. The dataset used in this study has already been translated and will be used in its current form.

3.2 QUALITY OF COLON CANCER MICROARRAY DATASET

It should be stated from the outset that the actual quality of the colon cancer microarray dataset is unknown and (as above) there is always room for human, and/ or machine, error in the creation of these arrays. There were originally approximately 6000 genes represented in the array, of these, "2000 genes were selected 'based on the confidence in the measured expression levels'" (Ben-Dor et al., 2000, p. 12). In fact, Ben-Dor et al. (2000, p. 20) note that;

> Cancer classification based on array-based gene expression profiling may be complicated by the fact that clinical samples, e.g tumor vs. normal, will likely contain a mixture of different cell types. In addition, the genomic instability inherent in tumor samples may lead to random

fluctuations in gene expression patterns.

Earlier (Alan et al., 1999) also noted that the normal colon biopsy also included smooth muscle tissues from the colon walls. For this reason, smooth muscle related genes exhibited high expression levels in the normal samples.

The contamination of the dataset with muscle specific genes and the fact that tumors exhibit higher metabolic rates may affect classification performance. Ben-Dor et al. (2000), took these factors into account and noted that they only affect the results in cases where very small sets of genes were examined, highlighting the fact that with an error score threshold of 10 (genes) or higher, there was no significant change in performance. It is, therefore, clear from previous research in microarray analysis that some form of pre-processing will be necessary. Different software packages may require different formats to facilitate correct data input.

3.3 PRE-PROCESSING

One of the challenges posed by genetic data analysis is the small number of samples (rows) compared to the very large number of columns (genes). This usually leads to 'false positives' – i.e. gene combinations which correlate with a target variable by chance and, as noted by Piatetsky-Shapiro et al. (2003), this is more of a problem for learning algorithms, such as neural networks and decision trees, which typically try and find complex non-linear combinations of features in order to build classifiers. One way of dealing with this situation is feature reduction, which transforms the raw data into a form that is suitable for analysis. These data preparation techniques include thresholding, which is used to filter out noisy data and affymetrix devices that measure expression levels indirectly.

Then there is the normalization procedure, where classification algorithms are able to use the gene expression measurements just as they are. However, clustering algorithms would require this additional step, so that, in this case, data would then be normalized to mean zero and have a standard deviation of one across all genes. There is also filtering, where some genes may not be expressed at all, in which case they can be omitted from our analysis. Typically, genes that have low variation across samples are omitted.

Once the data has been prepared, the next step is the feature selection process which reduces the dimensionality of the dataset. This was originally a concept in data-mining that was used to model the physical properties of an inhomogeneous ferromagnet (Blatt et al., 1997).

Some form of normalization may benefit the performance of the classifiers. In this study filtering, attribute selection methods, normalization and PCA will all be employed in different combinations to yield datasets that will then be used with various classification methods.

3.4 SOFTWARE PACKAGES

There are, of course, many possible methods and approaches that could have been utilized before the methodology chosen was finally settled upon. Such, useful, but discarded resources have been placed in Appendix 3: Part B. An initial, but ultimately only superficial analysis was begun and these results have been recorded in the aforementioned appendix. Therefore, this study will use Matlab with the intention of building the desired classifiers. MatLab Version 7.2.0.232 – R2006a was the version chosen for this study. Apart from the Matlab software, the WEKA package has also been used. It has been selected mainly due to the following features:

1. It contains modules that implement a number of data mining algorithms; including, most notably, decision trees; K Nearest Neighbours and Naïve Bayes among others.

2. WEKA supports meta-learning schemes that are often used in Bio Informatics such as Bagging and Adaboost.

3. WEKA contains data pre-processing support modules: including those which allow for the replacement of missing attributes, discretisation of numeric values, and the removal of attributes with only one distinct value.

The WEKA package also integrates data visualization, data preparation, feature selection and data mining algorithms, all of which make it an attractive package to explore within the scope of this study. Additionally, unlike Matlab, which requires a user's license, WEKA is open source software and therefore more easily obtained, thus, making any replication of such a study easy and affordable.

Data Cleaning was also used in order to ignore missing values (Han & Kamber, 2000). Also Noisy Data (random variance in dataset) can be removed using 'binning' (a technique used in Weka) and clustering. When clustering is used in this way the outliers are considered as noise. In the Matlab function, one step was to remove genes whose variability was low.

One of the techniques used for Data Transformation is Normalization. The colon cancer dataset was used in both its original form and the normalized version. Dimension Reduction will form a large part of the pre-processing phase due to the large dimensions (2000 attributes) of the dataset. Some of the better-known

techniques for this purpose include Principal Components Analysis (PCA), Linear Discriminant Analysis and Multidimensional Scaling. PCA will be used in this study as it is a standard methodology.

Cluster Analysis does not need class labels to analyze objects. Often clustering is carried out without class labels which are not known a priori. It can therefore be used to facilitate a taxonomy formation – the organization of objects in hierarchical classes in order to group similar events. Clustering has previously been used in microarray cancer studies to distinguish sub-classes of tumour (Sorlie et al., 2001).

3.5 CLASSIFICATION METHODS

Data Mining as defined by (Han & Kamber, 2000, p. 24) encompasses the task of classification: a "process of finding a set of...functions that describe...data classes...to use the model to predict the class of objects whose class label is unknown." The derived model is created using a set of training data whose class labels are known and is presented using classification rules, decision trees or neural networks. Often, as suggested above, classification is preceded by relevance analysis, where those attributes that do not contribute to the classification process are excluded.

There are a vast number of classification methods extant. However, the purpose of this study is not an exhaustive analysis of all possible classification methods – such an undertaking is well beyond the scope of this study. Instead, the aim will be to explore, in as much detail as possible, a subset of the most prominent and popular classification methods available.

3.6 CLASSIFICATION TECHNIQUES

Eighteen different methods from the four main categories (i.e. Trees, Bayes, Functions and Meta learning) were chosen (see Appendix 3: Part A). Mitchell (1997), mentions that a well-defined learning problem should have three features: the class of tasks, the measure of performance to be improved and the source of that experience. For the classifiers that this study is seeking to build, the task will is to accurately classify samples of colorectal tissue data as to whether they are being cancerous or not.

3.7 DESCRIPTION OF DATA

The colon cancer dataset consists of the expression levels for 2000 genes taken over 62 different samples. Of these samples 22 are taken from normal colon tissue while 40 samples are taken from cancerous tissue.

3.8 CROSS-VALIDATION

Since the number of samples is comparatively small, the measure of performance used will be cross validation. The number of folds will be set to 10 since for Witten et al. (2005, p.150), "(E)xtensive tests on numerous datasets, with different learning techniques, have shown that 10 is about the right number of folds to get the best estimate of error." The source of experience is the 62 available samples, divided into both testing and training sets.

3.9 CONFUSION MATRICES

To measure the accuracy of the classifiers' performance, confusion matrices were also recorded. Evaluation by classification accuracy assumes equal error costs, but in

most cases and especially in the domain of medical diagnosis there is a far greater cost attached to, for instance, a false negative result where the patient may lose treatment time because he/ she has been misdiagnosed as not suffering from a certain disease. In this case, a potentially fatal cancer. A two-class prediction, such as the one looked at in this study, (positive = bengin, negative = cancer) has four possible outcomes as shown in the table below:

Predicted Class			
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

Table 1: Confusion Matrix

Where TP = True Positive, TN = True Negative, FP = False Positive, FN =False Negative.

In the case of the FP, which is also an incorrect result, the patient would face emotional distress when in fact they would have had no reason to worry. A problem easily solved by a second opinion/ test. The FN number is the figure that the classification algorithm should be concerned with minimizing, as far as is possible. Therefore, if a classifier exhibits a very high classification performance but an examination of the relevant confusion matrix shows a high number of FN, then that classifier is not accurate and not suitable for diagnostic purposes.

CHAPTER 4: DATA ANALYSIS

INPUT AND VISUALISATION

The first step was to read the data into MatLab (see Appendix 2: Data Preparation). The M-file output includes a 2-dimensional graphic representation of the array variable Num (Figure 3).

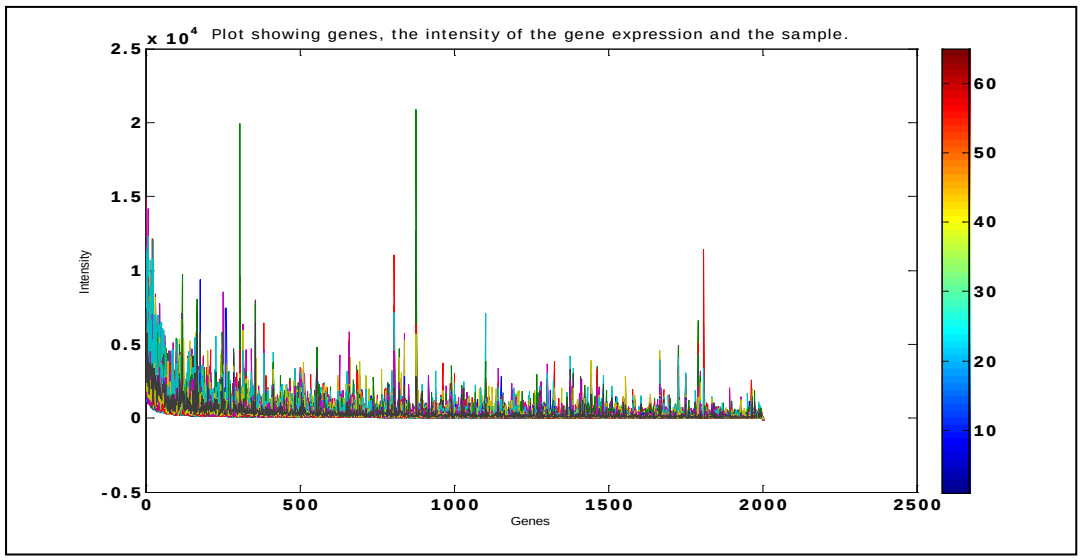


Figure 3: Gene Expression Versus Genes

Figure 3 represents each attribute with a different color – there is no reflection of interdependence of attributes. However, it shows that some gene intensities are far greater numerically than others. To deal with this problem a logarithmic graph of gene expression intensity versus genes was also plotted – Figure 4.

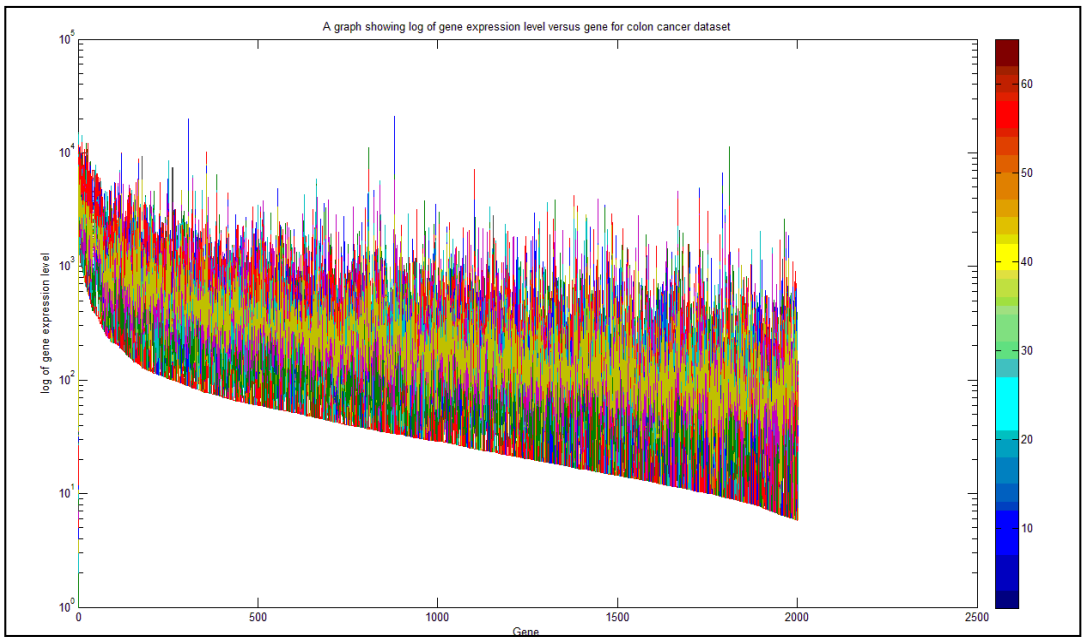


Figure 4: Log of Gene Expression Levels Versus Genes

Figure 4 seems to indicate a decrease in gene expression value throughout the dataset. It would seem that the genes are perhaps ordered in the dataset. However, there is no information available to indicate this or allow for a more informed guess.

To check if there is correlation, the Correlation Coefficients ('corrcoef') function was run. The Correlation coefficient (also called Pearson's product moment coefficient) is calculated using the following formula:

$$r_{A,B} = \frac{\sum (A - \overline{A})(B - \overline{B})}{(n-1)\sigma_A \sigma_B} = \frac{\sum (AB) - n\overline{AB}}{(n-1)\sigma_A \sigma_B}$$

Where: n is the number of records

$\overline{A}$, $\overline{B}$ are the means of *A* and *B*

σA and σB are the standard deviations of *A* and *B*

Σ(AB) is the sum of AB cross-product.

N.B.:

- If rA,B > 0, A and B are positively correlated (A's values increase as B's increases). The higher the value: the stronger correlation.
- rA,B = 0: independent
- rA,B < 0: negatively correlated

(Han J, 2009, slide 40)

Once the correlation coefficient matrix has been calculated the correlation coefficient values can be plotted against the attribute number (see Figure 5), where both axes represent the genes and the color represents the value of the correlation coefficients.

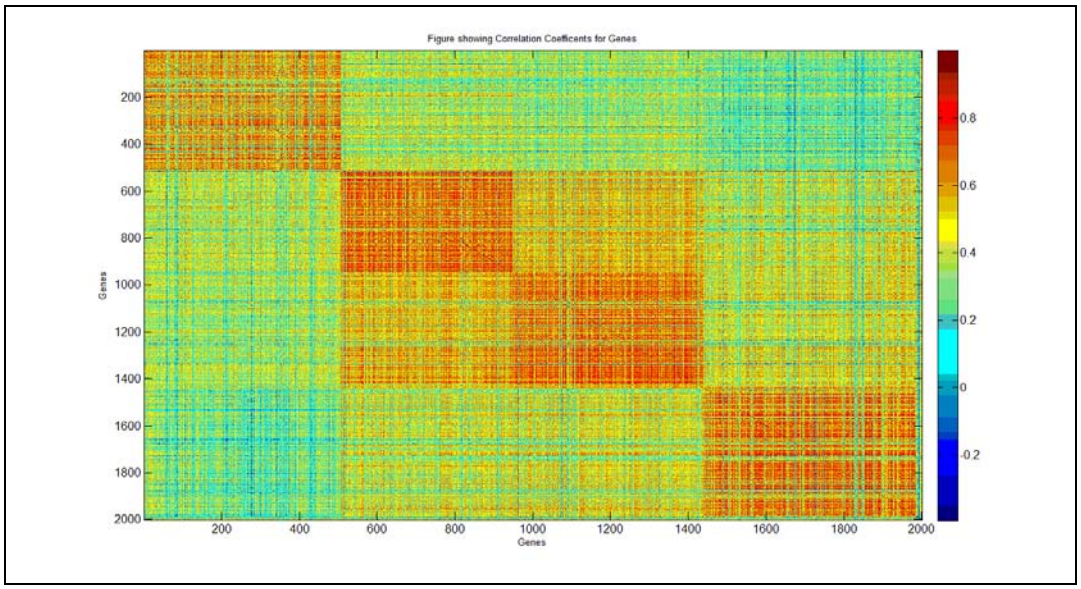


Figure 5: Plot of Correlation Coefficients.

The correlation coefficient is one measure of the relatedness of variables. Since the dataset that is being studied is one that contains biological data it was expected that there would be a high degree of correlation between attributes. From Figure 5 it can be deduced that this is indeed the case, since most values fall between the values of zero (no correlation) and one (strong positive correlation). There are also some negative correlations, but the maximum value for this is negative 0.4, which means the negative correlation between those particular genes is not very strong.

Since there are positive correlations between almost all variables in the dataset, it can be assumed that data reduction can be performed satisfactorily. This would mean that, upon traversing the dataset, groups of related attributes can be found and represented more concisely.

4.1 PART 1A: Initial Analysis of Data Which Has Not Been Normalized

The dataset was saved in the following form:

I. **FILE: colorectal_dataset.mat,** which contains the following variables:

Name	Size	Bytes	Class	Data
colon_class	1x62	496	Double	+1 or -1
colon_genes	2000x1	144024	Cell	Gene names
colon_genevalues	2000x62	992000	Double	Expressions
colon_samples	1x62	496	Double	Samples

FILTERING

The 'genevarfilter' function removed genes with low expression variability – a good indication that the gene is not actively involved in the process. This reduced the number of genes from 2000 to 1800. Next the 'geneentropyfilter' function was used to remove genes whose profiles showed energy levels in the bottom 15^{th} percentile of the dataset. This further reduced the number of genes to 1530. The reduced dataset was saved in following file:

II. **FILE: colorectal_dataset_reduced.mat**

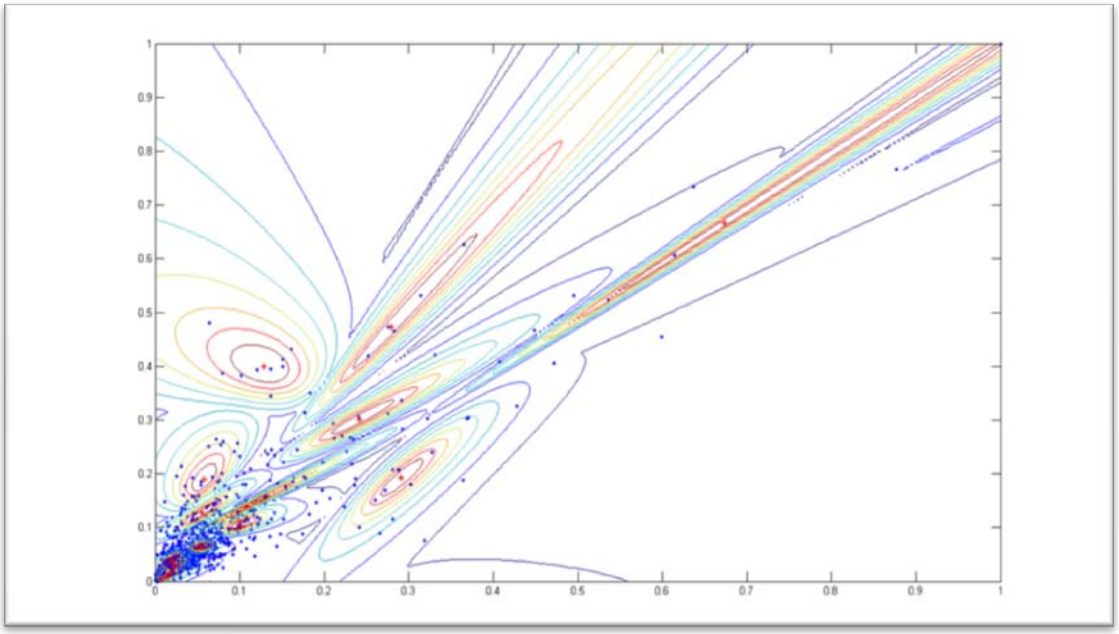
Name	Size	Bytes	Class	Data
colon_genes_reduced	1530x1	110182	cell	Gene names
colon_genevalues_reduced	1530x62	758880	double	Expressions

CLUSTERING

To help in visualizing and summarizing data clustering methods were used (Kumar et al., 2007). Most clustering algorithms require the number of clusters as an input argument. Therefore, in order to determine the optimal number of clusters the function ‘optnum’ was utilized. This came from by Abonyi et al.’s (2004) ‘Fuzzy Clustering Toolbox’.

This function uses the Gustafson-Kessel (GK) fuzzy clustering method (with squared Mahalanobis as the distance measure). GK is an extension of the standard fuzzy c-means (FCM) algorithm. The FCM algorithm uses Euclidean distance as a distance measure – which means that it can only detect clusters with a similar shape and orientation. The Gustaffson-Kessel algorithm modifies KCM to use an adaptive distance measure which allows for different geometrical clusters to be discovered in a given dataset (Abonyi et al., 2004).

The maximum number of clusters is set at 14, which is thus the default value. Figure 6 shows this algorithm dynamically creating clusters. The x and y axes show values for the data which is first normalized by the algorithm. The data points are represented as dots in the plot. The circular formations are the decision boundaries



calculated for each cluster by the algorithm which iteratively seeks the optimal number of clusters. During this process seven clustering validity measures were calculated by the sub-function labelled, 'modvalidity.' These functions were run on the original data (l: File: colorectal_dataset.mat). The results are represented in Figures 6-10.

Figure 7 shows two plots – the number of clusters versus the Partition Coefficient (PC) and the number of clusters verses the Classification Entropy (CE). Both of these are Cluster Validity Indices that are used in Fuzzy Clustering methods to determine the number of optimal clusters for given datasets.

Figure 6: Fuzzy Clustering Using the Gustafson-Kessel Algorithm

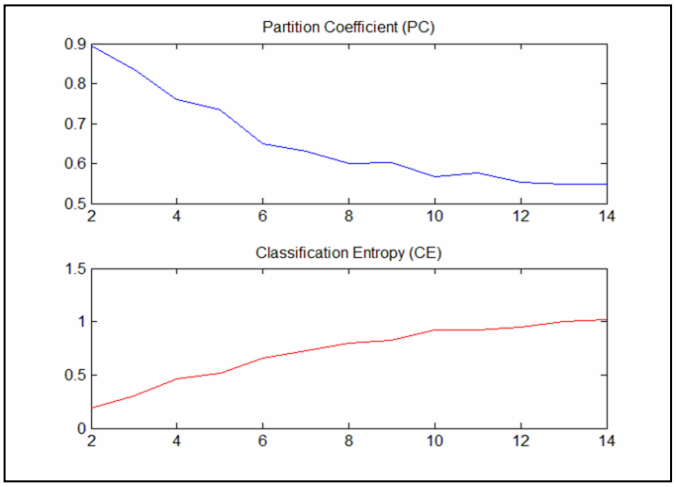


Figure 7: PC and CE (y-axis) vs. Number of Clusters (x-axis)

a) Partition Coefficient (PC) measures the amount of "overlapping" between clusters and is defined by Pal et al. (1995) as:

$$PC(c) = \frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}(u_{ij})^2$$

Where μ_{ij} denotes the membership (continuous value) of data value j in cluster i. The maximum value denotes the optimal number of clusters. PC has the disadvantage that it does not denote any direct relationship with the dataset properties.

b) Classification Entropy (CE), which is plotted in the second graph also measures the "fuzzyness" of the cluster partition. It is calculated using the following formula:

$$CE(c) = -\frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N} u_{ij}\log(u_{ij})$$

N.B.: The first local minimum value here gives the optimal number of clusters.

Figure 8 shows the following three indices calculated for the same clustering algorithm:

c) Partition Index (SC) - ratio of the sum of compactness and separation of the clusters.

$$SC(c) = \sum_{i=1}^{c}\frac{\sum_{j=1}^{N}(\mu_{ij})^m||x_j - v_i||^2}{N_i\sum_{k=1}^{c}||v_k - v_i||^2}$$

d) Separation Index (S) – validates the clusters depending on minimum separation distance and is computed using:

$$S(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^2||x_j - v_i||^2}{N min_{i,k}||v_k - v_i||^2}$$

e) Xie and Beni's Index (XB): a ratio of inter cluster variation and cluster separation:

$$XB(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^m ||x_j - v_i||^2}{N min_{i,j}||x_j - v_i||^2}$$

For the optimal number of clusters this index should be at a minimum value (which is at 5 clusters as shown in Figure 9).

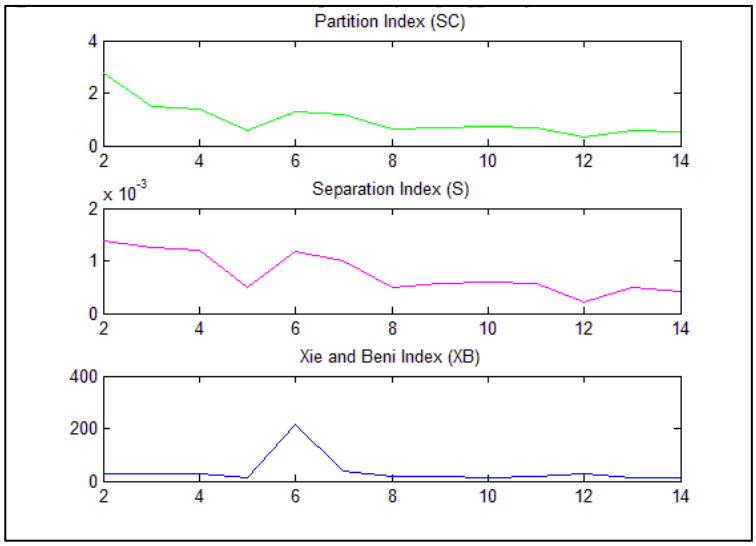


Figure 8: SC, S and XB Indices (y-axis) vs. Number of Clusters (x-axis)

Abonyi et al. (2004) noted that no validation index is reliable on its own. Thus, all the indexes must be calculated and then compared. They also noted that partitioning with fewer clusters is preferable when there are only minor differences between validation indices. Since both CE and PC exhibit monotonic change with an increasing number of clusters this suggests that both these indices have no connection to data attributes. Both methods indicate that the optimal number of clusters is 5. The remaining three indices i..e. SC, S and XB all show a local minimum at c=5.

This optimal number is also confirmed in Figure 9, which shows two more indices:

f) Dunn's Index (DI) – this is used to identify compact and well separated clusters. This is a computationally expensive index to calculate:

$$DI(c) = min_{i \in c}\{min_{j \in c, i \neq j}\{\frac{min_{x \in C_i, y \in C_j} d(x,y)}{max_{k \in c}\{max_{x,y \in C} d(x,y)\}}\}\}$$

g) Alternative Dunn's Index (ADI) – is a modified version of DI with the aim of simplifying the calculation.

Both DI and ADI show minimums at c=6. However, as the lower number of clusters (five) is preferable, this is the number of clusters that will be used for both K-means and hierarchical clustering.

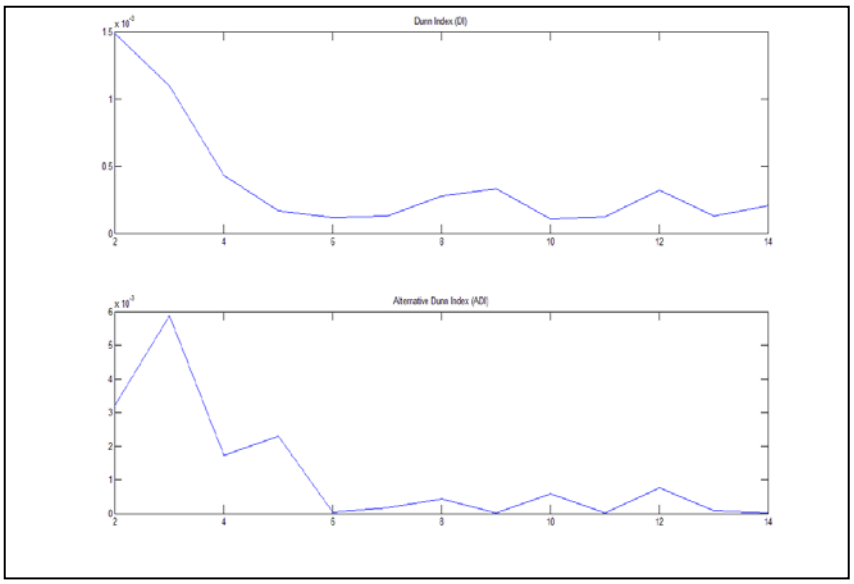


Figure 9: DI and ADI (y-axis) vs. Number of Clusters (x-axis).

N.B: The same functions were also applied to the reduced feature dataset (Ⅱ: File: colorectal_dataset_reduced.mat), with the same number of optimal clusters obtained. This is as expected since the filter methods were meant to remove genes that did not contribute to the process.

Hierarchical clustering was implemented using the 'cluster' function in Matlab with number of clusters set to 5. The x-axis shows the class which is either -1 (cancerous sample) or +1 (benign sample), whilst the y-axis shows the gene expression values. The results can be seen in Figure 10:

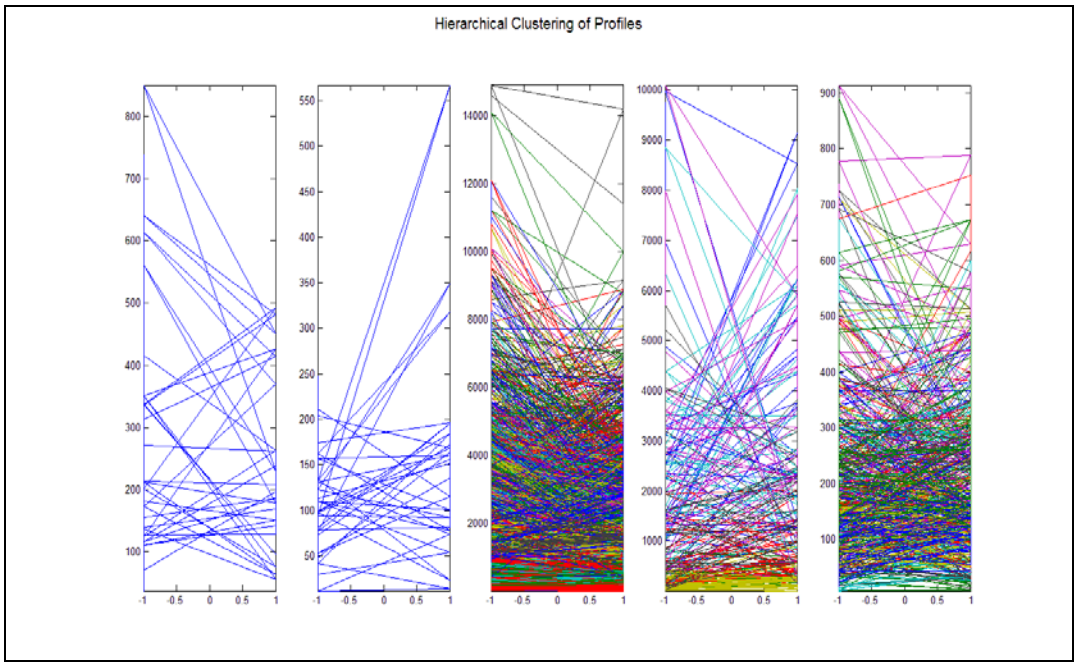


Figure 10: Hierarchical Clustering: Gene Expression (y-axis) vs. Class (x-axis) in 5 Clusters.

It is apparent from Figure 10 that there are fewer genes with lower expression values in clusters 1 and 2. Clusters 3, 4 and 5 on the other hand show a greater density of points – which include both positive and negative samples. Clusters 3 and 4 seem to have clustered genes which have far greater expression values than are found in the other groups.

A dendrogram was also drawn using the results of the hierarchical clustering (see Figure 11). Hierarchical clustering; unlike partition-based methods of cluster analysis, such as k-means, which begin with a set number of clusters; either merge

points or divide superclusters. The algorithm used here merges points into clusters. It is interesting to note that it was expected that the data would be classified into two clusters – one showing the cancerous tissues and the other the normal tissues. However, the clustering algorithms used are unsupervised learning methods, whereby the class data was not used by the learning algorithm. The resultant five clusters clearly show that there are complex underlying processes involving the data structures being studied. That is, the genes that show similarity may not always contribute to the process that leads to the development of cancer.

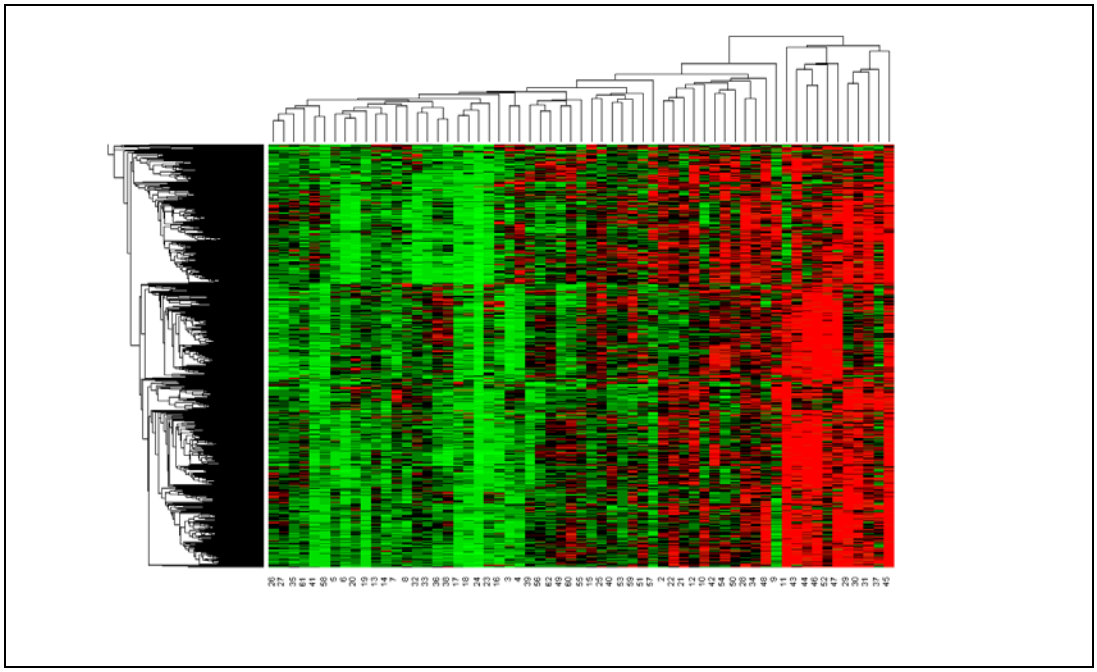


Figure 11: Hierarchical Clustering Dendrogram

K-means is one of the methods that ensures the distance between clusters is at a minimum. This function, which is a crisp/ hard clustering algorithm, was applied but with the number of clusters set to 5.

Results can be seen in Figure 12, which shows how the samples were placed by the algorithm. The x-axis shows the sample number whilst the y-axis indicates the distance of the point from the cluster center. It should be noted that all five clusters contain points that are far from the center of the cluster – possibly indicating outliers.

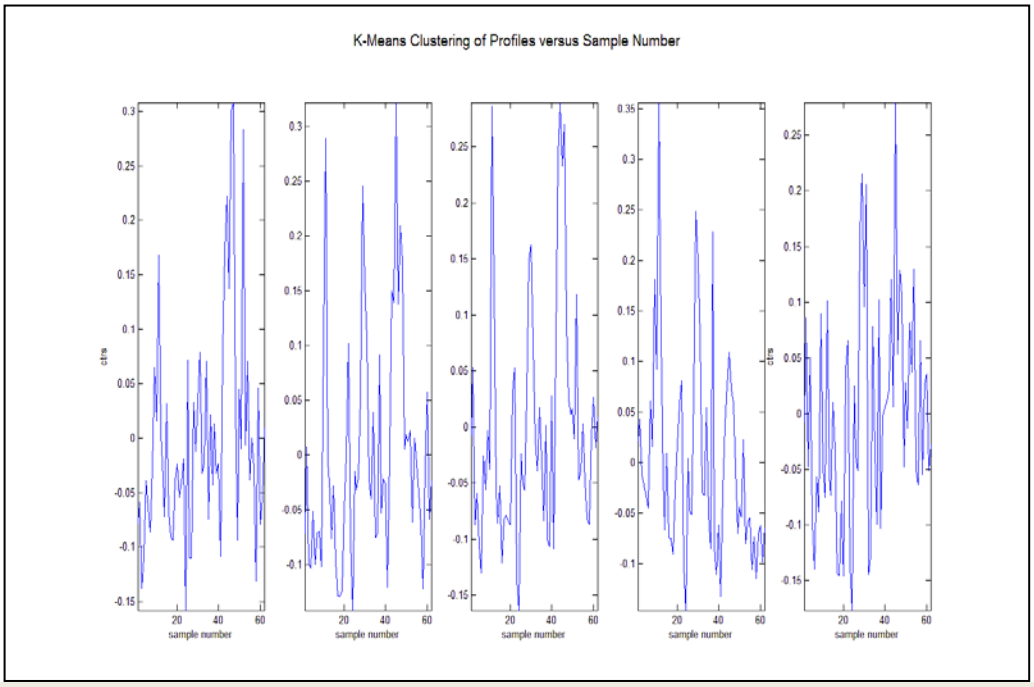


Figure 12: K-means Clustering Showing Five Clusters

To represent the samples placed in each cluster Figure 13 was obtained: the right axis shows positive (cancer: class variable = -1) and left axis shows negative (benign: class variable = +1) samples in each cluster.

It is clear that all the five clusters contain both positive and negative samples. That is the clustering has not partitioned the data into the two initial classes of positive and negative samples. This would seem to indicate that there are sub-classes within the dataset – possibly sub-clusters that contribute to the process but in a more complex way than initially assumed.

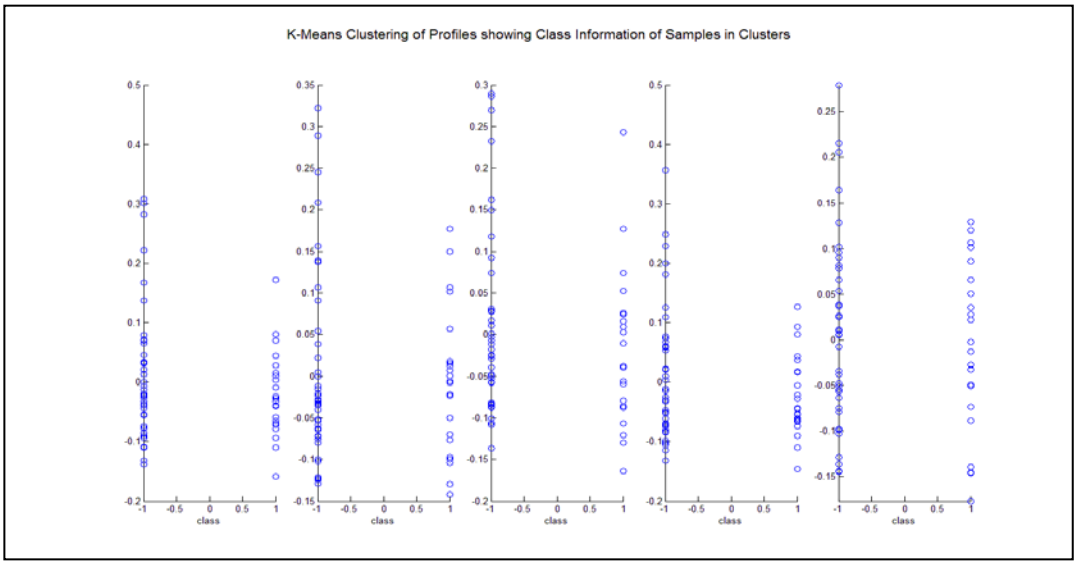


Figure 13: K-means Clusters With Class Information.

DIMENSIONALITY REDUCTION USING PCA

Microarrays have the problem that they display dimensionality, where the number of samples is exceeded by the number of attributes (Jiang et al., 2008; Tan et al., 2007; Li et al., 2006; Shen et al., 2005; Yoo at al., 2004). For any form of analysis some method of dimensionality reduction needs to be employed. There are a number of techniques that can be used for this purpose. One of the better-known methods for data reduction is Principal Components Analysis. As it is difficult for people to visualize data with greater than three dimensions, the idea behind PCA is that if there are groups of variables (attributes) that are behaving similarly, or have similar values, in any given situation, then this group can be replaced by a single variable. For simplification purposes this works well.

Initially the ‘mapcaplot’ function was used to create a plot of the principal components of the dataset (Figure 16). This figure shows a screenshot of the Matlab tool used to view principal component data.

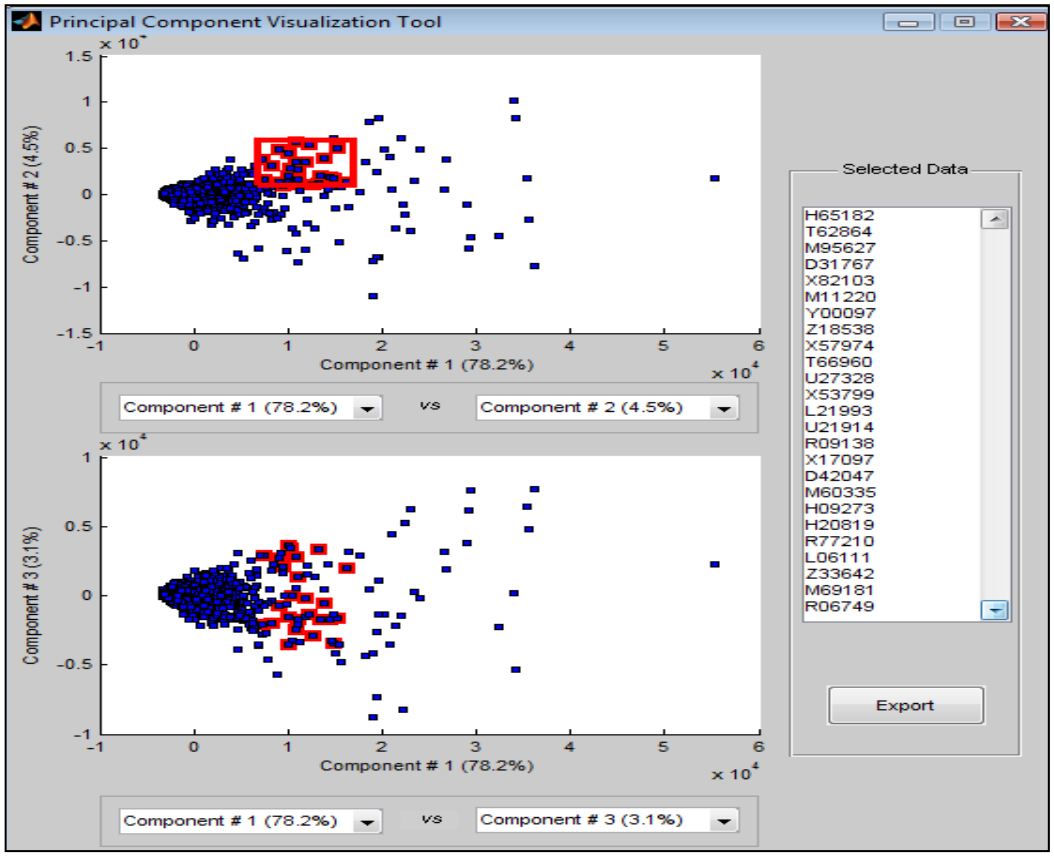


Figure 14: Scatter Plot Using PCA Visualisation Tool.

If data points are highlighted on one screen and corresponding points are also highlighted on the second screen, then the gene IDs will be displayed in the ‘Selected Data’ window. This is very useful for viewing multiple dimensions of the same dataset at the same time.

The function ‘pcvars’ gives a measure of the variance in the dataset. About 78% of the variance is contained in the first principal component. In fact, the first five

components account for about 90% of the variance, and the first two components account for 83% of data variance. The first two principal components have been represented on a scatter plot (Figure 15).

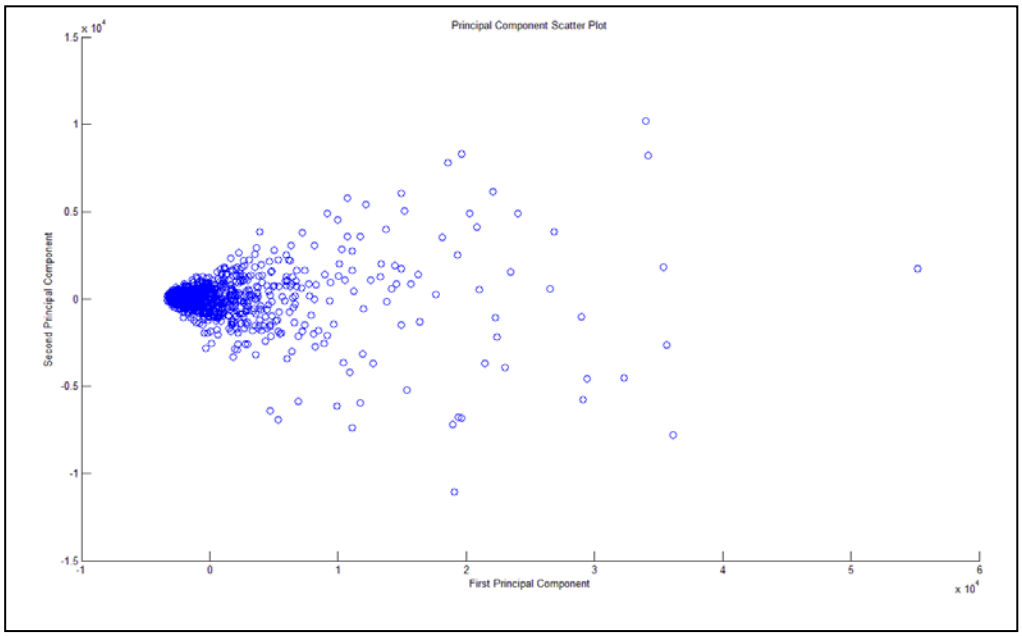


Figure 15: Scatter Plot Showing First Two Principal Components

A plot of the Eigen Values (Figure 16) shows that most of the original data can be captured using the first three principal components. In order to reduce the dimensionality of the dataset farther the first three principal components alone were saved in the file 'PCA_colon.csv'. These were then multiplied with the original data and the results were saved in the file labeled, 'colon_classification_data.mat'.

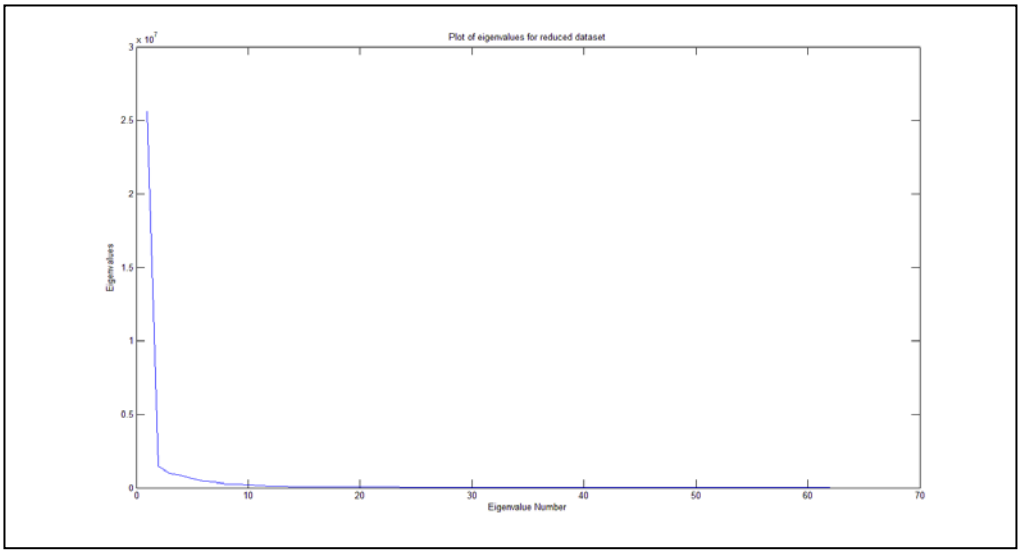


Figure 16: Plot of Eigen Values (y-axis) vs. Eigen Value Number (x-axis).

All files used for this analysis have been placed in the PCA folder; the Matlab codes are also placed in Appendix 2: Part B.

4.2 PART 1B: Initial Analysis of Normalized Data

Normalization.

Han & Kamber (2006 p. 114), describe normalization as being, "particularly useful for classification algorithms involving neural networks…nearest neighbor classification and clustering." Usefully, a normalized version of this dataset is readily available and has already been processed using the same methodology as described in PART 1A (above). Both datasets (normalized and un-normalized) will be used for classification in order to verify if indeed there is a difference in classification accuracy related to normalization.

I. FILE: colon_dataset_n.mat

Name	Size	Bytes	Class	Data
colon_class_n	1x62	496	double	+1 or -1
colon_genes_n	2000x1	144024	cell	Gene names
colon_genevalues_n	2000x62	992000	double	Expressions
colon_samples_n	1x62	496	double	Samples

FILTERING

When a filter for low variance was applied the number of genes reduced from 2000 in the initial dataset to 1799 in the latter. The last 15^{th} percentile of genes (those with the lowest activity in the dataset), were removed – leaving a final total of 1524 genes. The reduced dataset was placed in the file labeled, 'colorectal_dataset_reduced_n.mat'. The variables saved in the file were as follows:

II. FILE: colon_dataset_reduced_n.mat

Name	Size	Bytes	Class	Data
colon_genes_n_reduced	1524x1	110182	cell	Gene names
colon_class_n_reduced	1x62	496	double	Class Information
colon_samples_n_reduced	1x62	496	double	Sample Information
colon_genevalues_n_reduced	1524x62	758880	double	Expressions

CLUSTERING

The 'optnumber' function was used to determine the optimal number of clusters for the normalized dataset. In contrast to the dataset which had not been normalized,

marginally different results were obtained with the normalized dataset, whether filtered or unfiltered.

The function was initially run on the normalized, unfiltered data
(See Figures 17-20):

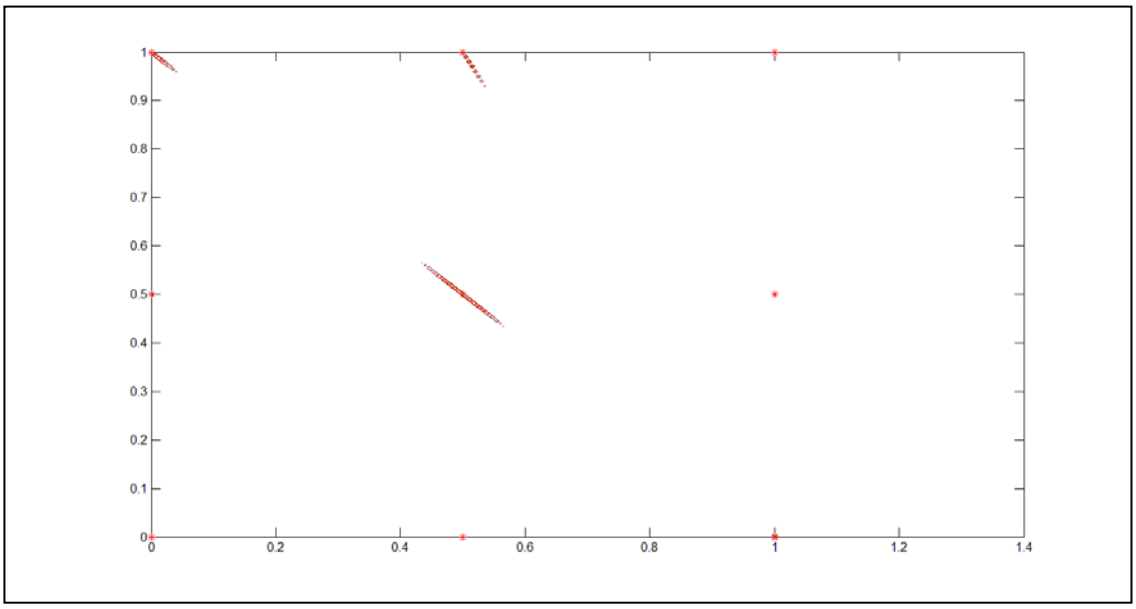


Figure 17: Fuzzy Clustering G-K for Normalised Data.

The Partition Coefficient (PC) and Classification Entropy (CE) can be seen in Figure 18.

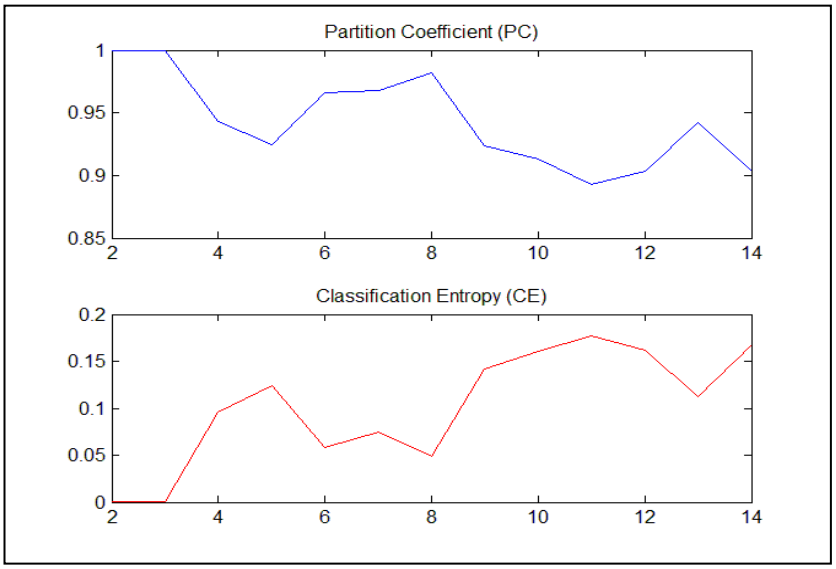


Figure 18: PC and CE for G-K Normalized Data

The first local maximum for PC occurs when the number of clusters is 6: the same is true for the first local minimum for CE.

Figure 19 shows the SC, S and XB validation indices:

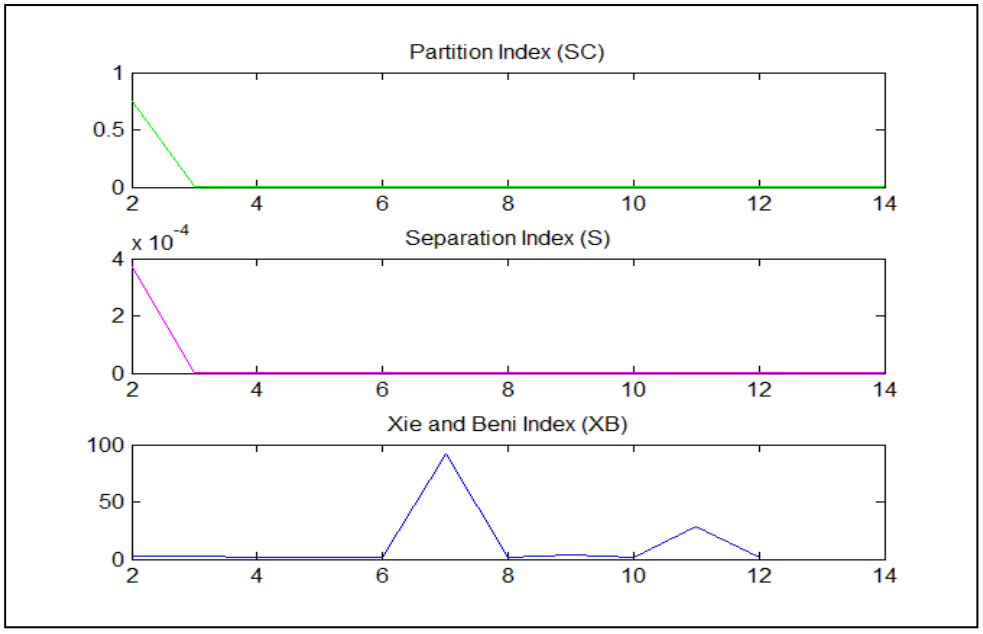


Figure 19: SC, S and XB Indices for Normalized Data.

The SC and S indices show their minimum points when the number of clusters is three, whilst XB seems to show a minimum at cluster number 4. Figure 20 shows the DI and ADI:

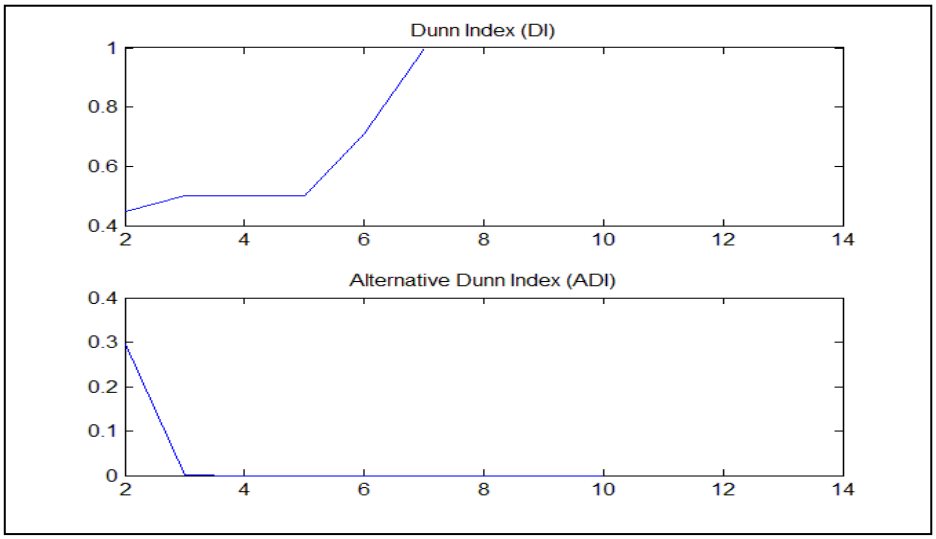


Figure 20: DI and ADI for Normalized Data.

Both of these indices indicate a local minimum value at cluster number three. Given that a lower number of clusters is the preferred outcome, the normalized version is considered to be optimally clustered when the cluster number is set to three. The complexity of the dataset has been clearly reduced as a result of normalization as can be seen in Figure 21.

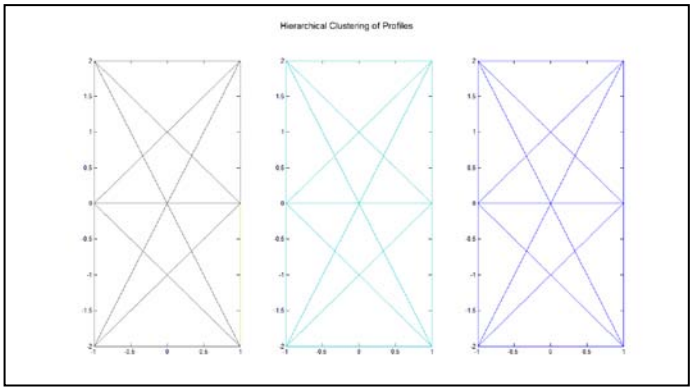


Figure 21: Hierarchical Clustering: Gene Expression (y-axis) vs. Class (x-axis) in 3 Clusters.

The results have also been plotted as a dendrogram (Figure 22).

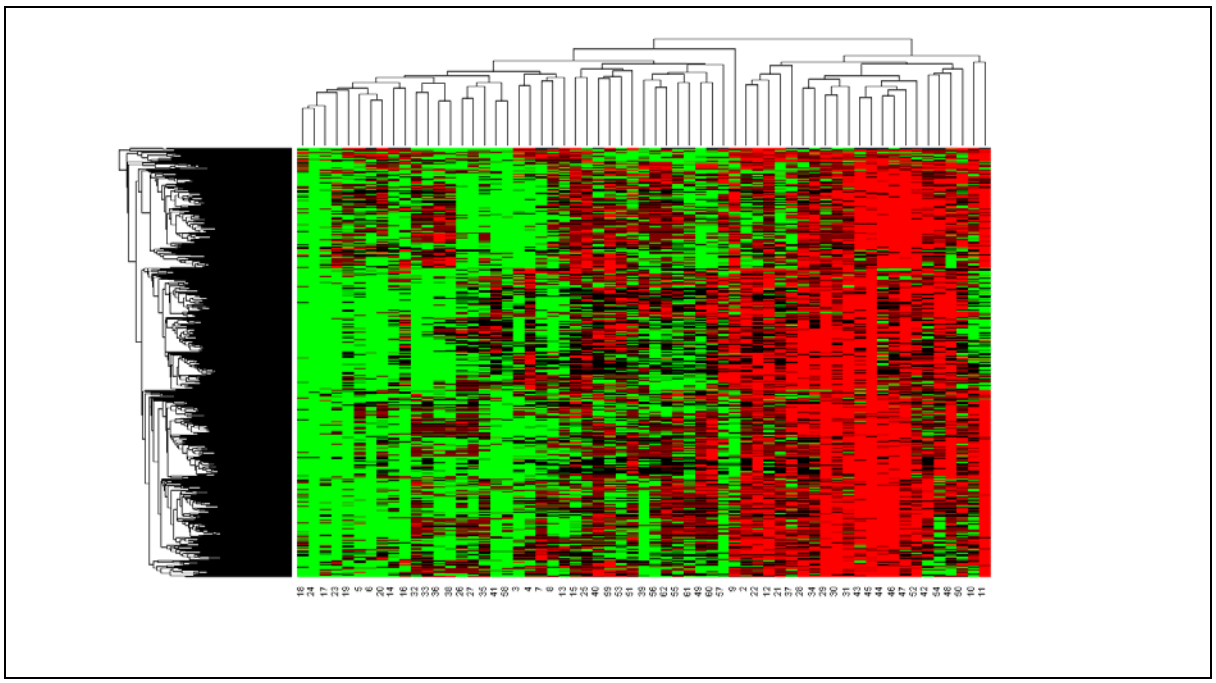


Figure 22: Dendrogram From Hierarchical Clustering of Normalized Data.

Figure 23 shows the results for K-means clustering run on the normalized dataset for three clusters:

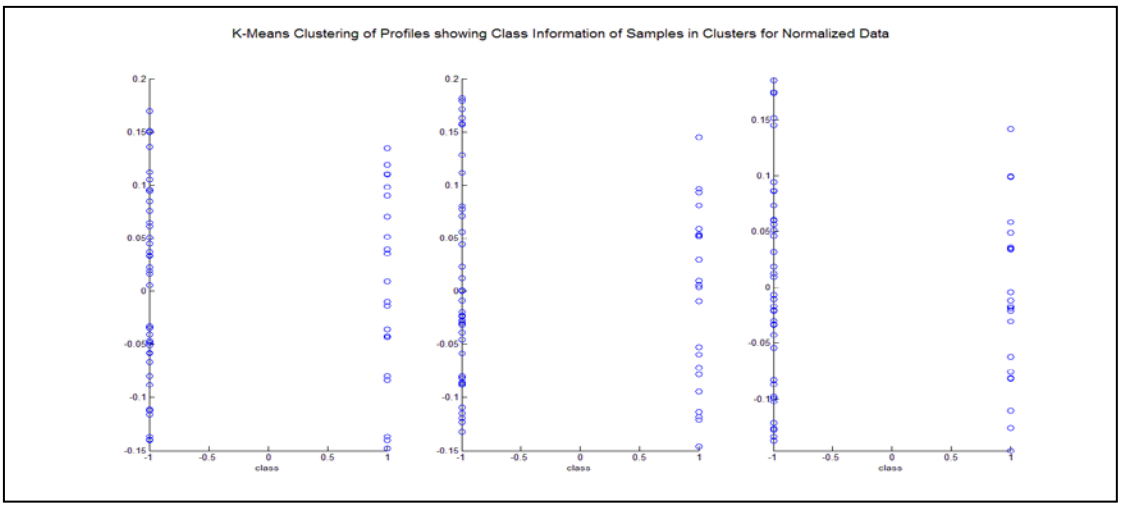


Figure 23: Class Data (x-axis) vs. Distance From Cluster Centre for Normalized Data.

It is obvious that the clustering process did not partition according to the two initial classes of cancerous (negative) or non-cancerous (positive) samples. A plot of sample number in clusters shows a similar trend.

PCA

The eigenvalue plot for this dataset (Figure 24) indicates that the first 10 vectors will represent the majority of the data:

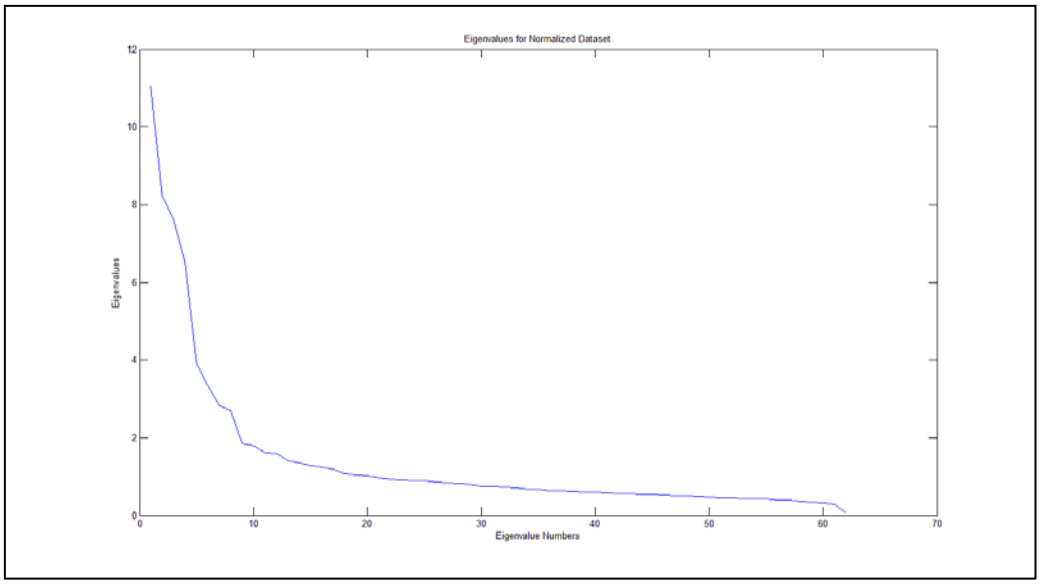


Figure 24: Eigen Values(y-axis) vs. Eigen Value Numbers (x-axis) for Normalized Data.

The first 10 components were saved in the file labeled 'PCA_colon_n.csv'. All relevant files have been placed in the folder 'PCA_n'. (Relevant code is placed in Appendix 2 – Parts A and B).

Figure 25 shows the first two principal components:

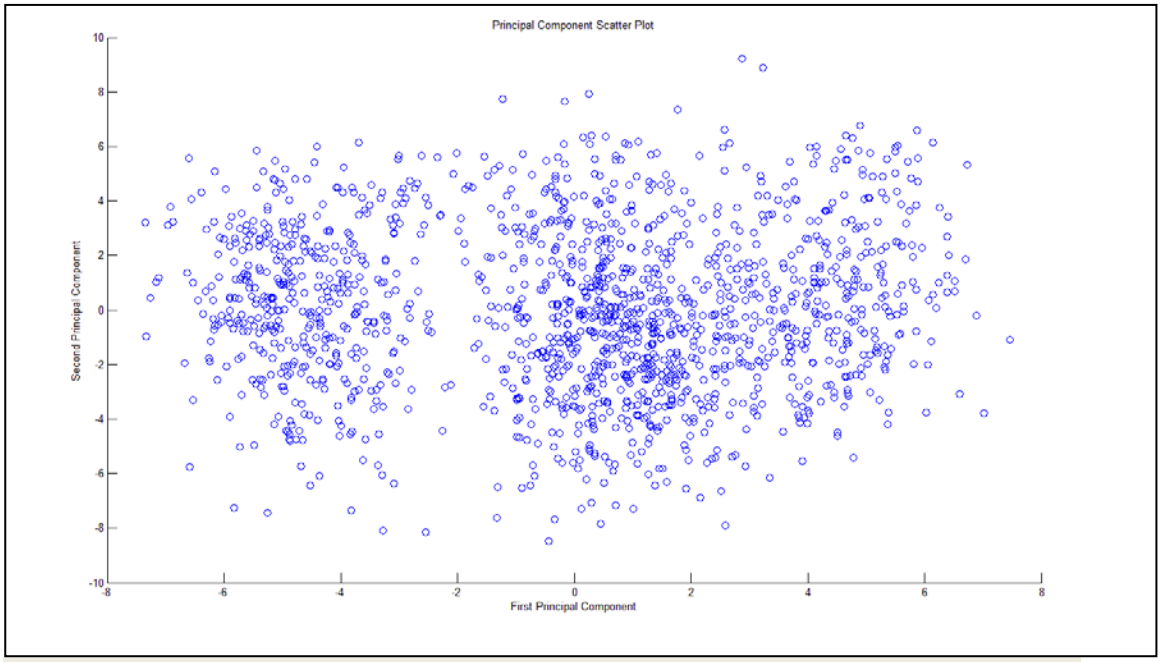


Figure 25: Plot of First Two Principal Components for Normalized Data.

Figure 25 indicates three clear clusters – although it is difficult to assess whether these clusters have partitioned data according to the negative/ positive class divide, since this information is not included in the plot. This analysis was run on the normalized, unfiltered dataset; and in order to see if there were any differences the same analysis was run with the normalized, filtered control version.

CLUSTERING

The optnum function was run with the normalized, filtered data (see Figures 26-29).

Both the PC and CE results indicate that five clusters is the optimal number for this dataset (Figure 26).

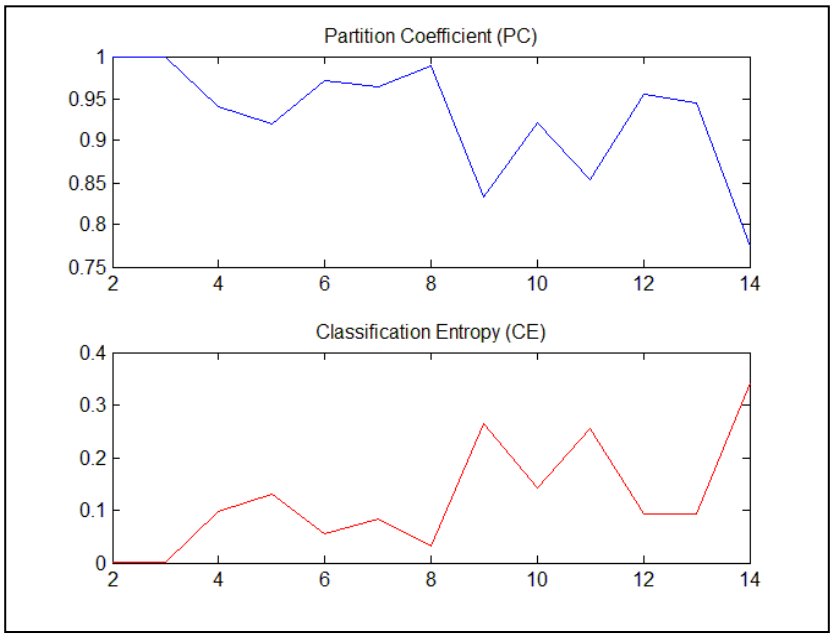


Figure 26: PC and CE for Normalized Data.

In Figure 27 the S and SC indices show a minimum at three clusters whilst the XB minimum is visible as 2 clusters.

Figure 30 shows the DI and ADI for the filtered, normalized dataset. Whilst the DI shows a minimum at two clusters, the ADI does so for three clusters. Thus, taking into account the optimal number given by all indices, the lower number is two and therefore this dataset can be clustered optimally using only two clusters.

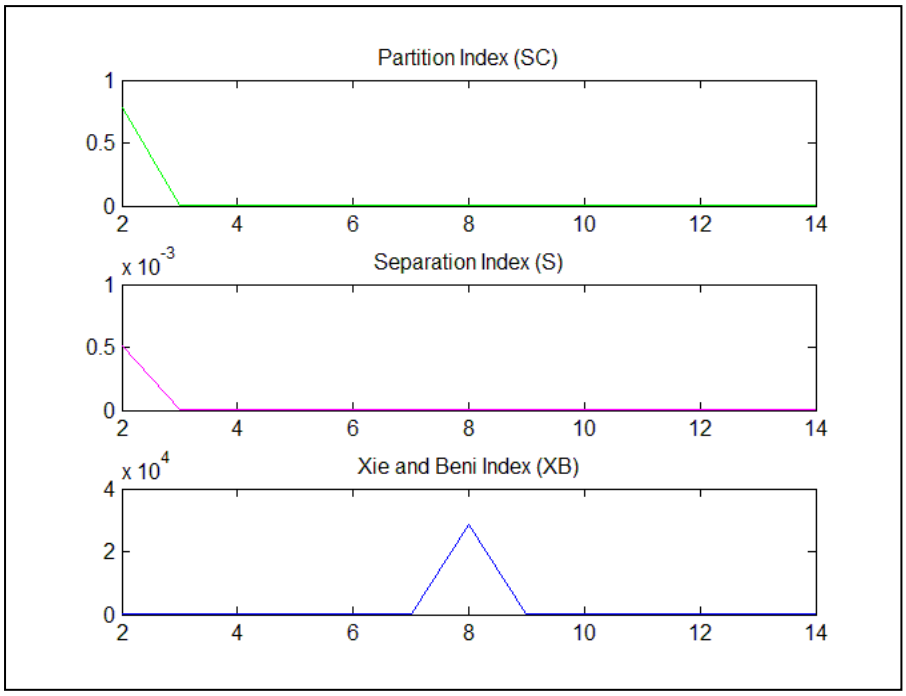


Figure 27: SC, S and XB for Normalized Filtered Data

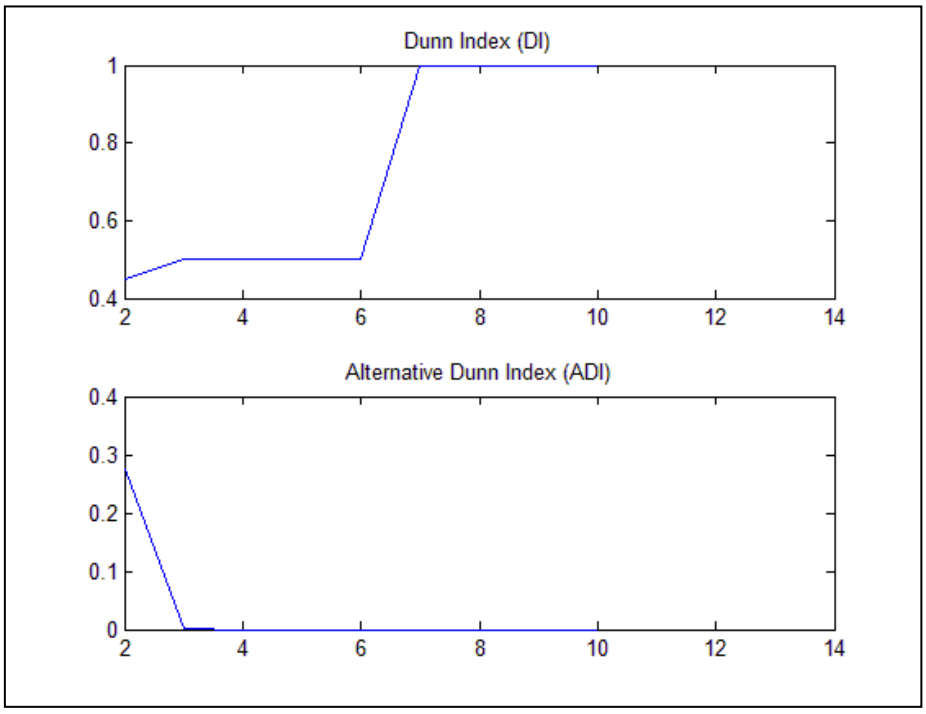


Figure 28: DI and ADI for Normalized Filtered Data.

The hierarchical clustering algorithm for this dataset produces a plot similar to Figure 21 but with two clusters only. The dendrogram is likewise similar to Figure 22 and both clusters contain positive and negative samples.

PCA

Figure 31 shows the eigenvalues for the normalized, filtered dataset. The first eight eigenvalues capture most of the variance in the data. Therefore, the first eight principal components were saved as 'PCA_colon_n_reduced.csv'. Additionally, the data to be used for the classification was stored in the 'colon_classification_data_n_reduced.mat' file.

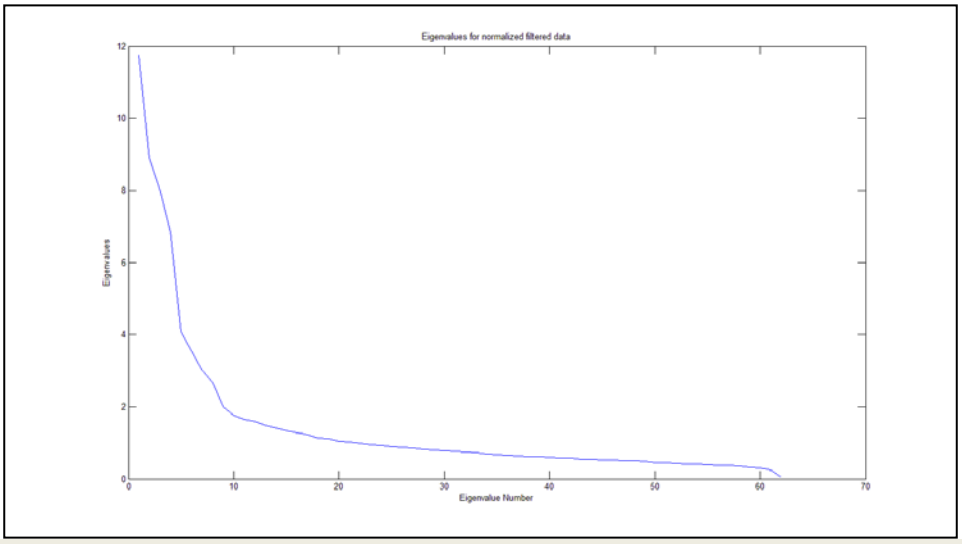


Figure 29: Eigen Values(y-axis) vs. Eigen Value Numbers (x-axis for Normalized, Filtered Data.

The next step was to use the three datasets (normalized, normalized and filtered, filtered) in the classification algorithms and compare the results.

4.3 PART 1C: Initial Analysis Using WEKA

Data Mining is the process of finding and describing structural patterns that can help to explain data and thus make useful predictions based upon these patterns. Occurrences in any given dataset are usually denoted by the values of features or attributes – in this case, the attributes are 2000 genes whose expression levels have been recorded for 62 samples or instances. The outcome shows whether any given sample is cancerous or not – indicated as negative (cancerous) and positive (benign).

WEKA has a limited capability when it comes to scaling and working with large datasets. Therefore, as part of the pre-processing phase heap size inside the package had to be increased in order for the data to be loaded into the WEKA application. Details of the data preparation procedure for WEKA have been placed in Appendix 2 – Part A.

The data (WEKA_Data.arff) is shown in Figure 30.

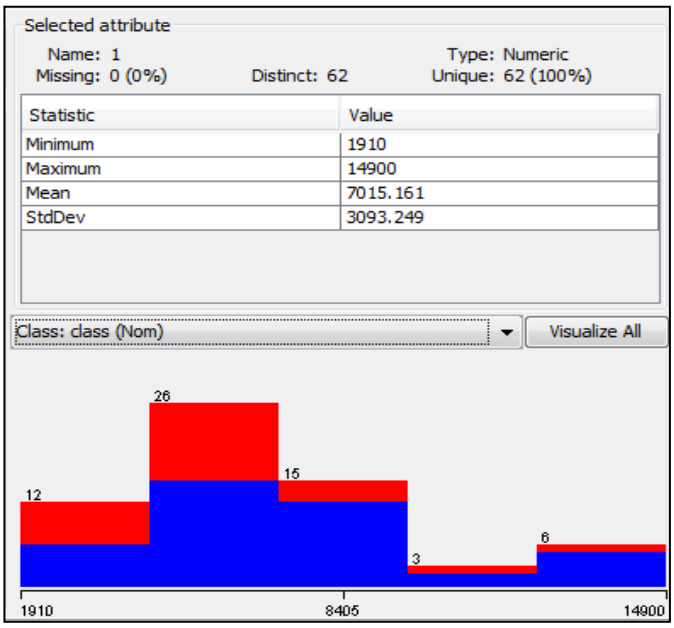


Figure 30: The WEKA_Data.arff file Loaded in WEKA.

This view shows all the attributes and the class distribution for the entire dataset – it represents a near normal distribution. The colors denote the two classes – positive and negative (blue for negative; red for positive). Some basic descriptive statistics are shown – highest gene expression value; the corresponding lowest value: the mean and also standard deviation. WEKA contains limited Visualization tools when compared to Matlab but as this has been explored elsewhere it will not be repeated.

DATA ANALYSIS, LEARNING METHODS AND RESULTS

For any form of data analysis to be carried out the dimensionality of the colon cancer dataset (2000 attributes) had to be reduced. According to (Hall et al., 2003, p. 1), "the success of many learning schemes...hinges on the reliable identification of a small set of highly predictive attributes…Regardless of whether a learner attempts to select attributes itself or ignores the issue, attribute selection prior to learning can be beneficial."

The Pre-processing module provided in WEKA contains both supervised and unsupervised methods. Supervised learning methods provide class data for attributes, while by way of contrast, unsupervised learning methods do not provide such class data – in fact, the number of classes is determined by the method. Therefore, in this case, because the class was already known (either negative or positive) the supervised learning methods were used.

The first supervised method used was 'AttributeSelection' from the Pre-process module provided by WEKA.

The AttributeSelection method has two options:

1) Evaluator – Determines how attributes/ attribute subsets are evaluated (18 options) and;
2) Search – Determines the search method (12 options).

Each of the options above represents different algorithms. To see if there was a difference in the number of attributes chosen by these options a comprehensive examination was conducted by running all available evaluators with all available search methods and the results were placed in the file labeled, 'AttributeSelection.xlsx' (Sheet = Attribute_Selection_1).

Certain methods resulted in lack of memory error messages (marked in blue), others ran slowly (marked in orange) and still others would not work with this particular dataset due to the design of the filter (see Table 2).

The size of the maxheap variable (RunWEKA.ini file) was changed from 128Mb to 512 Mb, and finally to 1012 MB to see if this would enhance the performance of the slow filter methods with no discernible change in the results.

The heap size was also increased in the JVM (version 1.6.0_13-b03) which created a problem due to limited RAM (2GB). This line of investigation was therefore curtailed, and it was thus decided that only this current depth of analysis could be aimed at for Attribute Selection methods in this instance.

Evaluation Methods	CfsSubsetEval	ChiSquaredAttributeEval	ClassifierSubsetEval	ConsistencySubsetEval	CostSensitiveAttributeEval	CostSensitiveSubsetEval	FileteredAttributeEval	FilteredSubsetEval	GainRatioAttributeEval	InfoGainAttributeEval	LatentSemanticAnalysis	OneRAttributeEval	PrincipalComponents	ReliefFAttributeEv	SVMAttributeEval	SymmetricalUncertAttributeEval	SymmetricalUncertAttributeSetEval	WrapperSubsetEval
Search Methods																		
Best First	27	/	1	/	/	/	/	27	/	/	/	/	/	/	/	/	/	1
Exhaustive Search	s	/	s	/	/	/	/	25	/	/	/	/	/	/	/	/	/	S
FCBF Search	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	16	/
Genetic Search	628	/	2	/	/	/	/	628	/	/	/	/	/	/	/	/	/	628
Greedy Stepwise	27	/	1	/	/	/	/	27	/	/	/	/	/	/	/	/	/	27
Linear Forward Selection	24	/	1	/	/	/	/	22	/	/	/	/	/	/	/	/	/	24
Race Search	/	/	b	/	/	/	/	/	/	/	/	/	/	/	/	/	/	S
Random Search	s	/	s	/	/	/	/	s	/	/	/	/	/	/	/	/	/	S
Ranker	/	2001	/	/	/	2001	2001	/	2001	2001	11	2001	b	2001	S	2001	/	/
Rank Search	48	/	2	/	/	/	/	48	/	/	/	/	/	/	/	/	/	2
Scatter SearchV1	b	/	b	b	/	/	/	b	/	/	/	/	/	/	/	/	/	S
SubsetSize Forward Selection		/	1	/	/	/	/	22	/	/	/	/	/	/	/	/	/	B

Table 2: Attribute Selection Methods in WEKA

In evaluating the performance of the various filters, only those filters that resulted in a number of attributes less than the original 2001 were placed in the file, AttributeSelection.xlsx (Sheet = Attribute_Selection_2). The results can also be seen in Table 3.

Evaluation Methods	CfsSubsetEval	ClassifierSubsetEval	FilteredSubsetEval	LatentSemanticAnalysis	SymmetricalUncertAttributeSetEval	WrapperSubsetEval
Search Methods						
BestFirst	27	1	27	/	/	1
ExhaustiveSearch	s	s	25	/	/	S
FCBFSearch	/	/	/	/	16	/
GeneticSearch	628	2	628	/	/	628
GreedyStepwise	27	1	27	/	/	27
LinearForwardSelection	24	1	22	/	/	24
RaceSearch	/	b	/	/	/	S
RandomSearch	s	s	s	/	/	S
Ranker	/	/	/	11	/	/
RankSearch	48	2	48	/	/	2
ScatterSearchV1	b	b	b	/	/	S
SubsetSizeForwardSelection		1	22	/	/	B

Table 3: Subset of Attribute Selection Methods in WEKA

It was neither possible, nor desirable, within the scope of this small-scale study to evaluate all the methods and methodologies available in order to determine whether the performance of the classifiers was influenced by the method used to select attributes. However, as this was, strictly speaking, not the ultimate goal of the study, it was decided to use the default method (marked in red in Table 3) for attribute selection.

This also had the advantage of being easily accessible and thus potentially replicable. That said, an assessment of all potential methods, in order to see which is the best for evaluating classification performance, would be an interesting further study in its own right.

THE DATASETS FOR CLASSIFICATION

In Parts 1A and 1B of this chapter there were three sets of data ready for evaluation using classifiers: the filtered, normalized, filtered and normalized datasets. To read these into WEKA they had to be converted into the .csv format. Details of these datasets have been placed in Appendix 2: Part A. Table 4 below shows a summary of the 7 different datasets that will be used with the classification algorithms in WEKA:

Dataset Name	Description
Data 1	Filtered but not Normalized Dataset – PCA Reduced to 62 x 3
Data 2	Normalized Dataset – PCA Reduced to 62 x 10
Data 3	Normalized and Filtered Dataset – PCA Reduced to 62 x 8
Data 4	Attribute Selection + PCA
Data 5	Discretization + PCA
Data 6	Attribute Selection + Discretization + PCA
Data 7	No Pre-Processing (Control)

Table 4: Summary Description of Datasets

RESULTS

The results for the first run of datasets and methods are shown in Table 4 (below). More detailed results tables are available in Appendix 2: Part C. The results revealed

very low classification accuracy for Data 1 (the highest being 69%). This was found to be due to an error in moving files from the .csv format from Matlab to the .arff format for Weka. The 'fixed' data results are placed in the column labeled 'Data 1_3 in Table 5 (below). Also seeAppendix 2: Part C.

All the methods shown in Table 5 used cross-validation as the performance matrix with the number of folds set to ten (McLachlan et al., 2004). PCA was applied to all datasets, except Dataset 7, before classification.

	Filtered	Data 1_3 (Data1 fixed).	Normalized	Filtered, Normalized	Attribute Selection	Discretizaton	Attribute Selection, Discretization	Unfiltered dataset
Classifier	**Data1**	**Data1_3**	**Data2**	**Data3**	**Data4**	**Data5**	**Data6**	**Data 7**
BayesNet	63.71	59.68	72.58	74.19	82.26	88.71	98.39	77.42
NaiveBayes	68.55	80.65	77.42	77.42	87.10	93.55	**100.00**	53.23
NaiveBayesSimple	68.55	80.65	72.58	75.81	85.48	95.16	**100.00**	NA
NaiveBayesUpdateable	68.55	80.65	74.19	77.42	87.10	93.55	**100.00**	53.23
RandomTree	67.34	70.97	64.52	69.35	74.19	62.90	70.97	67.74
REPTree	64.92	70.97	72.58	70.97	90.32	90.32	98.39	69.35
Logistic	69.35	82.26	83.87	83.87	77.42	82.26	93.55	74.19
Multilayer Perceptron	66.53	83.87	72.58	74.19	80.65	88.71	96.77	Slow
RBFNetwork	69.35	79.03	74.19	77.42	85.48	91.94	96.77	79.03
SimpleLogistic	68.95	80.65	83.87	79.03	87.10	83.87	96.77	77.42
SMO	64.11	75.81	85.48	88.71	87.10	91.94	96.77	85.48
Voted Perceptron	64.52	64.52	77.42	82.26	83.87	91.94	98.39	75.81
AdaBoostMI	65.73	79.03	72.58	75.81	83.87	95.16	96.77	74.19
Bagging	64.52	79.03	75.81	79.03	87.10	90.32	96.77	79.03
LogitBoost	67.34	69.35	72.58	72.58	88.71	91.94	96.77	75.81
MultiBoostAB	65.73	77.42	74.19	75.81	85.48	95.16	96.77	85.48
MultiClassClassifier	69.35	82.26	83.87	83.87	77.42	82.26	93.55	74.19
OrdinalClassClassifier	64.52	79.03	66.13	77.42	87.10	93.55	98.39	75.81
Random Committee	69.35	72.58	61.29	77.42	82.26	72.58	91.94	72.58

Table 5: Results of classification on all datasets.

The results in Table 4 show that, overall, Dataset 6, which was the dataset that had undergone the most pre-processing in WEKA yielded the best classification results. This is consistent with the available literature and consequent expectation.

100% classification results were also obtained with the Naïve Bayes, NaïveBayesSimple, and NaiveBayesUpdateable algorithms. All three of these (see Appendix 3: Part A) are versions of the same basic algorithm. Naïve Bayes is also one of the simplest classification algorithms used in Data Mining. Given the complex nature of the dataset it was surprising that the best classification results were achieved with this algorithm as opposed to a more complex one such as the MultiLayerPerceptron among others.

To evaluate the results Leave One Out Cross-Validation (LOOCV) was used. This is a special case in cross-validation where the number of folds is chosen to be n-1 where n is the total number of samples. In the case of this study n = 62 so the number of folds was chosen to be 61 and the classifiers were run again on the subset of results that had yielded a 100% classification.

The results of running LOOCV on Dataset 6 can be seen in Table 6.

	Classification Method	Correctly Classified Instances	Incorrectly Classified Instances	Confusion Matrix	
				a,b <-- classified as	
1	Naïve Bayes	100%	0%	40 0 | a = negative	0 22 | b = positive
2	Naïve Bayes Simple	100%	0%	41 0 | a = negative	1 22 | b = positive
3	Naïve Bayes Updateable	100%	0%	42 0 | a = negative	2 22 | b = positive

Table 6: Results of running LOOCV on Dataset 6.

As can be seen, Naïve Bayes performs very well – giving a hundred percent classification with very accurate results. There were 40 negative samples that were classified correctly, and 22 positive samples. There were no False Negative (FN) or False Positive (FP) results with this classifier. That said, variations of Naïve Bayes returned slightly anomalous results in their respective covariance matrices.

One sample was incorrectly classified by NaïveBayesSimple as negative (giving 41 negative samples as opposed to the original 40) and consequently there was one misclassification for the positive samples (22). A similar trend can also be observed in the results obtained from NaiveBayesUpdateable. In this case though there is an error of two samples i.e. two samples are classified as positive making the total 42 when in fact it should be 40. Therefore, both NaiveBayesSimple and NaiveBayesUpdateable are misclassifying samples but this error is not reported as a misclassification. This would seem to be a problem with the error estimate (in this case LOOCV).

LOOCV and Cross-Validation are very useful in cases where samples are sparse. LOOCV in fact uses the greatest amount of training data, thus increasing the accuracy of the classifier and also this procedure, unlike cross-validation, is not randomized but is deterministic. However, there is the disadvantage that this method cannot be stratified – in fact LOOCV guarantees a non-stratified sample.

N. B.: Stratification means that the correct proportion of examples of each class are placed into the test set and this is difficult when, as in the case of LOOCV, the test set contains only a single sample. Statistically the original dataset contained a bias as it held 40 samples of one class (negative or cancerous) as compared to only 22 samples of the second class(positive or benign). This may be one reason for the anomalous results shown in Table 5 above.

CHAPTER 5: IMPLICATIONS

The aim of this study was to build a classifier that accurately classifies microarray samples of both normal and tumor tissues for colon cancer. During the investigation of nineteen different classification algorithms it was found that Naïve Bayes returned 100% classification results that were also highly accurate.

This study also sought to find out whether normalization, filtering and discretization had any significant effect on the classifiers' performances. When the results from the control Dataset 7 (Table 13: Appendix 2: Part B) are compared to the normalized dataset Dataset 2 (Table 8: Appendix 2: Part B) it would seem that the accuracy of the classifiers is increased when normalization is applied. The misclassification of positive samples is not as significant as for negative samples, perhaps as positive samples are under-represented in the dataset.

Similarly, when the results of the filtered dataset Dataset 1_3 (Table 7: Appendix 2: Part B) and the control Dataset 7 (Table 13) are compared, it is clear that the accuracy and classification is much improved where the data has been filtered.

To evaluate whether discretization has had any effect on classification performance results from the control (Table 13) and Dataset 5 (Table 11) were also compared. It was once again clear that discretized data aids the performance of almost all classification methods. It is also clear that a combination of filtering, discretization and normalization yields the best results (see to Table 4 in Chapter 3 -above). This combination of methods was applied to Dataset 6 and the results are available as an appendix – Appendix 2: Part B: Table 12.

Most recent studies that have looked at classification of microarray data proposed complex algorithms that are better suited to non-linear class separation problems

such as Neural Networks (MLPs). Although this method also returns good result with Dataset 6 (96.77% correctly classified instances), and even when looking at the confusion matrix, it remains highly accurate, it still seemed unusual and a little surprising that a method as simple as that of Naïve Bayes returned the most accurate set of results. Since this is the case, it can be safely assumed that once the data has been pre-processed it is in fact linearly separable. Most datasets used in more recent studies contain greater dimensions than the colon cancer dataset used here – they often also contain a greater number of samples. This may account for this particular dataset being linearly separable rather than requiring a more complex method for class separation.

This dataset was first made publicly available almost a decade ago when microarray technology was still developing. There is also is a degree of sample contamination. The normal (i.e. non-cancerous) samples which were already under-represented in the dataset (22 out of a total of 62) contained some gene expressions specific to smooth muscle tissue. Thus, instead of just containing tissues from a specific organ the samples also contained some surrounding tissue. This may well have biased some of the classification methods. Ben-Dor et al. (2000), suggested that such bias is considered significant if less than 10 dimensions are used: with some of the datasets PCA reduced the number of dimensions down to lower numbers such as 8, or even 3.

However, it should be noted here that a single 10-fold cross-validation may not have been enough to obtain a reliable error estimate. It is well-known that cross-validation randomly partitions the data which means that there is a different result obtained with each run due to the effect of random variation in choosing the folds. To reduce this effect, cross-validation should be repeated 10 times and the results averaged. “When seeking an accurate error estimate…repeat the cross-validation process 10

times – that is, 10 times 10-fold cross-validation-and average the results", (Witten et al. 2005, p. 151). This procedure was not implemented due to sample size and other operational constraints, yet to fully verify all results this should perhaps have been the case.

All the above notwithstanding the purpose of the study was achieved in that a Naïve Bayes classifier was built that classified a filtered, discretized and PCA-reduced dataset with 100% accuracy. As above, cancer is fast becoming a major global disease and cancer treatment can be both financially prohibitive and distressingly invasive. Therefore, especially in developing nations, with fewer available medical resources, a comprehensive study looking at how genetic information can help to prevent cancer (and other diseases) could have potential far-reaching effects.

CHAPTER 6: CONCLUSIONS

The aims behind this study were largely realised. However, as with many such pieces of small-scale research, it was obviously that the scope of the study was limited and that there is much further research needed in this and other related areas/ fields of study. This could include applying similar techniques to benchmarked microarrays whose quality has been pre-established.

The Attribute Selection method used in WEKA (see Chapter 3) was only a single setting from amongst a larger number. Attribute Selection was one of the pre-processing procedures used in Dataset 6, which recorded the best classification results. This would imply that attribute selection is very important for classification purposes. Therefore, a more detailed study of the different Attribute Selection options available and how they affect the performance of the classifiers would be extremely informative.

Another direction worth exploring is the performance of non-linear learners such as Neural Networks, Support Vector Machines and Genetic Algorithms on this particular dataset.

Clustering Analysis was used in this study using fuzzy clustering algorithms which revealed, depending on the pre-processing of the data, cluster numbers that were greater than the expected result of two: given that there should have been only two classes present in the dataset. It would again be interesting to look at clustering in more detail to determine if the results obtained are biologically significant.

In this study genes were not isolated using their IDs – it was not determined which genes were removed during filtering, because this study was not looking at the biological significance of the results – rather the intention was to find the most accurate classifier for this dataset. However, this dataset has biological significance

and to make more sense of the results gene IDs could be mapped at all stages in the procedure. That said gene identities and their physiological importance remains, at present, the domain of medical and scientific experts.

Exploring if we can identify the exact gene sequence that can lead to colon cancer is, obviously, of great significance. The identification of the genes involved in such a highly complex process is not straightforward but could certainly prove very rewarding and important. If cancer sufferers are to experience the life-enhancing effects of early detection then microarray analysis might well prove a most useful tool, whether in developing countries with few resources to fight this disease, or in more developed nations such as the U.A.E ., where lifestyle issues (diet, tobacco consumption, etc.) often result in the virulent spread of this potentially fatal disease.

B I B L I O G R A P H Y

1. Abonyi J, Balasko B, and Feil B 2004, Fuzzy Clustering and Data Analysis Toolbox, 2004, available at <http://www.fmt.vein.hu/softcomp/fclusttoolbox/>

2. Altshuler D, Daly MJ, Lander ES 2008, **'**Genetic Mapping in Human Disease**',** *Science***,** vol. 322, no. 5903, pp. 881 – 888.

3. Almal AA, Mitra AP, Data RH, Lenehan PF, Fry DW, Cote RJ, Worzel WP 2006, 'Using Genetic Programming to Classify Node Positive Patients in Bladder Cancer', *Proceedings of the 8th annual conference on Genetic and evolutionary computation,* Seattle, Washington, USA, pp. 239 - 246. Available from ACM Portal.

4. Alon U, Notterman N, Gish DA, Ybarra K, Mack S, D & AJ Levine, 1999, 'Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays', *Proceedings of the National Academy of Sciences USA*, vol. 96, June 1999, pp. 6745-6750.

5. Blatt M, Wiseman S, Domany E 1997, 'Data Clustering Using a Model Granular Agent', *Neural Computation,* vol. 9, pp. 1805-1842.

6. Brazma A, Vilo J 2000, 'Gene expression data analysis', *FEBS Letters*, v.480, pp.17-24

7. Chandana S, Leung H, Trpkov K 2009, 'Staging of Prostate Cancer Using Automatic Feature Selection, Sampling and Dempster-Shafer Fusion ', *Cancer Informatics*, vol. 7, pp. 57-73.

8. Chang K-H, Kwon YK, Parker DS 2007, 'Finding Minimal Sets of Informative Genes in Microarray Data' in *Bioinformatics Research and Application*s, Springer, Berlin/Heidelberg, vol. 4463/2007, pp. 227-236. Available at: Springer Link.

9. Chung LWK, Isaacs WB & Simons JW 2007, 'Linkage Studies of Prostate Cancer Families to Identify Susceptibility Genes' in *Prostate Cancer,* Second Edition, pp. 285-299. Available from: Springer Link.

10.De Soto JA & Deng C-X 2006, 'PARP-1 inhibitors: are they the long-sought genetically specific drugs for BRCA1/2-associated breast cancers?', International Journal of Medical Sciences, vol. 3, no.4 , pp.117–123. Available from: PubMed Central.

11.Deegalla S. & Bostron H 2007, ' Classification of Microarrays with kNN: Comparison of Dimensionality Reduction Methods' in *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, Springer, Berlin/Heidelberg, vol. 4881/2007, pp. 800-809. Available at: Springer Link.

12.Drost DR , Novaes E , Boaventura-Novaes C , Benedict CI, Brown RS, Yin T, Tuskan GA, Kirst M 2009, 'A microarray-based genotyping and genetic mapping approach for highly heterozygous outcrossing species enables localization of a large fraction of the unassembled *Populus trichocarpa* genome sequence', *The Plant Journal*, vol. 58, no. 6, pp. 1054 – 1067.

13.Domany E, 2003 'Cluster Analysis of Gene Expression Data', *Journal of Statistical Physics*, vol. 110, no. 3-6, pp. 117-1139. Available from Springer Link.

14.Donoho D, Jon J 2008, 'Higher criticism thresholding: Optimal feature selection when useful features are rare and weak', *Proceedings of the National Academy of Sciences*, vol. 105 no. 39 14790-14795, Available at <http://www.pnas.org/content/105/39/14790.full>

15.Eisen MB, Brown PO, 1999, 'DNA Arrays for Analysis of Gene Expression', Methods Enzymol, vol. 303 (1999), pp. 179-205. Available from: <rana.lbl.gov/papers/Eisen_MEZ_1999.pdf>

16.Eisen MB, Spellman PT, Brown PO, Botstein D 1998, 'Cluster analysis and display of genome-wide expression patterns', *Proceedings of the Natural Academy of Science*, vol. 95, pp. 14863-14868.

17.Ferracin M, Gafà R, Miotto E, Veronese A, Pultrone C, Sabbioni S, Lanza G, Negrini M, 2008, 'The methylator phenotype in microsatellite stable colorectal cancers is characterized by a distinct gene expression profile', The Journal of Pathology, vol. 214, No. 5, Pages 594 – 602. Available from: Wiley InterScience Online Portal.

18.Frederiksen CM, Knudsen S, Laurberg S, Orntoft TF 2003, 'Classification of Dukes' B and C colorectal cancers using expression arrays', *Journal of Cancer Research Clinical Oncology*, vol. 129, pp.263-271.

19.Giordano TJ, Kuick R, Thomas DG, Misek DE, Vinco M, Sanders D,Zhu Z, Ciampi R, Roh M, Shedden K, Gauger P, Doherty G,Thompson NW, Hanash S, Koenig RJ, Nikiforov YE 2005 ,'Molecular classification of papillary thyroid carcinoma: distinct BRAF, RAS, and RET/PTC mutation-specific gene expression profiles discovered by DNA microarray analysis', *Oncogene*, vol. 24, pp.6646–6656.

20.Goldstraw P, Crowley J, Chansky K, Giroux DJ, Groome PA, Rami-Porta R, Postmus PE, Rusch V, Sobin L 2007, 'The IASLC Lung Cancer Staging Project: Proposals for the Revision of the TNM Stage Groupings in the Seventh Edition of the TNM Classification of Malignant Tumours' , *Journal of Thoracic Oncology*, vol. 2, no. 8., pp. 706-714.

21.Guo JK, Deng W, Zhang L, Li C, Wu P, Mao P 2007, 'Prediction of Prostate Cancer Using Hair Trace Elements Concentration and Support Vector Machine Method', *Biological Trace Elements Research*, vol. 116, pp. 257-271.

22.Guyon I, Weston J, Barnhill S, Vapnik V 2002, 'Gene Selection for Cancer Classification using Support Vector Machines', *Machine Learning*, vol.46, pp.389-422.

23.H Hong, L Jiuyong, Plank A, Wang H, Daggard G, 2006, 'A comparative study of classification methods for Microarray data analysis', *Proceedings of the fifth Australasian conference on Data mining and analytics*, vol. 61, pp. 33-37.

24.Hall MA, Holmes G 2003, 'Benchmarking Attribute Selection Techniques for Discrete Class Mining', *IEEE Transactions on Knowledge and data engineering*, vol. 15, no. 3, pp. 1-15.

25.Han J, 2009, Data Mining Concepts and Techniques, lecture notes used in University of Illinois at Urbana-Champaign on July 17, 2009. Chapter 2, Available at:http://www.cs.uiuc.edu/homes/hanj/bk2/slidesindex.html

26.Han J & Kamber M 2000, Data Mining Concepts and Techniques, Morgan Kaufmann.

27.Hand, D, Mannila, H & Smyth P 2001, *Principles of Data Mining*, MIT, Massachusetts.

28.Hernandez JCH, Duval B, Hao J-K 2007, 'A Genetic Embedded Approach for Gene Selection and Classification of Microarray Data', 5th European Conference on Evolutionary Computation, LNCS 4447, pp.90-101.

29.Hernandez JCH, Duval B, Hao J-K 2008, 'A Study of Crossover Operators for Gene Selection of Microarray Data' in *Artificial Evolution*, vol. 4926/2008, pp. 243-254. Available from Springer Link.

30.Huang T-M, Kecman V 2005, 'Gene Extraction for Cancer Diagnosis by Support Vector Machines - an Improvement and Comparison with Nearest Shrunken Centroid Method', *Proceedings of the Artificial Neural Networks: Biological Inspirations – ICANN 2005 15th International Conference*, Warsaw, Poland, September 11-15, 2005. Proceedings, Part I, Volume 3696/2005.

31.Kumar, Steinbach, Tan 2007, Introduction to Data Mining, Chapter 8 slides. Available at: < ke.cse.nsysu.edu.tw/~chlin/dm/tan/clust-tan-ch8.pdf>

32.Kumar L, Futschik M E 2007, 'Mfuzz: A software package for soft clustering of microarray data', Bioinformation, vol 2, no. 1, pp. 5-7.

33.Lausser L, Buchholz M, Kestler HA 2008, 'Boosting Threshold Classifiers for High-Dimensional Data in Functional Genomics' in *Artificial Neural Networks in Pattern Recognition*, vol. 5064/2008, pp. 147-156. Available from Springer Link.

34.Lee G, Rodriguez C, Madabhush A 2007, 'An Empirical Comparison of Dimensionality Reduction Methods for Classifying Gene and Protein Expression Datasets' in *Bioinformatics Research and Applications*, Springer, Berlin/Heidelberg, vol. 4463/2007, pp. 170-181. Available at: Springer Link.

35.Li S, Wu X, Hu X 2008, 'Gene selection using genetic algorithm, and support vector machines', *Soft Computing - A Fusion of Foundations , Methodologies and Applications*, vol. 12, no. 7, pp.693-698. Available from: SpringerLink [June 23 2009].

36.Llora X, Reddy R, Matesic B, Bhargava R 2007, 'Towards Better than Human Capability in Diagnosing Prostate Cancer Using Infrared

Spectroscopic Imaging', *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, London, England, pp. 2098 - 2105. Available from ACM Portal.

37.Lonning PE, Sorlie T, Perou CM, Brown PO, Botstein D, Borresen-Dale A-L 2001, 'Microarrays in primary breast cancer-lessons from chemotherapy studies', *Endocrine-Related Cancer*, v. 2001, no. 8, pp. 259-263.

38.Lu C, Devos A, Suykens JAK, Ar'us C, Van Huffel S 2002, 'Bagging linear sparse Bayesian Leaning Models for variable selection in cancer diagnoses', Journal of Latex Class files, vol. 1, no. 11, pp. 1-18.

39.Madeira SC, Oliveira AL 2004, 'Biclustering Algorithms for Biological Data Analysis: A Survey', *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 1, no.1, pp. 24-45.

40.Malekzadeh R, Bishehsari F, Mahdavinia M, Ansari R 2009, 'Epidemiology and molecular genetics of colorectal cancer in Iran: a review', Archive of Iran Medicine, vol. 12, no.2, pp.161-169. Available from: PubMed.

41.Marchiori E, Sebag M 2005, Bayesian Learning with Local Support Vector Machines for Cancer Classification with Gene Expression Data in *Applications on Evolutionary Computing*, Springer, Berlin/Heidelberg, Available at: Springer Link.

42.Meleth S, Chatla C, Katkoori VR, Anderson B, Hardin JM, Jhala NC, Bartolucci A, Grizzle WE, Manne U 2007, 'Comparison of Predicted Probabilities of Proportional Hazards Regression and Linear Discriminant Analysis Methods Using a Colorectal Cancer Molecular Biomarker Database', *Cancer Informatics*, vol. 2007, no.2, pp.125-132.

43.Mitchell, TM 1997, Machine Learning, McGraw-Hill, Singapore.\

44.Mukkamala S, Liu Q, Veeraghattam R, Sung AH 2006, 'Feature Selection and Ranking of Key Genes for Tumor Classification: Using Microarray Gene Expression Data', in *Artificial Intelligence and Soft Computing - ICAISC 2006*, Springer, Berlin/Heidelberg, vol. 4029/2006, pp. 951-961, Available at: Springer Link.

45.Mundra PA & Rajapakse JC 2007, 'SVM-RFE with Relevancy and Redundancy Criteria for Gene Selection' in Pattern Recognition in

Bioinformatics, Springer, Berlin/Heidelberg, vol. 4774/2007, pp. 242-252. Available at: Springer Link.

46. Nguyen H-N, Vu T-N, Ohn S-Y, Park Y-M, Han MY, Kim CW 2006, 'Feature Elimination Approach Based on Random Forest for Cancer Diagnosis' in *MICAI 2006: Advances in Artificial Intelligence*, Springer, Berlin/Heidelberg, vol. 4293/2006, pp. 532-542, Available at: Springer Link.

47. Pal, N.R. Bezdek, J.C 1995, 'On cluster validity for the fuzzy c-means model', IEEE Transactions on Fuzzy Systems, vol. 3, no. 3, pp. 370-379.

48. Pang S, Havukkala I. Hu Y, Kasabov N 2006, 'Classification consistency analysis for bootstrapping gene selection', *Neural Computing & Applications*, Vol. 16, No. 6. (October 2007), pp. 527-539.

49. Perou MC, Sorlie T, Eisen MB, Van de Rijn M, Jeffery SS, Rees CA, Pollack JR, Ross DT, Johnsen H, Akslen LA, Oystein F, Pergamenschikov A, Williams C, Zhu SX, Lonning PE, Borreson-Dale AL, Brown PO, Botstein D 2000, 'Molecular Portraits of human breast tumors', *Nature*, vol. 406, pp.747-752.

50. Piatetsky-Shapiro G, Khabaza T, Ramaswamy S, 2003, 'Capturing Best Practise for Microarray Gene Expression Data Analysis', *Proceedings of the ninth ACM SIGKDD international conference on discovery and data mining*, Washington D.C, pp. 407-415.

51. Polakis P 2007, 'The many ways of Wnt in cancer', Current Opinion in Genetics & Development, Vol. 17, no. 1, pp. 45-51. Available from: Science Direct.

52. Rao RB, Niculescu RS, Germond C, Rao H, 'Clinical and Financial Outcomes Analysis with Existing Hospital Patient Records', *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data minin*g, Washington DC, pp. 416-425. Available from: ACM Portal.

53. Rennert G 2009 a, 'Are We Getting Closer to Molecular Population Screening for Colorectal Cancer?', National Cancer Institute, 2009, Jun 17. [Epub ahead of print]Advance Access Oxford Journals.

54.Rennert G 2009 b,'Promising Biomarker and Candidate Tumor Suppressor Gene Identified for Colorectal Cancer', Memo to the Media, *Journal of the National Cancer Institute*, Advance Access published online on June 17, 2009. Available at: <http://jnci.oxfordjournals.org/cgi/content/full/djp226v1>

55.Salvado C & Cram D 2007, 'Microarray Technology for Mutation Analysis of Low-Template DNA Samples' in Single Cell Diagnostics, vol. 132, pp. 153-173. Available at: Springer Link.

56.Shen L & Tan EC 2005, 'Dimension Reduction-Based Penalized Logistic Regression for Cancer Classification Using Microarray Data', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol.2, no.2, pp.166-175.

57.Sorlie T, Perou CM, Tibshirani R, Aas T, Geisler S, Johnsen H, Hastie T, Eisen MB 2001, 'Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications', *Proceedings of the National Academy of Science*, vol. 98, no.19, pp.10869-10874.

58.Sorlie T, Tibshirani R. Parker J, Hastie T 2003, 'Repeated observation of breast tumor subtypes in independent gene expression data sets', *Proceedings of the National Academy of Science*, vol. 100, no.14, pp. 8418-8423.

59.Tan F, Fu X, Wang H, Zhang Y, Bourgeois A 2006, 'A Hybrid Feature Selection Approach for Microarray Gene Expression Data' in *Computational Science - ICCS 2006*, Springer, Berlin/Heidelberg, pp.678-685, Available at: Springer Link.

60.Takata R, Katagiri T, Kanehira M, Tsunoda T, Shuin T, Miki T, Namiki M, Kohri K, Matsushita Y, Fujioka T, Nakamura Y 2005, 'Predicting Response to Methotrexate, Vinblastine, Doxorubicin, and Cisplatin Neoadjuvant Chemotherapy for Bladder Cancers through Genome-Wide Gene Expression Profiling', Clinical Cancer Research, vol. 11, pp. 2625-2636. Available online: < http://clincancerres.aacrjournals.org/cgi/content/abstract/11/7/2625>

61.Tamura K, Furihata M, Tsunoda T, Ashida S, Takata R, Obara W, Yoshioka H, Daigo Y, Nasu Y, Kumon H, Konaka H, Namiki M, Tozawa K, Kohri K, Tanji N, Yokoyama M, Shimazui T, Akaza H, Mizutani Y, Miki T, Fujioka T, Shuin T, Nakamura Y, Nakagawa H 2007, **'**Molecular Features of

HormonenCells by Genome-Wide Gene Expression Profiles', *Cancer Research*, vol.67, pp. 5117. Available online: < <http://cancerres.aacrjournals.org/cgi/content/abstract/67/11/5117>

62. Tan Q, Thomassen M, Kruse TA 2007, 'Feature Selection for Predicting Tumor Metastases' in *Microarray Experiments using Paired Design*, *Cancer Informatics*, vol. 2007, no. 2, pp. 133-138.

63. Tang C, Zhang A 2002, 'An Iterative Strategy for Pattern Discovery in High-Dimensional Data Sets', *Proceedings of the eleventh international conference on Information and knowledge management*, pp. 10 - 17. Available from: ACM Portal [10 Feb 2009]

64. Wang L, Zhu J, Zou H 2007, 'Hybrid Huberized Support Vector Machines for Microarray Classification', *Proceedings of the 24th international conference on Machine learning*, Corvalis, Oregon, vol. 227, pp. 983 - 990. Available from ACM Portal. Wang H-Q & Huang D-S 2005, 'Non-linear cancer classification using a modified radial basis function classification algorithm', *Journal of Biomedical Science*, vol. 12, pp. 819-826.

65. Wang X & Gotah O 2009, 'Microarray-Based Cancer Prediction Using Soft Computing Approach', *Cancer Informatics*, vol. 7, pp. 123-139.

66. Wang X, Ghosh S, Guo SW 2001, 'Quantitative quality control in microarray image processing and data acquisition', *Nucleic Acids Research*, vol. 29, no.15 e75.

67. Winawer SJ 2007, Colorectal cancer screening, Best Practice & Research Clinical Gastroenterology, vol. 21, no. 6, pp. 1031-1048.

68. Winkler SM, Affenzeller M, Wagner S 2009, 'Using enhanced genetic programming techniques for evolving classifiers in the context of medical diagnosis', *Genetic Programming and Evolvable Machines*, vol. 10, no. 2, pp. 111-140.

69. Ye J, Li T, Xiong T, Janardan R 2004,' Using Uncorrelated Discriminant Analysis for Tissue Classification with Gene Expression Data', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol.1, no.4, pp181-190.

70. Xiong W, Zhang C, Zhou C, Liang Y 2006, 'Selection for Feature Gene Subset in Microarray Expression Profiles Based on a Hybrid Algorithm Using SVM and GA' in *Frontiers of High Performance Computing and Networking*, Springer, Berlin/Heidelberg, pp.637-647, Available at: Springer Link.

71. Xu R, Anagnostopoulos GC, Wunsch II DC 2007, 'Multiclass Cancer Classification Using Semisupervised Ellipsoid ARTMAP and Particle Swarm Optimization with Gene Expression Data', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 1, pp. 65-77.

72. Yoo S-H & Cho S-B 2004 ,'Optimal Gene Selection for Cancer Classification with Partial Correlation and k-Nearest Neighbor Classifier' in *Lecture Notes in Computer Science*, Springer, Berlin/Heidelberg, Available at: Springer Link.

73. You Z, Wang S, Gui J, Zhang S 2008, 'A Novel Hybrid Method of Gene Selection and Its Application on Tumor Classification' in *Advanced Intelligent Computing Theories and Applications With Aspects of Artificial Intelligence*, vol. 5227/2008, pp. 1055-1068. Available from Springer Link.

74. Yousef M, Jung S, Showe LC, Showe MK 2007, 'Recursive Cluster Elimination (RCE) for classification and feature selection from gene expression data', *BMC Bioinformatics 2007*, Vol. 8, no.144, p1-12. Available at: <http://www.biomedcentral.com/1471-2105/8/144/abstract> BioMed Central.

75. Yue F, Wang K, Zuo W 2007, 'Informative Gene Selection and Tumor Classification by Null Space LDA for Microarray Data' in *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*, Springer, Berlin/Heidelberg, vol. 4614/2007, pp. 435-446. Available at: Springer Link.

76. Zeng X-Q, Li G-Z, Wu G-F, Zou H-X 2007, 'On the Number of Partial Least Squares Components in Dimension Reduction for Tumor Classification' in *Emerging Technologies in Knowledge Discovery and Data Mining*, Springer, Berlin/Heidelberg, vol. 4819/2007, pp. 206-217. Available at: Springer Link.

77. Zhao Y, Chen Y, Zhang X 2007, 'A Novel Ensemble Approach for Cancer Data Classification' in *Advances in Neural Networks - ISNN 2007*, Springer,

Berlin/Heidelberg, vol. 4492/2007, pp. 1211-1220. Available at: Springer Link.

78. Zhou W, Zhou C, Zhu H, Liu G, Chang X 2006, 'Feature Selection for Microarray Data Analysis Using Mutual Information and Rough Set Theory', in *Artificial Intelligence Applications and Innovation*s, Springer, Berlin/Heidelberg, vol. 204/2006, pp. 492-499. Available at: Springer Link.

APPENDIX 1: EST & DATASETS

The ultimate goal of any investigation involving genes is not so much to find particular genes but more to find out how and why a gene in the DNA sequence is 'turned on'. This is also known as gene expression and is the first step towards understanding the role a given gene plays in coding for proteins.

This is, by no means, an easy task – the link between proteins and genes has usually been found in two ways:

1. Clinical studies may indicate a relationship – researchers then isolate the protein and determine its function and the genes that held the particular code for it
2. Researchers conduct linkage studies to identify the chromosomal location of genes and then biochemical methods are used to isolate that gene and the protein it hold the code for.

Both these methods are time consuming and only reveal small portions of the genes in the human DNA code.

However, in the last few years this process has been speeded up with the emergence of a technique that yields EST (Expressed Sequence Tags) for genes. An Expressed Sequence Tag is a tiny portion (200-500 nucleotides long) of an entire gene used to help identify unknown genes and to map their positions within a genome. [1] The Sequence pertains to either one or both ends of an expressed gene.

---

[1] http://www.ncbi.nlm.nih.gov/About/primer/est.html

The basic premise is that all living organisms contain DNA (genes) that belong to the same pool. So if an expressed gene is tagged in a certain organism, it can be used to identify the same gene in another organism by matching base pairs.

This is a complicated task because of the different genomic sizes of different organisms. This fact, and the existence of interruptions in DNA sequences known as introns, adds to the complexity of the task. In fact, gene identification is very difficult in the human genome because of the existence of a large number of introns.

Genes are expressed as proteins – a two step process:

1. Each gene is converted (transcribed) into messenger RNA (mRNA) which is a template for protein synthesis.
2. The mRNA then becomes a guide for the synthesis of a protein through a process called translation.

mRNA does not contain introns and therefore is the key to gene identification. However, it is unstable outside the cell – so through enzyme action the mRNA is converted into complementary DNA (cDNA).

cDNA is a much more stable compound and because it is generated from mRNA it represents only expressed DNA sequences (exons).

FROM cDNA to EST

Once cDNA from a given expressed gene is isolated either extremity of the molecule can be sequenced to give two kinds of EST:

1. 5' EST is the sequence for the beginning portion of the cDNA and is usually the code for a protein. This sequence is often conserved across species and is very similar in gene families (group of genes that code similar proteins).
2. 3' EST is the sequence that signals the end of the cDNA molecule and is often found in untranslated regions (UTR) – these are gene parts that do not translate into proteins. For this reason 3'ESTs are not usually seen across species.

ESTs: TOOLS FOR GENE MAPPING AND DISCOVERY

The human genome is made of billions of nucleotides – so to be able to navigate to given genes genome maps are necessary – these are constructed using Sequence Tagged Site (STS) mapping.

An STS is a short, unique DNA sequence - 3' ESTs are often used as an STS because they are likely to be unique to a given species and because they point directly to a particular expressed gene.

An EST is a copy of only the expressed part of the genome, which is why they are a very powerful tool in identifying genes that participate in a given process. This has been demonstrated in the hunt for gene sets that are involved in hereditary diseases such as Alzheimer's.

ESTs are generated rapidly and inexpensively and so there are practical reasons for this being a preferred method of gene identification. Scientists usually look for observable biological clues to first identify EST's that may participate in disease causing processes. Once this is done patient DNA is analysed to try and confirm the identity of genes that have mutated.

ESTs and NCBI

There has been a rapid proliferation of EST's in the last few years. In the beginning once an EST was generated it was submitted to GenBank (NIH sequence database). With multiple parties submitting EST identification became a problem. So, the dbEST was set up where tagged and identified ESTs were recorded. This database records data on human ESTs as well as over 300 other organisms. Records are annotated with DNA and mRNA text data.

There was a great deal of redundancy found in EST data because the same mRNA can be expressed by a given gene multiple times. The solution was to create UniGene – a database that automatically partitions the sequences into non-redundant clusters.

The use of ESTs is an efficient method for understanding processes within organisms – however there are limitations. mRNA is difficult to isolate from tissues which means that in some cases there is limited data available for some genes. This in turn affects result reliability. Also, important gene sequences may be found in introns. These introns are removed in order to obtain the ESTs – which means some valuable data may be lost.

APPENDIX 2: DATA PREPARATION, CODE AND RESULTS

PART A: DATA PREPARATION

NORMALISED DATA:

The software known as mRMR (Miniumum Redudancy Maximum Relevance) <http://penglab.janelia.org/proj/mRMR/index.htm> is a tool for variable selection. The dataset that is input should be discretized and the colon cancer dataset is in fact available in this format (discretized to three states) on the project website (saved as 'test_colon_mRMR_discretized.csv').

The authors mention the process they used, "We discretized the observations of each gene expression variable using the respective σ (standard deviation) and μ (mean) for this gene's samples: any data larger than $\mu + \sigma/2$ were transformed to state 1; any data between $\mu-\sigma/2$ and $\mu+\sigma/2$ were transformed to state 0; any data smaller than $\mu - \sigma/2$ were transformed to state -1. These three states correspond to the over - expression, baseline, and under-expression of genes."

When this dataset is uploaded into the online system the top ranking 50 genes are returned. The result is stored as 'mRMR_result.docx'.

MATLAB DATA PREPARATION

For calculating covariance the data had to be transposed and to avoid inadvertent deletion of any data during this copying and appending phase, all the data (tissues, names and gene) were collated into a single file. This data was stored as a new sheet (data_names_tissues) in the I2000_Excel_Fixed_with_Names.xls file.

The first row contains the sample number and the second row contains data indicating whether the sample was taken from a cancerous (negative) or non-cancerous (positive) tissue. The Matlab code associated with this procedure was also changed (data_2_read.m).

FILE: colorectal_dataset.mat, which contains the following variables:

Name	Size	Bytes	Class	Data
colon_class	1x62	496	double	+1 or -1
colon_genes	2000x1	144024	cell	Gene names
colon_genevalues	2000x62	992000	double	Expressions
colon_samples	1x62	496	double	Samples

The files used to read the data from Excel and text files were saved in the folder ‘colorectal_dataset’. All the files used for initial filtering were saved in folder ‘initial_analysis'. This folder also contains:

FILE: colorectal_dataset_reduced.mat, which contains variables:

Name	Size	Bytes	Class	Data
colon_genes_reduced	1530x1	110182	cell	Gene names
colon_genevalues_reduced	1530x62	758880	double	Expressions

FILE AND DATA MANIPULATION

The signed integers, used to denote class information (both positive and negative) were substituted with lexical items (i.e. word labels) before loading the data into the

WEKA application. In order to achieve this, a worksheet (Sample_Class_Word) was created inside the ‘tissue_transpose’ file.

It was observed that the text file I2000_Names contained double quotation marks at the end of certain lines. These irrelevant quotation marks were replaced by spaces to aid the data processing stage. Then the data was copied back into the Excel spreadsheet I2000_Names.xlsx, inside a worksheet labeled as ‘Raw_no_spaces’. To simplify the data all gene identifiers were amalgamated into a single column using the ‘Concatenate’ function in Excel. This single column was then placed in file, I2000_Names.xls, inside the worksheet ‘Raw_single_cloumn’.

A large amount of white space was attached to some names (which could be costly in terms of space and processing time). Thus, in order to see the location of these spaces, they were replaced by asterisks and the results saved in the I2000_Names_single_column.txt file. Next, the Excel function ‘Trim’ removed the spaces and the results were placed in the I2000_Names spreadsheet in the worksheet labeled ‘Final_Single_Column’. The same column was appended to Spreadsheet ‘I2000_Excel_without_blank_lines’ and data from the file, ‘tissue_transpose.xls’ was inserted as the last row.

All of this data was gathered in one single spreadsheet and saved as a Workbook labeled ‘WEKA_data’. The final step converted this data to a format that WEKA could work with - the ARFF (Attribute Relation File Format) format. This format defines the dataset but does not specify which of the attributes are to be predicted. For this reason the file can be used to investigate how well each attribute can be predicted, or for determining association rules or clustering.

To convert the data, it was then saved using the CSV (Comma Separated Value) format (Weka_Transpose.csv) and placed in WEKA’s working directory. The data

format was changed to general number – to remove scientific notation, which interfered with the conversion process from .csv to .arff.

To ensure that the genes were read as attributes, the data was transposed. However, the large number of attributes led to problems with WEKA hanging due to a lack of memory, meaning that the heap size had to be increased. The Simple Command Line Interface (CLI) included in the WEKA GUI was used to change the amount of memory allocated. The command used was:

java weka.core.SystemInfo

This returned a list of system properties:
memory.initial: 4.9MB (5177344)
memory.max: 127.1MB (133234688)

The maximum heap size was changed in the RunWeka.ini file using the variable 'maxheap' with an initial value of 128m (128MB later changed to 512mb (MB). The WEKA GUI Chooser then converted the .csv file to an .arff format file. In the Simple GUI console the following command was typed:

java weka.core.converters.CSVLoader Weka_Transpose.csv > Weka_Data.arff

This read the Weka_Transpose file, converted it to a .arff format and saved it as a file in the WEKA Directory. The contents of the .arff file were also saved as a text file:

- The top-level relation in this file is:

@relation Weka_Transpose

- This file contained 2000 attributes such as:

@attribute 1 numeric

@attribute 2 numeric...and so on for the 2000 attributes.

- The last attribute was:

@attribute class {'negative ',positive}

- The data was represented using the:

@data relation

Note:

- to convert .csv files to the .arff format the following command was used in the CLI in WEKA:

java weka.core.converters.CSVLoader PCA_filtered_set1_2.csv > Weka_Data_2_2.csv

- to convert .arff files to the .csv format the following would be used in the WEKA CLI:

java weka.core.converters.CSVSaver -i Weka_Data.arff -o Weka_Data.csv

In both instances the files being converted must be placed in the working directory. This is also where the output file will be saved.

DATASET 1

The filtered (but not normalized) data was saved in the file labelled, 'colon_classification_data.mat'. This file contained the following variables:

Variable	Dimensions	Data
PCA	62 x 3	un-normalized dataset
colon_class	1x62	Class information
colon_samples	1x62	Sample numbers

The files were saved as .csv files and then the colon_class was transposed and placed inside the PCA variable. This was then saved as one file: 'PCA_filtered_set1.csv'. The numeric values were changed from scientific to general. Also a header row was added and the class variable saved with the lexical labels 'negative' and 'positive'. Once this was done the dataset was saved as 'Matlab_Data1.arff'.

DATASET 2

The second dataset was the normalized dataset, which had been processed in Matlab. The variables were saved in the file labelled, 'colon_classification_data_n.mat' as follows:

Variable	Dimensions	Data
PCA_n	62 x 10	normalized dataset
colon_class_n	1x62	Class information
colon_samples_n	1x62	Sample numbers

These variables were then saved as .csv files; and the class data was appended to the PCA data before the results were saved in the file PCA_normalized_set2.csv. This was finally converted and saved as 'Matlab_Data2.arff'

DATASET 3

The third dataset, also processed in Matlab, was the normalized, filtered dataset. The variables were saved in the file labelled, 'colon_classification_data_n_reduced.mat' as follows:

Variable	Dimensions	Data
PCA_n_reduced	62 x 8	Normalized, filtered dataset
colon_class_reduced	1x62	Class information
colon_samples_reduced	1x62	Sample numbers

These variables were then written to .csv files, and the class data was appended to the PCA data before the results were saved in the PCA_normalizedfiltered_set3.csv file. This was finally converted and saved as 'Matlab_Data3.arff' for use in WEKA.

DATASET 4

This was the original dataset that was pre-processed using Attribute Selection in WEKA followed by PCA, where the data was normalized and 95% of the variance was captured. This data was saved as 'WEKA_Data4.arff'.

DATASET 5

In this dataset the data were first discretized and then PCA (with normalization) was applied. The resulting data was saved as **'**WEKA_Data5.arff'

DATASET 6

Attribute Selection was applied to the original dataset followed by discretization and then PCA with normalization. The results were saved in the file 'WEKA_Data6.arff'.

DATASET 7

This was the raw dataset that had not been normalized, filtered nor discretized and was saved as 'WEKA_Data.arff'.

PART B: CODE

PYTHON CODE:

(Python_remove_blank_lines.py)

```
#open file and read contents into string variable.
fd = open("Raw_Text.txt")
file = fd.readlines()
fd.close()

new_file = []

#remove blank lines
for line in file:
    # Strip whitespace, should leave nothing if empty line was just "\n"
    if not line.strip():
        continue
    # We got something, save it
    else:
        new_file.append(line)

# Print file sans empty lines
```

```
#print "".join(new_file)

p = "".join(new_file)
#send output to a new file.

#f=open('Raw_Text.txt', 'w')
```

MATLAB CODE:

To read the data into Matlab the file ‘data_1_read.m’ was written which read the excel data and placed it in an array variable labeled Num. The text column containing gene identification data was separated into a new array by Matlab. The M-file output includes a 2-dimensional graphic representation of the array variable Num:

data_1_read.m:

```
clear
num = xlsread('I2000_Excel_Fixed_with_Names.xls', 'data_1');
size(num)
plot(num)
```

data_2_read.m:

```
clear
num = xlsread('I2000_Excel_Fixed_with_Names.xls', 'data_names_tissues');
size(num)
plot(num)
```

colon_microarray_creation.m

```
clear
colon_genevalues = xlsread('Cell_Matrix.xlsx', 'only_values');
colon_class = transpose(xlsread('Cell_Matrix.xlsx', 'class'));
[colon_genes] = textread('Cell_Matrix.txt','%s');

save('colorectal_dataset', 'colon_genes', 'colon_genevalues', 'colon_class')
```

microarray_creation_n.m

```
clear
```

```
colon_genevalues_n = transpose(xlsread('test_colon_mRMR_discretized.csv',
'test_colon_mRMR_discretized'));
%colon_class_n = xlsread('test_colon_mRMR_discretized.csv', 'class');
[colon_genes_n] = textread('Cell_Matrix.txt','%s');
colon_class_n = transpose(xlsread('Cell_Matrix.xlsx', 'class'));

save('colorectal_dataset_n', 'colon_genes_n', 'colon_genevalues_n', 'colon_class_n');
```

colorectal_reduce.m

```
%original data is loaded
load colorectal_dataset.mat
colon_genes=transpose(colon_genes);
% Filtering the Genes
mask = genevarfilter(colon_genevalues);
colon_genevalues = colon_genevalues(mask,:);
colon_genes = colon_genes(mask);

% remove low entropy profile genes
[mask, colon_genevalues, colon_genes] =
geneentropyfilter(colon_genevalues,colon_genes,...
                                        'prctile',15);
save('colorectal_dataset_reduced', 'colon_genes', 'colon_genevalues')
%new file saved with reduced dimensions.

%Hierarchical clustering
corrDist = pdist(colon_genevalues, 'corr');
clusterTree = linkage(corrDist, 'average');
clusters = cluster(clusterTree, 'maxclust', 5);

figure
for c = 1:5
   subplot(1,5,c);
   plot(colon_class,colon_genevalues((clusters == c),:)');
   axis tight
end
suptitle('Hierarchical Clustering of Profiles');

% Dendrogram from the output of the hierarchical clustering.
clustergram(colon_genevalues(:,2:end),'RowLabels',colon_genes,...
                        'ColumnLabels', colon_samples(2:end))
```

```
%  K-means clustering function.
rand ('twister', 0);
[cidx, ctrs] = kmeans(colon_genevalues, 5, 'dist','corr', 'rep',5,...
                                        'disp','final');

% plot just the centroids.
figure
for c = 1:5
   subplot(1,5,c);
   plot(colon_class,ctrs(c,:)');
   axis tight
   axis off    % turn off the axis
end
suptitle('K-Means Clustering of Profiles with class information');
```

colorectal_reduce_n.m

```
%hierarchical clustering for noramlized filtered data
load colon_dataset_n_reduced


%Hierarchical clustering
corrDist = pdist(colon_genevalues_n_reduced, 'corr');
clusterTree = linkage(corrDist, 'average');
clusters = cluster(clusterTree, 'maxclust', 2);

figure
for c = 1:2
   subplot(1,2,c);
   plot(colon_class_n_reduced,colon_genevalues_n_reduced((clusters == c),:)');
   axis tight
end
suptitle('Hierarchical Clustering of Profiles');

% Dendrogram from the output of the hierarchical clustering.
clustergram(colon_genevalues_n_reduced(:,2:end),'RowLabels',colon_genes_n_re
duced,...
                        'ColumnLabels', colon_samples_n_reduced(2:end))
```

kmeans_n.m

```
load colorectal_dataset_n.mat


%  K-means clustering function.


rand ('twister', 0);
[cidx, ctrs] = kmeans(colon_genevalues_n, 3, 'dist','corr', 'rep',5,...
                                          'disp','final');

%cidx contains cluster number 1-3
%ctrs contain cluster centers

%Clusters showing sample numbers.
figure
for c = 1:3
   subplot(1,3,c)
   plot(colon_samples_n,ctrs(c,:)');
   xlabel('sample number')
   axis tight
end
suptitle('K-Means Clustering of Profiles versus Sample Number for Normalized
Data');

figure
for c = 1:3
   subplot(1,3,c);
   scatter(colon_class_n, ctrs(c,:));
   xlabel('class');
end;
   axis tight
suptitle('K-Means Clustering of Profiles showing Class Information of Samples in
Clusters for Normalized Data');
```

optnumber.m

```
load colorectal_dataset_reduced.mat
data.X=colon_genevalues(:,[1 2]);
[N,n]=size(data.X);

%data normalizaiton
data = clust_normalize(data,'range');
```

```
%parameters
ncmax=14; %maximal number of cluster
param.m=2;
param.e=1e-3;
%
ment=[];
figure(1)
for cln=2:ncmax
param.c=cln;
   param.ro = ones(1,param.c);
   result=GKclust(data,param);
   clf
   plot(data.X(:,1),data.X(:,2),'b.',result.cluster.v(:,1),result.cluster.v(:,2),'r*');
   hold on
   new.X=data.X;
   clusteval(new,result,param)
   %validation
   result=modvalidity(result,data,param);
   ment{cln}=result.validity;

end


PC=[];CE=[];SC=[];S=[];XB=[];DI=[];ADI=[];

   for i=2:ncmax
      PC=[PC ment{i}.PC];
      CE=[CE ment{i}.CE];
      SC=[SC ment{i}.SC];
      S=[S ment{i}.S];
      XB=[XB ment{i}.XB];
      DI=[DI ment{i}.DI];
      ADI=[ADI ment{i}.ADI];
  end
   figure(2)
   clf
   subplot(2,1,1), plot([2:ncmax],PC)
   title('Partition Coefficient (PC)'
   subplot(2,1,2), plot([2:ncmax],CE,'r')
   title('Classification Entropy (CE)')
   figure(3)
```

```
subplot(3,1,1), plot([2:ncmax],SC,'g')
title('Partition Index (SC)')
subplot(3,1,2), plot([2:ncmax],S,'m')
title('Separation Index (S)')
subplot(3,1,3), plot([2:ncmax],XB)
title('Xie and Beni Index (XB)')
figure(4)
subplot(2,1,1), plot([2:ncmax],DI)
title('Dunn Index (DI)')
subplot(2,1,2), plot([2:ncmax],ADI)
title('Alternative Dunn Index (ADI)')
```

PCA_colon.m

```
%Use reduced dataset
load colorectal_dataset_reduced.mat

% Principal Component Analysis
mapcaplot(colon_genevalues, colon_genes)

%pc = matrix of the principal components
% zscores, are the principal component scores
%pcvars, contains the principal component variances.

[pc, zscores, pcvars] = princomp(colon_genevalues);

% pcvars./sum(pcvars) * 100;
% cumsum(pcvars./sum(pcvars) * 100);

plot (pcvars)

%figure
scatter(zscores(:,1),zscores(:,2));
xlabel('First Principal Component');
ylabel('Second Principal Component');
title('Principal Component Scatter Plot');

%outputfilename
csvwrite('PCA_colon_2.csv', zscores (:,1:3));

%ColorSet = varycolor(50);
```

```
PCA_2=(colon_genevalues)'*PCA_colon_2;
save('colon_classification_data', 'PCA_2', 'colon_class', 'colon_samples')
```

PART C: RESULTS

Dataset 1 was filtered and it was expected that if anything the classification performance would be better than the control dataset (highest classification performance 85%). The low classification performance for Dataset 1 was true for all methods which seemed to indicate a problem possibly with the processing steps. (see Table 6 below).

Upon investigation it was found by looking at the confusion matrices that the number of instances for this dataset was saved as 248 instead of the sample number of 62. This was found to be because there were some appended data in the file 'PCA_filtered_set1.csv'. These errors were fixed and the file saved as 'PCA_filtered_set1_2.csv' and was then converted to the .arff format as 'Weka_Data_1_2.arff'. There were still problems running classification algorithms – many were not available.

Further investigation revealed that the class attribute had not been converted to text. This was done and the file saved as 'PCA_filtered_set1_3.csv'. This file was converted and saved as 'Weka_Data_1_3.arff'. The methods used on the other datasets were also run on this dataset and the results are shown in Table 4 in Chapter 3 (above).

	Classification Method	Correct	Incorrect	Confusion Matrix	
				a,b <-- classified as	
1	BayesNet	63.71	36.29	157 3 | a = negative	87 1 | b = positive
2	NaiveBayes	68.55	31.45	154 6 | a = negative	72 16 | b = positive
3	NaiveBayesSimple	68.55	31.45	154 6 | a = negative	72 16 | b = positive
4	NaiveBayesUpdateable	68.55	31.45	154 6 | a = negative	72 16 | b = positive
5	RadomTree	67.34	32.66	153 7 | a = negative	74 14 | b = positive
6	REPTree	64.92	35.08	159 1 | a = negative	86 2 | b = positive
7	Logistic	69.35	30.65	154 6 | a = negative	70 18 | b = positive
8	Multilayer Perceptron	66.53	33.47	148 12 | a = negative	71 17 | b = positive
9	RBFNetwork	69.35	30.65	157 3 | a = negative	73 15 | b = positive
10	SimpleLogistic	68.95	31.05	155 5 | a = negative	72 16 | b = positive
11	SMO	64.11	35.89	159 1 | a = negative	88 0 | b = positive
12	Voted Perceptron	64.52	35.48	160 0 | a = negative	88 0 | b = positive
13	AdaBoostMI	65.73	34.27	151 9 | a = negative	76 12 | b = positive
14	Bagging	64.52	35.48	160 0 | a = negative	88 0 | b = positive
15	LogitBoost	67.34	32.66	152 8 | a = negative	73 15 | b = positive
16	MultiBoostAB	65.73	34.27	153 7 | a = negative	78 10 | b = positive
17	MultiClassClassifier	69.35	30.65	154 6 | a = negative	70 18 | b = positive
18	OrdinalClassClassifier	64.52	35.48	160 0 | a = negative	88 0 | b = positive
19	Random Committee	69.35	30.65	157 3 | a = negative	73 15 | b = positive

Table 6: Results for Dataset 1

There was clearly a problem shown by the confusion matrices recorded. E.g. taking the confusion matrix for the first classifier BayesNet:

Predicted Class			
		Positive	Negative
Actual Class	Positive	157	3
	Negative	87	1

This would indicate that there are 157 + 3 + 87 + 1 = 248 samples. In fact, the total number of samples was only 62. This led to the creation of Data 1_3 as mentioned above.

The results for classification were much better for Data 1_3 as compared to Dataset 1 (see Table 7):

	Classification Method	Correct	Incorrect	Confusion Matrix	
				a,b <-- classified as	
1	BayesNet	59.68	40.32	29 11 | a = negative	14 8 | b = positive
2	NaiveBayes	80.65	19.35	34 6 | a = negative	6 16 | b = positive
3	NaiveBayesSimple	80.65	19.35	34 6 | a = negative	6 16 | b = positive
4	NaiveBayesUpdateable	80.65	19.35	34 6 | a = negative	6 16 | b = positive
5	RandomTree	70.97	29.03	30 10 | a = negative	8 14 | b = positive
6	REPTree	70.97	29.03	37 3 | a = negative	15 7 | b = positive
7	Logistic	82.26	17.74	35 5 | a = negative	6 16 | b = positive
8	Multilayer Perceptron	83.87	16.13	33 7 | a = negative	3 19 | b = positive
9	RBFNetwork	79.03	20.97	35 5 | a = negative	8 14 | b = positive
10	SimpleLogistic	80.65	19.35	35 5 | a = negative	7 15 | b = positive
11	SMO	75.81	24.19	35 5 | a = negative	10 12 | b = positive
12	Voted Perceptron	64.52	35.48	40 0 | a = negative	22 0 | b = positive
13	AdaBoostMI	79.03	20.97	33 7 | a = negative	6 16 | b = positive
14	Bagging	79.03	20.97	36 4 | a = negative	9 13 | b = positive
15	LogitBoost	69.35	30.65	30 10 | a = negative	9 13 | b = positive
16	MultiBoostAB	77.42	22.58	33 7 | a = negative	7 15 | b = positive
17	MultiClassClassifier	82.26	17.74	35 5 | a = negative	6 16 | b = positive
18	OrdinalClassClassifier	79.03	20.97	37 3 | a = negative	10 12 | b = positive
19	Random Committee	72.58	27.42	32 8 | a = negative	9 13 | b = positive

Table 7: Results for Data1_3

This data was filtered and reduced to three dimensions using PCA.

	Classification Method	Correctly Classified Instances	Incorrectly Classified Instances	Confusion Matrix	
				a,b <-- classified as	
1	BayesNet	72.58	27.42	30 10 | a = negative	7 15 | b = positive
2	NaiveBayes	77.42	22.58	33 7 | a = negative	7 15 | b = positive
3	NaiveBayesSimple	72.58	27.42	30 10 | a = negative	7 15 | b = positive
4	NaiveBayesUpdateable	74.19	25.81	32 8 | a = negative	8 14 | b = positive
5	RandomTree	64.52	35.48	29 11 | a = negative	11 11 | b = positive
6	REPTree	72.58	27.42	36 4 | a = negative	13 9 | b = positive
7	Logistic	83.87	16.13	36 4 | a = negative	6 16 | b = positive
8	Multilayer Perceptron	72.58	27.42	34 6 | a = negative	11 11 | b = positive
9	RBFNetwork	74.19	25.81	33 7 | a = negative	9 13 | b = positive
10	SimpleLogistic	83.87	16.13	35 5 | a = negative	5 17 | b = positive
11	SMO	85.48	14.52	37 3 | a = negative	6 16 | b = positive
12	Voted Perceptron	77.42	22.58	29 11 | a = negative	3 19 | b = positive
13	AdaBoostMI	72.58	27.42	34 6 | a = negative	11 11 | b = positive
14	Bagging	75.81	24.19	35 5 | a = negative	10 12 | b = positive
15	LogitBoost	72.58	27.42	33 7 | a = negative	10 12 | b = positive
16	MultiBoostAB	74.19	25.81	34 6 | a = negative	10 12 | b = positive
17	MultiClassClassifier	83.87	16.13	36 4 | a = negative	6 16 | b = positive
18	OrdinalClassClassifier	66.13	33.87	31 9 | a = negative	12 10 | b = positive
19	Random Committee	61.29	38.71	32 8 | a = negative	16 6 | b = positive

Table 8: Results for Dataset 2

This dataset was normalized and PCA reduced to 10 dimensions. The results are more accurate than those obtained for the control (Dataset 7 – see Table 13). Interestingly, Data 1_3 seemed to return more accurate results than those obtained from Dataset 2. E.g. looking at the Naïve Bayes results for the negative (tumor) class there were 6 misclassifications out of 40 for Data 1_3 and 7 misclassifications for Dataset 2.

This dataset was both normalized and filtered; it was PCA-reduced to 8 dimensions. Generally the classification results seem better than those reported in Tables 6-8. N.B. the accuracy seems to have improved.

	Classification Method	Correctly Classified Instances	Incorrectly Classified Instances	Confusion Matrix	
				a,b <-- classified as	
1	BayesNet	74.19	25.81	28 12 | a = negative	4 18 | b = positive
2	NaiveBayes	77.42	22.58	34 6 | a = negative	8 14 | b = positive
3	NaiveBayesSimple	75.81	24.19	34 6 | a = negative	9 13 | b = positive
4	NaiveBayesUpdateable	77.42	22.58	34 6 | a = negative	8 14 | b = positive
5	RandomTree	69.35	30.65	30 10 | a = negative	9 13 | b = positive
6	REPTree	70.97	29.03	32 8 | a = negative	10 12 | b = positive
7	Logistic	83.87	16.13	36 4 | a = negative	6 16 | b = positive
8	Multilayer Perceptron	74.19	25.81	33 7 | a = negative	9 13 | b = positive
9	RBFNetwork	77.42	22.58	33 7 | a = negative	7 15 | b = positive
10	SimpleLogistic	79.03	20.97	37 3 | a = negative	10 12 | b = positive
11	SMO	88.71	11.29	38 2 | a = negative	5 17 | b = positive
12	Voted Perceptron	82.26	17.74	32 8 | a = negative	3 19 | b = positive
13	AdaBoostMI	75.81	24.19	35 5 | a = negative	10 12 | b = positive
14	Bagging	79.03	20.97	35 5 | a = negative	8 14 | b = positive
15	LogitBoost	72.58	27.42	34 6 | a = negative	11 11 | b = positive
16	MultiBoostAB	75.81	24.19	34 6 | a = negative	9 13 | b = positive
17	MultiClassClassifier	83.87	16.13	36 4 | a = negative	6 16 | b = positive
18	OrdinalClassClassifier	77.42	22.58	32 8 | a = negative	6 16 | b = positive
19	Random Committee	77.42	22.58	36 4 | a = negative	10 12 | b = positive

Table 9: Results for Dataset 3

This dataset was created using Attribute Selection with PCA in WEKA. The accuracy shown by the confusion matrices is better than reported in earlier Tables.

	Classification Method	Correctly Classified Instances	Incorrectly Classified Instances	Confusion Matrix	
				a,b <-- classified as	
1	BayesNet	82.26	17.74	36 4 | a = negative	7 15 | b = positive
2	NaiveBayes	87.10	12.90	38 2 | a = negative	6 16 | b = positive
3	NaiveBayesSimple	85.48	14.52	38 2 | a = negative	7 15 | b = positive
4	NaiveBayesUpdateable	87.10	12.90	38 2 | a = negative	6 16 | b = positive
5	RandomTree	74.19	25.81	33 7 | a = negative	9 13 | b = positive
6	REPTree	90.32	9.68	38 2 | a = negative	4 18 | b = positive
7	Logistic	77.42	22.58	29 11 | a = negative	3 19 | b = positive
8	Multilayer Perceptron	80.65	19.35	35 5 | a = negative	7 15 | b = positive
9	RBFNetwork	85.48	14.52	34 6 | a = negative	3 19 | b = positive
10	SimpleLogistic	87.10	12.90	35 5 | a = negative	3 19 | b = positive
11	SMO	87.10	12.90	38 2 | a = negative	6 16 | b = positive
12	Voted Perceptron	83.87	16.13	35 5 | a = negative	5 17 | b = positive
13	AdaBoostMI	83.87	16.13	34 6 | a = negative	4 18 | b = positive
14	Bagging	87.10	12.90	36 4 | a = negative	4 18 | b = positive
15	LogitBoost	88.71	11.29	37 3 | a = negative	4 18 | b = positive
16	MultiBoostAB	85.48	14.52	35 5 | a = negative	4 18 | b = positive
17	MultiClassClassifier	77.42	22.58	29 11 | a = negative	3 19 | b = positive
18	OrdinalClassClassifier	87.10	12.90	36 4 | a = negative	4 18 | b = positive
19	Random Committee	82.26	17.74	37 3 | a = negative	8 14 | b = positive

Table 10: Results for Dataset 4

This dataset was created using Discretization and PCA in WEKA. Again, the results seem to have improved compared to earlier Tables.

	Classification Method	Correctly Classified Instances	Incorrectly Classified Instances	Confusion Matrix	
				a,b <-- classified as	
1	BayesNet	88.71	11.29	36 4 | a = negative	3 19 | b = positive
2	NaiveBayes	93.55	6.45	39 1 | a = negative	3 19 | b = positive
3	NaiveBayesSimple	95.16	4.84	39 1 | a = negative	2 20 | b = positive
4	NaiveBayesUpdateable	93.55	6.45	39 1 | a = negative	3 19 | b = positive
5	RandomTree	62.90	37.10	31 9 | a = negative	14 8 | b = positive
6	REPTree	90.32	9.68	36 4 | a = negative	2 20 | b = positive
7	Logistic	82.26	17.74	33 7 | a = negative	4 18 | b = positive
8	Multilayer Perceptron	88.71	11.29	36 4 | a = negative	3 19 | b = positive
9	RBFNetwork	91.94	8.06	39 1 | a = negative	4 18 | b = positive
10	SimpleLogistic	83.87	16.13	34 6 | a = negative	4 18 | b = positive
11	SMO	91.94	8.06	37 3 | a = negative	2 20 | b = positive
12	Voted Perceptron	91.94	8.06	38 2 | a = negative	3 19 | b = positive
13	AdaBoostMI	95.16	4.84	40 0 | a = negative	3 19 | b = positive
14	Bagging	90.32	9.68	37 3 | a = negative	3 19 | b = positive
15	LogitBoost	91.94	8.06	38 2 | a = negative	3 19 | b = positive
16	MultiBoostAB	95.16	4.84	40 0 | a = negative	3 19 | b = positive
17	MultiClassClassifier	82.26	17.74	33 7 | a = negative	4 18 | b = positive
18	OrdinalClassClassifier	93.55	6.45	39 1 | a = negative	3 19 | b = positive
19	Random Committee	72.58	27.42	37 3 | a = negative	14 8 | b = positive

This dataset showed the best classification results which were more accurate than all the other datasets used. This dataset was created using a combination of Attribute Selection, PCA and Discretization in WEKA.

	Classification Method	Correctly Classified Instances	Incorrectly Classified Instances	Confusion Matrix	
				a,b <-- classified as	
1	BayesNet	98.39	1.61	40 0 | a = negative	1 21 | b = positive
2	NaiveBayes	100.00	0.00	40 0 | a = negative	0 22 | b = positive
3	NaiveBayesSimple	100.00	0.00	41 0 | a = negative	1 22 | b = positive
4	NaiveBayesUpdateable	100.00	0.00	42 0 | a = negative	2 22 | b = positive
5	RandomTree	70.97	29.03	33 7 | a = negative	11 11 | b = positive
6	REPTree	98.39	1.61	40 0 | a = negative	1 21 | b = positive
7	Logistic	93.55	6.45	39 1 | a = negative	3 19 | b = positive
8	Multilayer Perceptron	96.77	3.23	40 0 | a = negative	2 20 | b = positive
9	RBFNetwork	96.77	3.23	40 0 | a = negative	2 20 | b = positive
10	SimpleLogistic	96.77	3.23	39 1 | a = negative	1 21 | b = positive
11	SMO	96.77	3.23	39 1 | a = negative	1 21 | b = positive
12	Voted Perceptron	98.39	1.61	40 0 | a = negative	1 21 | b = positive
13	AdaBoostMI	96.77	3.23	39 1 | a = negative	1 21 | b = positive
14	Bagging	96.77	3.23	39 1 | a = negative	2 21 | b = positive
15	LogitBoost	96.77	3.23	39 1 | a = negative	2 21 | b = positive
16	MultiBoostAB	96.77	3.23	39 1 | a = negative	2 21 | b = positive
17	MultiClassClassifier	93.55	6.45	39 1 | a = negative	3 19 | b = positive
18	OrdinalClassClassifier	98.39	1.61	39 1 | a = negative	0 22 | b = positive
19	Random Committee	91.94	8.06	40 0 | a = negative	5 17 | b = positive

Table 12: Results for Dataset 6

This was the control dataset, which was loaded into WEKA without any processing. Some of the methods would not work with the dataset since it was not normalized (NaïveBayesSimple) and others were too slow (MultilayerPerceptron). The accuracies of the classifications are quite low with high false negative and false positive results.

	Classification Method	Correctly Classified Instances	Incorrectly Classified Instances	Confusion Matrix	
				a,b <-- classified as	
1	BayesNet	77.42	22.58	31 9 | a = negative	5 17 | b = positive
2	NaiveBayes	53.23	46.77	19 21 | a = negative	8 14 | b = positive
3	NaiveBayesSimple	NA			
4	NaiveBayesUpdateable	53.23	46.77	19 21 | a = negative	8 14 | b = positive
5	RandomTree	67.74	32.26	26 14 | a = negative	6 16 | b = positive
6	REPTree	69.35	30.65	37 3 | a = negative	16 6 | b = positive
7	Logistic	74.19	25.81	36 4 | a = negative	12 10 | b = positive
8	Multilayer Perceptron	Slow			
9	RBFNetwork	79.03	20.97	35 5 | a = negative	8 14 | b = positive
10	SimpleLogistic	77.42	22.58	34 6 | a = negative	8 14 | b = positive
11	SMO	85.48	14.52	36 4 | a = negative	5 17 | b = positive
12	Voted Perceptron	75.81	24.19	36 4 | a = negative	11 11 | b = positive
13	AdaBoostMI	74.19	25.81	35 5 | a = negative	11 11 | b = positive
14	Bagging	79.03	20.97	36 4 | a = negative	9 13 | b = positive
15	LogitBoost	75.81	24.19	35 5 | a = negative	10 12 | b = positive
16	MultiBoostAB	85.48	14.52	37 3 | a = negative	6 16 | b = positive
17	MultiClassClassifier	74.19	25.81	36 4 | a = negative	12 10 | b = positive
18	OrdinalClassClassifier	75.81	24.19	33 7 | a = negative	8 14 | b = positive
19	Random Committee	72.58	27.42	36 4 | a = negative	13 9 | b = positive

Table 13: Results for Dataset 7

APPENDIX 3: METHODS AND RESOURCES

PART A: CLASSIFICATION METHODS USED IN WEKA

BAYES

1. BayesNet – this function learns Bayesian networks under the assumptions that attributes are nominal and there are no missing values. For some of the processed datasets the nominal assumption will hold – for the raw dataset it does not. In this case the attributes are discretized by the method.

   Search is carried out using two possible algorithms:

   - TAN
   - K2

   In this study, this function is run using K2 because the TAN algorithm is linear in the number of instances and quadratic in the number of attributes. It will therefore be computationally more expensive. To improve search speed AD trees was also used.

2. NaiveBayes – implements a probabilistic Naive Bayes classifier. This technique gives equal weight to all attributes in the dataset and because of this principle the performance of the classifier can be affected by redundant attributes. Also, in this algorithm it is assumed that attributes are normally distributed. Since this is a biological dataset it would seem that such an assumption is reasonable. Discretization of the attributes will nullify the need for such an assumption. This method can use kernel density estimators to determine the kind of attribute distribution. In this study, the method was run with and without the kernel estimator. A better classification result was then recorded.

3. NaiveBayesSimple – uses normal distribution to model numeric attributes.

4. NaiveBayesUpdateable – processes one instance at a time and can also use a kernel estimator but will not discretize the data.

TREES

5. RandomTree – builds a tree using tests based on a given number of random features at each node. This is done with no pruning.

6. REPTree – the regression tree used by this method is built by using variance reduction and prunes the tree using reduced-error pruning. This method is optimized for speed so it only sorts numeric attributes once.

FUNCTIONS

7. Logistic – builds and uses multinomial logistic regression models with a ridge estimator to guard against over-fitting.

8. MultilayerPerceptron – this function allows the user to change the number of passes (epochs) through the data as well as the learning rate and the momentum. The standard gradient descent method is used to determine weights for the units – this method looks for a minimum which is the value assigned. However, it can only detect a local minimum and MLP error functions often exhibit a number of minima. This is therefore a drawback.

9. RBFNetwork – this method implements a Gaussian radial basis function network. This kind of network has three layers and only differs from

Multilayer Perceptrons in the way the hidden units perform the calculations. The activation of each unit depends on the distance between the unit and the test point. Most often the activation function takes the shape of a Gaussian function – each unit may have differing widths for their associated functions. RBF networks assign all attributes an equal weight and so cannot deal with irrelevant attributes unlike MLPs.

10. SimpleLogistic – builds simple regression models fitting the models using LogitBoost with simple regression functions as base learners and deciding the number of iterations by performing cross-validation.

11. SMO is a method that implements the sequential minimal optimization algorithm for training a support vector classifier. It was run using a polynomial kernel.

12. VotedPerceptron – in a linear perceptron the algorithm iterates through the training set and updates the weight vector every time there is a misclassification. In this perceptron the solution vector depends on the order in which instances are encountered. To make this algorithm more stable all weight vectors encountered during the learning process are allowed to vote on a prediction. This is the salient feature of a VotedPerceptron algorithm.

META

The Metalearning algorithms in WEKA are often used in more recent times in cancer classification. Some of these have been used in this study including:

13. AdaBoostMI – boosting aims to try and combine classification methods that complement each other. It uses voting from the different algorithms to combine the output of individual models. Boosting combines models of the same type (for instance a number of different decision trees). Boosting is an iterative process, i.e. new models are influenced by those built before. New models are encouraged to become experts for instances misclassifie by previous models.

14. Bagging – this is similar to boosting.

15. LogitBoost – performs additive logistic regression.

16. MultiBoostAB – combines boosting with a variant of bagging to prevent overfitting.

17. MultiClassClassifier.

18. OrdinalClassClassifier.

19. Random Committee – an ensemble of base classifiers is built and their predictions are averaged. Each classifier is based on the same data but uses a different seed number.

PART B: RESOURCES

Software packages and algorithms available for research purposes include:

1. Matlab.

2. Weka and BioWeka (extension library).

3. Clementine (PASW Modeler) – SPSS. (to build classifier decision trees, k-nn, svm, C5.0, C&RT, Neural Nets).

4. XCluster is a software available from <http://fafner.stanford.edu/~sherlock/cluster.html>. Re-compilation may be necessary.

5. ScanAlayze is a software for analysing microarray data available from: < http://rana.lbl.gov/eisen/?page_id=41> . Four softwares are also available for Cluster Analysis and Visualization Software < http://rana.lbl.gov/eisen/?page_id=42>

6. SVM have been mentioned extensively in the literature. They are described in details at :< http://www.isis.ecs.soton.ac.uk/resources/svminfo/> which also includes a Matlab Toolbox for SVMs.

7. The MAExplorer software can be used to analze microarray datasets and is available at: < http://maexplorer.sourceforge.net/>

8. CLUTO is a family of data clustering and cluster analysis programs. They are available for download at: < http://glaros.dtc.umn.edu/gkhome/views/cluto>

9. A number of software packages are mentioned at <http://www.genopole-lille.fr/logiciel/microarray/norm_tools.html> for microarray analysis.

10. J-Express allows users some clustering analysis capability: < http://www.molmine.com/tryMain.php>

11. Infer.Net is a state of the art package from Microsoft that allows users to build Bayesian networks. http://research.microsoft.com/en-us/um/cambridge/projects/infernet/default.aspx

12. The Spider is an object oriented environment inside Matlab < http://www.kyb.tuebingen.mpg.de/bs/people/spider/>

13. Least Squares –SVM have been implemented as a Matlab toolbox at : < http://www.esat.kuleuven.ac.be/sista/lssvmlab/>

14. An implementation of PAM (Prediction Analysis for Microarrays) is available at
< http://www-stat.stanford.edu/~tibs/PAM/>

15. A list of Open Source packages for microarray data analysis are available at:
< http://www.kdnuggets.com/solutions/microarray.html#free>

16. Matlab code is available for some methods at:
<http://www.public.asu.edu/~jye02/Software/index.html>at:
<http://www.tsi.enst.fr/~gfort/GLM/Programs.html> and at:
http://www.tech.plym.ac.uk/spmc/links/bioinformatics/microarray/microarray_matlab.html

17. A list of SVM implementations is available at: <http://www.support-vector-machines.org/SVM_soft.html> and http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/index.html

18. A package for normalization geNorm is available at :< http://medgen.ugent.be/~jvdesomp/genorm/#download> (password for download obtained via email) and candidate genes for normalization can be obtained using NormFinder at: < http://www.mdl.dk/publicationsnormfinder.htm>

19. The CS.4 algorithm is mentioned at <http://datam.i2r.a-star.edu.sg/datasets/krbd/>. An email was sent to request source code with no response.

20. The Data Description (DD) toolbox available at:< http://ict.ewi.tudelft.nl/~davidt/dd_tools.html> provides resources but for one-class problems.

21. The Netlab toolbox provides methods coded in Matlab and can be found at: < http://www.ncrg.aston.ac.uk/netlab/index.php>

22. An Open Source version of SVM is also available at: < http://svmlight.joachims.org/>
    There is also a set of functions that make it easier to call the method from within Matlab: < http://ida.first.fraunhofer.de/~anton/software.html>

23. A number of Clustering implementations are listed at:
    < http://www.dcorney.com/ClusteringMatlab.html>

24. K-means resources are listed at
    < http://people.revoledu.com/kardi/tutorial/kMean/Resources.htm>

25. Fuzzy Clustering resources listed at
    < http://people.revoledu.com/kardi/tutorial/kMean/Resources.htm>

26. A number of software packages for microarray analysis can also be accessed at:
    < http://www.broadinstitute.org/science/software>

27. A number of classification tree algorithms can be found at:
    < http://www.stat.wisc.edu/~loh/>

28. An implementation of an ensemble classifier is available at:
    < http://www.ams.sunysb.edu/~hahn/research/CERP.html>

29. An SVM library can be found at :< http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

30. An implementation of Random Forests is available at
    < http://ligarto.org/rdiaz/Papers/rfVS/randomForestVarSel.html>

31. A visualisation package and other methods are included at
    < http://hanchuan.peng.googlepages.com/hanchuan%27ssoftware>

32. The source code for some methods include some forms of SVMs are available at:
< http://showelab.wistar.upenn.edu/>

33. The Boosting method has been implemented as an R package and is available at: < http://showelab.wistar.upenn.edu/>

34. GIST is a set of software tools for SVM and k-PCA analysis. It is implemented in C and can be accessed at < http://www.bioinformatics.ubc.ca/gist/index.html>

35. An optimization Matlab implemented toolbox is available at:
< http://www.ise.ncsu.edu/mirage/GAToolBox/gaot/>

36. Another SVM Matlab implementation can be accessed at:
< http://svm.sourceforge.net/download.shtml>

37. A Gene Pattern Analysis Suite is also available online: < http://www.gepas.org/>

38. A number of software tools for microarrays are available at:
< http://www.jcvi.org/cms/research/software/#c622>

39. A web application for the integrated analysis of global gene expression patterns in cancer can be accessed at: < http://bioinformatics2.pitt.edu/GE2/GEDA.html>

40. AMIADA is an integrated computer program for organizing, exploring, visualizing, and analyzing microarray data and can be accessed at:
< http://dambe.bio.uottawa.ca/amiada.asp>

41. Another package that can be explored is KNIME
< http://www.knime.org/introduction>

Not all of the listed resources were used in this study. The scope of this study was limited some packages were not as functional given limitations in processing power,

the need for expensive licenses, etc. A brief analysis was begun the results of which can be seen in the Table below.

Resource Analysis

Resource	Analysis	Used
XCluster	Oracle installation needed.	No
ScanAlyze	Used for microarray raw image data analysis	No
Cluster/ Tree View	Installed	Yes
MAExplorer	Problem with installation files.	No
SeqExpress	Machine hangs on loading data	No
JExpress	J-Express config file cannot be read.	No
Fuzzy Clustering Toolbox	Matlab codes. <http://www.mathworks.com/matlabcentral/fileexchange/7486>	Yes
My Clustering Toolbox	Similar algorithms coded as Fuzzy Clustering Toolbox.	No
Neuralware	http://www.neuralware.com/products.jsp Commercial use	No
MatArray	http://www.ulb.ac.be/medecine/iribhm/microarray/toolbox/getstarted.html Kmeans and hierarchical clustering codes. Most methods already coded in Matlab.	No