ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

MULTI-LAYER PERCEPTRON ARTIFICIAL NEURAL NETWORK PREDICTIVE MODELLING OF GENOMIC AND MASS SPECTROMETRY DATA IN BIOINFORMATICS

LEE JAMES LANCASHIRE

A thesis submitted in partial fulfilment of the requirements of The Nottingham Trent University for the degree of Doctor of Philosophy

September 2006

DECLARATION

The author has not been a registered candidate nor an enrolled student for another award of the University or other professional institution during this research programme. Material contained in this thesis has not been used in any other submission for an academic award and is entirely the author's individual contribution. The author has attended appropriate lectures, seminars and conferences in partial fulfilment of the requirements of the degree.

Signed.....

(Candidate)

Signed.....

(Director of studies)

COPYRIGHT

ACKNOWLEDGEMENTS

In memory of Joan Lancashire, Mary and George Nicol, and Millie Smith

First of all I am indebted to Graham Ball for his overwhelming support and advice throughout the course of this research. His enthusiasm for the subject area has passed onto me and without him none of this would have been possible. I would like to thank Bob Rees for the ideas, advice and encouragement he has supplied during the research. I would also like to thank Bal Matharoo-Ball, to whom I am extremely grateful for the discussions, advice and support during my research and writing. To John, Vicky and Simon, for keeping me sane and for never letting things get too serious in the office! To Catherine who I am truly thankful for her support and encouragement, and also for the patience and tolerance she showed firstly when I was writing and secondly when she was proof-reading this thesis! Finally, I would like to thank my parents Mary and Terry, and sister Samantha for always being there unconditionally, without them this would also not have been possible.

ABSTRACT

The development of proteomic and genomic applications for the research into different diseases has paved the way for the development of novel approaches for the way in which these systems can be investigated. This provides a novel insight into how proteins and genes are being regulated under different conditions. These approaches are self-limited by the volume of data which they produce, with the majority often being noisy and redundant. Therefore these technologies must be coupled with appropriate computational approaches that are capable of identifying components that are the most important in differentiating between disease states of interest. These must be robust enough to cope with data of this size and nature in order to provide an in depth understanding of these complex proteomic and genomic patterns. This in turn will lead to methods for prognosis and diagnosis of diseases such as cancer, by providing an insight into the proteins and genes which are being expressed differentially depending on the current status of the disease. The research contained within this thesis describes the development and validation of multi-layer perceptron Artificial Neural Network based methodologies for variable selection, biomarker identification and predictive modelling of mass spectrometry and genomic data. Many datasets were used from a range of different sources, such as the mass spectrometry analysis of bacterial pathogens, the mass spectrometry analysis of patients suffering from different grades of melanoma, and gene expression analysis of patients with breast cancer. Results showed that robust and reproducible predictive models could be generated, which predicted class to extremely high accuracies (greater than 95 %) for blind datasets. These approaches were enhanced further to allow for the interrogation of biomarkers identified during the course of the analyses with techniques such as response surface analysis and population structure analysis. Response surfaces showed the direction of response of a biomarker of interest, in relation to whether it was being up or down regulated in a given disease outcome under study. As an adjunct population profiling showed the potential for identifying sub-groups of patients which could subsequently be used to identify those at risk of disease spread based upon their genetic profiles. Finally methods for the derivation of gene regulatory networks have been proposed which allows interactions and pathways to be derived to show how the change in expression of one gene causes a resulting change in many others. As such the results from the experimental work performed in this thesis have resulted in novel contributions to the field of bioinformatics.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AGP	Alpha 1-acid glycoprotein
ANN	Artificial Neural Network
ART	Adaptive Resonance Theory
AUC	Area under the curve
BLM	Bloom syndrome
BP	Back-Propagation
CA IX	Carbonic Anhydrase IX
CA125	Cancer antigen 125
CAPing	Correlated activity pruning
ER	Oestrogen receptor
EST	Expressed sequence tag
***K*NN**	*K*-Nearest Neighbours
LDA	Linear Discriminant Analysis
LN	Lymph node
m/z	mass/charge ratio
MALDI-TOF	Matrix-assisted laser desorption/ionsisation- time of flight
MLP	Multi-layer perceptron
MS	Mass spectrometry
MSE	Mean Squared Error
NCTC	National Collection of Type Culture
PCA	Principal components analysis
PSA	Prostate specific antigen
RBF	Radial Basis Function
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RSCV	Random sample cross validation
SELDI	Surface Enhanced Laser Desorption Ionisation
SVM	Support Vector Machine

CHAPTER 1. INTRODUCTION

1.1. Aims

The research contained within this thesis describes the development and validation of Artificial Neural Network (ANN) methodologies aimed specifically towards the analysis of complex biological data, such as that generated by mass spectrometry based proteomic and microarray based genomic analysis. Here, computer science and biology have been combined to apply bioinformatic approaches to these types of datasets in order to generate models of the biological systems contained within. The main aims of this thesis are as follows:

- To use multi-layer perceptron artificial neural networks to develop and validate predictive and classification models from real biological datasets, where the gold standard for classification was the class that was originally assigned to the cases by the clinician.
- To determine the importance and relationships of individual variables with respect to a range of biological datasets within these models, thus deriving meaning from the ANN "black box".
- To design, write and implement software tools for the automated identification and characterisation of key subsets of biomarkers, and the subsequent validation of these methods.
- To show how the models and biomarkers within these can be interrogated further still, in order to characterise their behaviour with regards to the problem of interest, for example disease status.
- To investigate the potential for the use of ANN modelling in identifying interactions between biomarkers identified in these complex systems.

These aims are substantiated by the work contained within this thesis. This includes the design, writing and subsequent implementation of software and the results from the experimentation carried out, resulting in novel contributions to the field of bioinformatics.

1.2. Introduction to disease biomarkers

As diseases are generally multi-factorial in nature, the use of biomarkers in detecting early onset of disease, predicting clinical outcome or even response to therapy is of paramount importance. There are two main categories that biomarkers generally belong to, and these can be grouped according to their clinical applications. The first are diagnostic markers that are used to initially define the disease, whilst the second are prognostic markers that can aid in the prediction of how the disease will develop (Xiao, et al., 2005). The term biomarker could be representative of almost any changes occurring in the cell, serum, or biological system. It is believed that the direct measurement of protein expression may be one of the most accurate methods in providing an insight into the development of disease (Xiao, et al., 2005). The discovery and validation of biomarkers with strong predictive power for a biological condition or response is extremely challenging. Extensive medical research programs have focused on developing techniques for the detection of novel markers which potentially have an application in a clinical setting for the detection and diagnosis of disease. The standard measurements of value to clinicians to assess the effectiveness of a given marker are sensitivity, specificity and accuracy. Sensitivity is the percentage, or probability, of the occurrence of correct diagnosis for a diseased patient. Specificity is the probability of correct diagnosis for a healthy patient, whilst accuracy gives an overall measure of correct diagnosis of both healthy and diseased combined.

By far the most active area of disease biomarker research is that in the field of cancer. Cancer remains a major health problem, with untold physical, psychological and economic costs to society. Elimination of cancer would reduce health care costs and enhance quality of life. Unfortunately the markers currently available are often inaccurate and unreliable. Serum is a particularly useful source for possible biomarker identification because it is easily obtained, has a high protein concentration and many of the protein and peptide constituents are shed from cells and tissues (Kennedy, 2002; Sasaki, et al., 2002). For these reasons specific serum markers are routinely used to identify disease states, for example the up-regulation of prostate specific antigen (PSA) is currently used as the best available marker for prostate cancer detection. Unfortunately PSA lacks specificity (Adam, et al., 2002; Djavan, et al., 1999). Cancer antigen 125 (CA125) is the most widely used biomarker in ovarian cancer. CA125 concentrations are abnormal in around 80 % of patients with advanced disease. However, CA125 is only increased in around 50 % of patients with Stage I disease, resulting in a positive predictive value of less than 10 %

when used as a single marker (Petricoin, et al., 2002). Other markers such as CA14.3, researched for its use in breast cancer detection, shows only 23 % and 69 % sensitivity and specificity respectively in detecting early stage disease (Li, et al., 2002). Melanoma is another important disease with a similar problem. It is increasing in incidence and in later stages is non-responsive to chemo- or radiotherapies. The diagnosis of malignant melanoma remains one of the most difficult to render in surgical pathology, partially because of its extreme histological variability. When diagnosed early, simple surgical treatment can be extremely effective but survival rate decreases to just 5% over 5 years for metastatic (Stage IV) melanoma. Serum LDH has been found to be a sensitive indicator of tumour load, but unfortunately is only of value in Stage IV disease (Deichmann, et al., 1999; Franzke, et al., 1998). Limits in the sensitivity and/or specificity of the currently available melanocytic markers such as anti-S100, HMB45, and anti-MelanA further complicate this problem.

This all further highlights the requirements for developments in the identification of novel biomarkers for early disease diagnosis and detection, as well as to provide novel targets for drug discovery and therapeutic intervention. Because of the rapid increase in applications of proteomic and genomic technologies to disease, methods focused on applying computational algorithms in data mining either the mass spectra of proteins and peptide fragments, or the gene expression data from patients in blood and serum are becoming extremely popular and showing great potential (Hilario, et al., 2004; Li, et al., 2004; Paweletz, et al., 2000; Petricoin, et al., 2002; Petricoin and Liotta, 2002; Petricoin, et al., 2002).

1.3. Proteomic technologies

The proteome has been defined as the full complement of proteins expressed by a genome at any specific point in time (Wasinger, et al., 1995). So the field of proteomics lends itself to the development of methods which are capable of characterising these proteins in any given biological sample, or determining protein changes between different disease states. Proteomics uses a combination of laboratory techniques in concert with bioinformatics to quantify proteins, and then to identify disease associated protein biomarkers to assist in disease diagnosis or prognosis, or in selecting novel targets for drug therapy. Recently, proteomic analysis does not even rely upon knowing the identities of the proteins which are being detected in a sample, but rather the pattern of proteins present, which may then be used as a fingerprint to diagnose and predict disease.

Therefore in order to identify proteins of interest which may then serve as potential biomarkers, there must be fast and sensitive protein identification methods in place which are capable of generating meaningful information from complex samples. Thus there exists a need for the development of novel methods which are capable of aiding in the discovery of novel protein biomarkers that possess high predictive and prognostic characteristics which can then be transferred into clinical practice (Xiao, et al., 2005).

1.3.1. MALDI-TOF MS

One of the most common methods used in assessing the proteome of a given sample is mass spectrometry (MS). This is capable of determining the changes in protein abundance expressed in a cell, which leads to the potential for characterisation. A mass spectrometer consists of an ion source, a mass analyser to measure the mass/charge ratio (*m/z*) of the analytes which have been ionised (mass spectrometers do not measure mass directly, but rather the mass to charge ratio of ions formed), and finally a detector that records the number of ions at each *m/z* value, generating a "fingerprint" for the sample being analysed. An overview of the MS system can be seen in Figure 1.

One of the most common MS instruments used is the MALDI-TOF analyser (Matrix-assisted laser desorption/ionisation-time of flight). Here, the sample to be analysed is mixed together with a chemical matrix, for example sinapinic acid, which absorbs light at a specific wavelength. This mixture of sample and matrix is then spotted onto a plate and

then allowed to evaporate in air to form a crystal lattice structure in which the sample is integrated. An ion source, or laser, is then fired at this target, ejecting the proteins or peptides in the sample from the target surface into the gas phase, in the form of ions (Liebler, 2002). Although multiply charged ions can be produced, only singly charged ions are usually observed in MALDI (Wysocki, et al., 2005), and therefore the *m/z* value of a given ion is equal to its molecular mass in Daltons (Da.). The MALDI-TOF system, as the name suggests, utilises a time-of-flight mass analyser. These accelerate the ions by applying a short voltage gradient and measuring the time taken for these ions to travel down a field free flight tube to the other side where they strike a detector (Wysocki, et al., 2005). The speed with which the ions fly down the analyser tube is proportional to their *m/z* value, the greater the *m/z* the faster they fly (Liebler, 2002).

Since MS is capable of high throughput protein identification, high analytical sensitivity, and very good mass accuracy, it is fast becoming the method of choice in biomarker discovery (Aebersold and Mann, 2003). With this method, there lies the possibility of early disease detection through large-scale analyses, and more importantly, the non-invasive collection of biological samples (Hilario, et al., 2004). MALDI-TOF MS is one of the most popular forms of MS because of its robustness and ease of use (Colantonio and Chan, 2005), and is much used to identify proteins by peptide mapping, or peptide-mass fingerprinting (Aebersold and Mann, 2003). MALDI allows for the analysis of high molecular weight compounds with high sensitivity and little or no fragmentation (Dhingra, et al., 2005).

1.3.2. SELDI-TOF MS

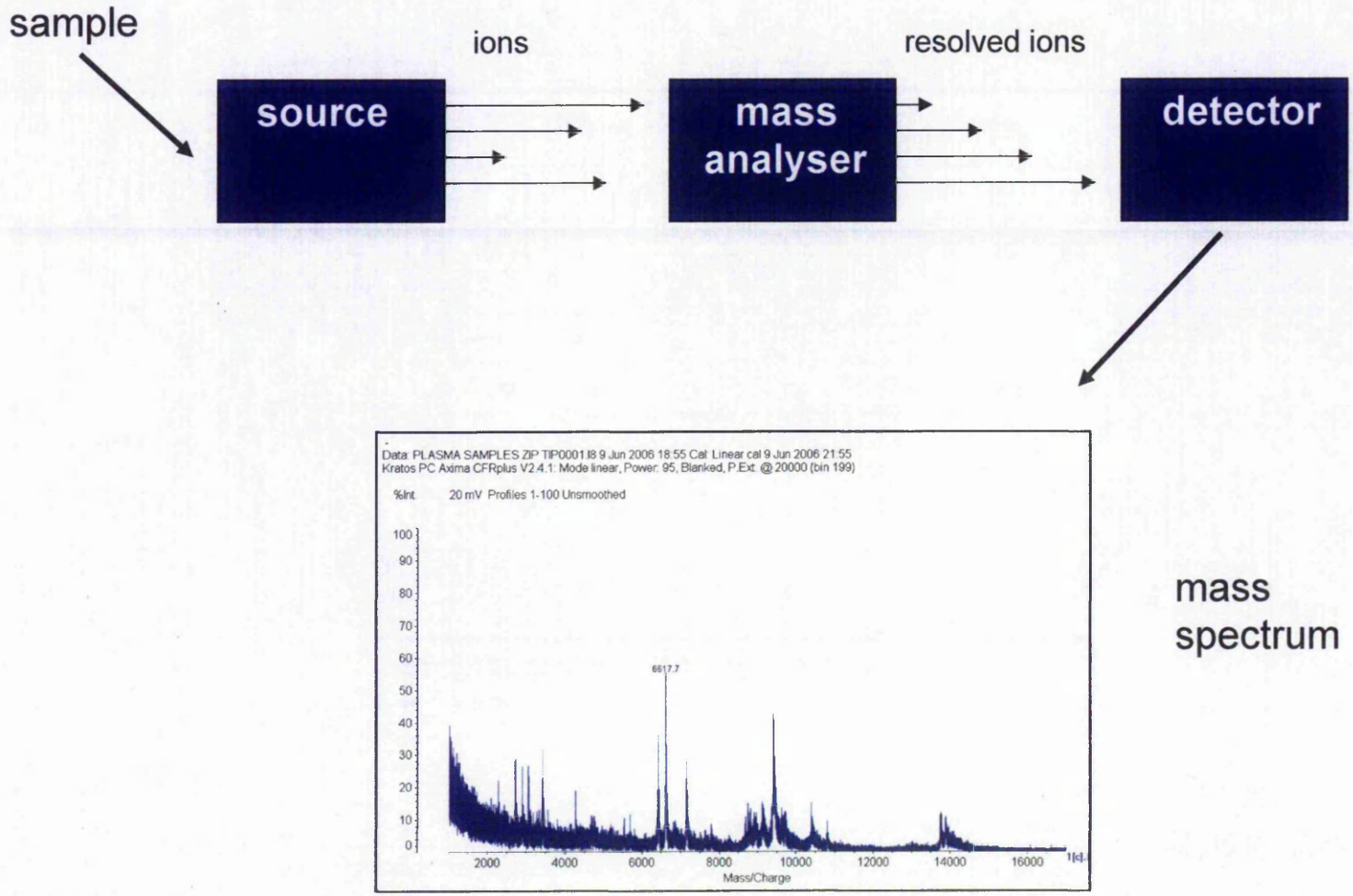
A recent development in proteomics has been the introduction of Surface Enhanced Laser Desorption Ionisation (SELDI) MS. This system is currently implemented in the ProteinChip System by Ciphergen Biosystems Inc. (Fremont, CA, USA) and is a high-throughput array based technology which can be used in the analysis of complex mixtures. The key difference between SELDI and MALDI MS is that SELDI utilises metal surfaces to which the sample is immobilised, and the sample is washed to remove any compounds that do not bind. In this way a large number of proteins from small quantities of a complex sample can be analysed very rapidly. It produces spectra of complex mixtures based on the *m/z* ratio of the proteins and on their binding affinity to the chip surface, and therefore differentially expressed proteins may be determined from these protein profiles by

comparing the patterns of peak intensities from samples representing different pathological states (Seibert, et al., 2004). These mass spectra are organised as an *m* (peaks) x *n* (samples) matrix, where the peak intensity corresponds to the relative abundance of the proteins at a specific molecular mass (White, et al., 2004).

This system produces a protein fingerprint of the masses, and this is the major advantage and at the same time, disadvantage of the technology. The advantage is that the system provides an alternative method for the differential display of biomarkers, and it provides information regarding the masses which are required to make a correct diagnosis. However, in order to identify potential therapeutic targets, the identification of these biomarkers needs to be known, along with their biological function, and it is this information that the SELDI system does not provide.

The basic procedure for analysing samples using the SELDI technology is said to be relatively straightforward (Seibert, et al., 2004), with it being possible to apply almost any type of solution to the ProteinChip Arrays. The chips themselves are usually chromatographic surfaces with specific characteristics which retain proteins with the required characteristics. Examples of the most popular chips used are hydrophobic, cationic, anionic, metal ion presenting or hydrophilic, and it is common to combine multiple chip types to retain proteins of different groups from the samples of interest (Seibert, et al., 2004).

Figure 1. Schematic of the mass spectrometry process



1.4. Genomic technologies: Gene expression profiling using microarrays

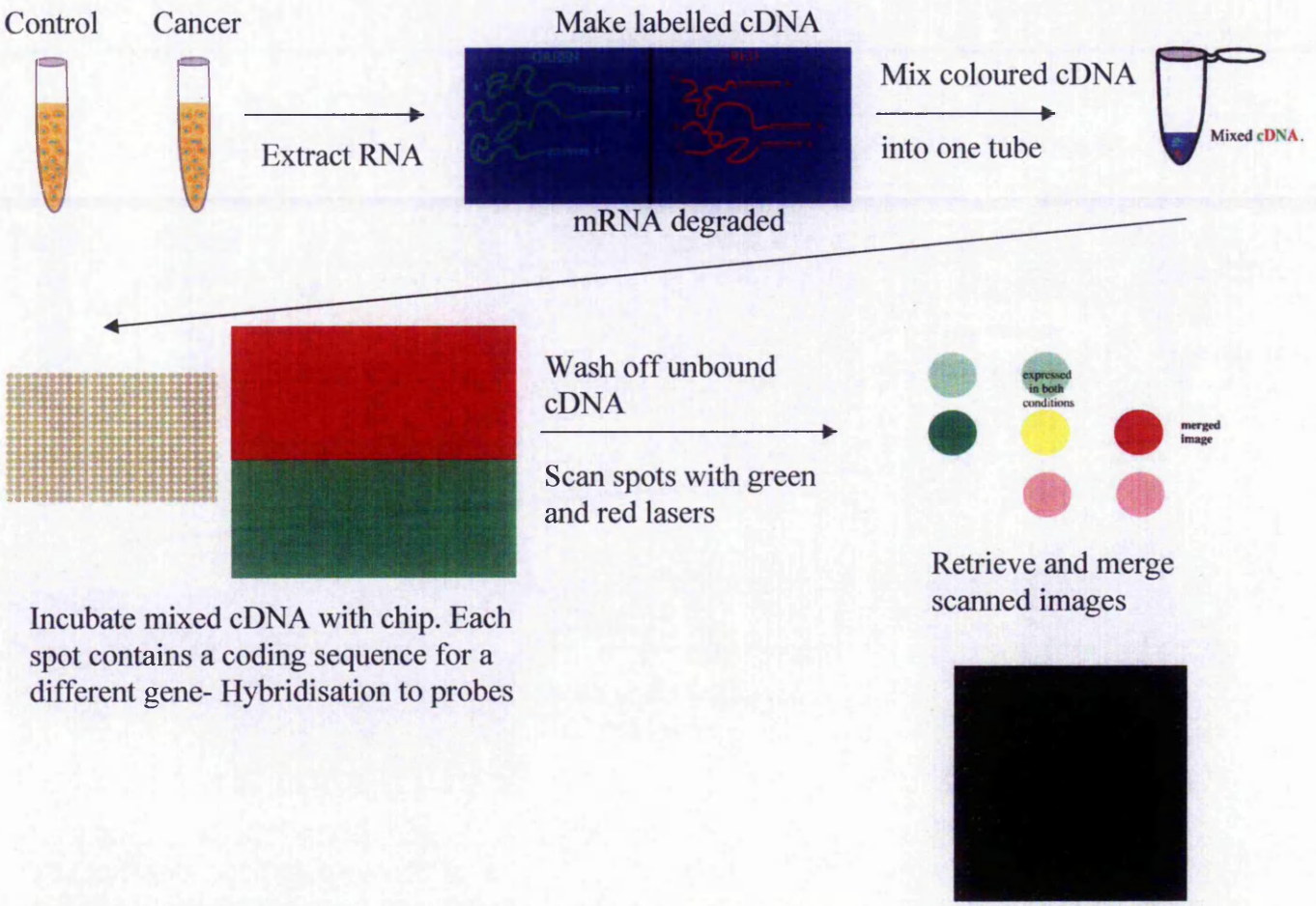
Just as it is important to identify which proteins in a biological system are differentially expressed during different disease states, it is also essential to study the expression patterns of genes within this system. If it is possible to monitor how different cellular states affect the expression of genes, then this will further our understanding of the various mechanisms that determine disease onset and progression. As proteomics is the study of the protein complement of a cell, the term genomics has been attributed to the study of the complete genetic makeup of an organism. DNA microarrays may be used here to measure the expression patterns of thousands of genes in parallel.

The most common application of microarrays is in the study of differential gene expression in disease. Diversion from normal physiology is frequently accompanied by a host of histological and biochemical changes, including changes in gene expression as a result of disease (Debouck and Goodfellow, 1999). A DNA microarray consists of a solid surface, onto which DNA molecules have been chemically bonded. The purpose of microarrays is to detect the presence and the abundance of labelled nucleic acids in a given biological sample, which will then hybridise to the DNA on the array, and become detectable via the label. The source of the labelled nucleic acids is the mRNA of the sample of interest, so therefore the purpose of a microarray is to measure gene expression. As there may be thousands of different DNA molecules bonded to an array, it is possible to measure the expression of many thousands of genes simultaneously, leading to the potential for extremely high throughput analysis.

There are several distinct steps in using a microarray in order to measure the gene expression of a sample of interest. These include (i) sample preparation and labelling, (ii) hybridisation, (iii) washing, and (iv) image acquisition (Stekel, 2003). Figure 2 shows a summary of the steps involved in the microarray process. To summarise, mRNA is extracted from the samples of interest, for example samples from healthy and diseased subjects. The total pool of mRNA from each cell population is used to prepare fluorescently labelled cDNA. To allow for the direct comparison of the abundance of each gene in the two samples, the two samples are labelled with different dyes. The different dyes most commonly used are Cy3 and Cy5. Cy3 is excited by a green laser, whilst Cy5 is excited by a red laser. The next step is to mix the two fluorescently labelled cDNAs and hybridise with a DNA microarray in which each gene is represented as a distinct spot of

DNA. The cDNA sequences in the samples of interest then hybridise specifically with the corresponding gene sequence in the array. One may then determine the relative amount of gene expression for each gene on the array by the corresponding ratio of red to green fluorescence (Brown and Botstein, 1999; Duggan, et al., 1999).

Figure 2. Steps involved in the microarray analysis procedure
Source: Adopted from http://www.bio.davidson.edu/courses/genomics/chip/chip.html



1.5. Computational applications to the analysis and modelling of genomic and proteomic data

The continual increase in the amount of information a proteomic and/or genomic experiment can provide enables further opportunities for the extraction of biologically relevant information. Computational techniques allow for the ability to identify patterns in complex data, where a human could not. Humans generally use heuristics when making a clinical decision, so when faced with difficult and complex scenarios bias and error may result. Computational methods however have the ability to consider every possible variable and its probability, and therefore humans and computers have complimentary strengths that may be combined to surpass the abilities of either alone. There is so much data currently available that clinicians and healthcare workers face the unenviable task of deciding what may or may not be of use in a clinical setting. Therefore one of the goals of bioinformatics is to identify relevant features within these datasets which may be translated and integrated into information which can be of use (Burnside, 2005).

The two key aims which researchers tend to be most interested in when analysing new datasets is to (i) determine the relationships and patterns between the variables (for example between proteins or genes) or groups of individuals within the population, and (ii) predict the classes to which samples belong to based on their profiles. These aims can be accomplished by various computational approaches to data-mining, pattern recognition, clustering and classification of the datasets (Cox, et al., 2005). Some of the more commonly used methods will be outlined in the next sections.

1.5.1. Finding relationships and patterns: Clustering and principal components analysis

Perhaps the two most popular methods of analysing the relationships between the variables in the data, or between the patients themselves are hierarchical clustering and principal components analysis (PCA).

Hierarchical clustering is routinely used as the method of choice when analysing gene expression data (Alon, et al., 1999; Pomeroy, et al., 2002; van 't Veer, et al., 2002; Welsh, et al., 2001), and functions by arranging the profiles of samples into a tree-like structure so that the most similar profiles lie close together, and profiles very different to one-another

lie farther apart, allowing for the rapid visual assessment of patterns within the data. The methodology is based on the construction of a distance matrix which enables the two samples with the most similar profiles to be determined. These are then placed together in the tree to form a cluster, and the distance between this newly defined cluster and the remaining samples is calculated. A new cluster is then determined and this process is repeated until all of the samples have been placed in a cluster. There are various linkage methods used for calculating distance, such as single linkage, complete linkage and average linkage. Single linkage computes the distance as the distance between the two nearest points in the clusters being compared. Complete linkage computes the distance between the two farthest points, whilst average linkage averages all distances across all the points in the clusters being compared. Similarly, there are also several distance metrics which can be used to compute this value, such as Pearson correlation and Euclidean distance. Different linkage methods and methods of calculating distances often lead to very different dendrograms, so it is recommended that many methods are applied before drawing conclusions regarding the relationships in the data (Stekel, 2003). Clustering has been used in the medical field mainly for the analysis of gene expression data from cancer patients. van't Veer and colleagues (van 't Veer, et al., 2002) used hierarchical clustering to analyse primary breast tumours to identify a gene expression signature which was predictive of a 'poor prognosis signature', that is, a signature capable of predicting patients at risk of distant metastases. Here, they identified 70 genes from an initial 25,000 which could predict correctly disease outcome for 83 % of patients. Pomeroy and co-workers (Pomeroy, et al., 2002) used a similar approach when investigating tumours of the central nervous system, a group of tumours whose diagnosis is often difficult. Using hierarchical clustering, they showed that tumours could be clearly separated according to the different tumour types. Welsh and associates (Welsh, et al., 2001) used oligonucleotide microarrays of approximately 6,000 genes to identify candidate markers of epithelial ovarian cancers. They found that normal tissues were easily separated from tumour tissues, and the tumours could be further divided into groups correlating with known histological and clinical observations. Alon and colleagues (Alon, et al., 1999) used a two way cluster analysis to cluster firstly the tissues based upon tumour and normal colon cancer tissue samples, and secondly the genes where families of genes grouped together. They revealed groups of genes whose expression was correlated across different tissue types, for example, the majority of EST's homologous to ribosomal proteins clustered together. The one major problem concerning clustering is that it suffers from the curse of dimensionality when analysing complex datasets. In a high dimensional space, it is likely that for any given pair

of points within a cluster, there will exist dimensions on which these points are far apart from one another. Therefore distance functions using all input features equally may not be truly effective (Domeniconi, et al., 2004). Furthermore, clustering methods will often fail to identify coherent clusters due to the presence of many irrelevant and redundant features (Greene and Cunningham, 2005).

PCA is one of the most widely used multivariate techniques for input dimensionality reduction in datasets where the number of inputs far exceeds the number of cases. PCA transforms the input space into a new space described by what are known as principal components, which are expressed as linear combinations of the original variables. These principal components lie orthogonal to one another and are ranked according to an eigenvalue. By selecting the vectors with the largest eignevalues, the vectors which map the largest variations in the input space are determined. So, the ultimate aim of PCA is to capture those vectors, or principal components which explain the most variation in the data, thus reducing the dimensionality of the data space (Haykin, 1999). The main limitation of using PCA for proteomic and gene expression data is the inability to verify the association of a principal component vector with the known experimental variables. This makes it extremely difficult to accurately identify the importance of the proteins or genes in the system. Marengo et al. (Marengo, et al., 2004) applied PCA analysis to proteomic data generated from neuroblastoma tumour samples and identified two groups of samples in the data set. By analysing the loadings of the principal components they could identify the discriminatory variables and by following this up with mass spectrometry they identified proteins responsible for the differences occurring between healthy and diseased samples. Liu and co-workers (Liu, et al., 2005) analysed gene expression data using PCA as a dimensionality reduction tool, followed by logistic regression for classification purposes. This approach was applied to five publicly available tumour based datasets, and was able to distinguish different classes with high accuracy. The authors showed this approach to be very comparable to SVMs and ANNs when classifying gene expression data.

The limitations of the current PCA approach include an inability to verify (or deny) association of a principal component vector to the known experimental conditions and then to accurately identify the critical genes.

1.5.2. Predicting sample class

In addition to analysing relationships in the data, there are also numerous methods available which allow for the classification of samples, and are subsequently able to position new samples with unknown class membership into a particular group. Here, the ultimate aim is to identify genes or proteins (or subsets of these) which may be used either in the development of prognostic tests (to predict for example, response to therapy in a patient specific manner) or alternatively in the development of novel diagnostic tests. This is achieved by generating predictive models using the measurements across a number of variables (for example ion mass intensities or gene expression ratios) for samples whose class is known *a priori*. This is known as supervised learning. Artificial Neural Networks were used as the focal point during this thesis for generating predictive models enabling accurate sample classification, and these are explained in detail in chapter 2. Here, a summary of alternative methods will be discussed briefly, with examples of their application.

1.5.2.1. *K*-Nearest Neighbours

K-Nearest Neighbours (*K*NN) is one of the simplest methods to perform when deciding which class an unknown sample belongs to. This method compares the profiles of an unknown sample with the profiles of samples with known group membership. The class of the unknown sample is then determined to be the same as the class of the known sample to which it is most similar to. There are two parameters to consider when using this method, k and l; where k is the number of nearest samples to look at, and l is the margin of victory required in order for a class decision to be made. If this margin is not met, then the sample is unclassified. So for example, if $k = 3$ and $l = 3$, then the unknown sample would be assigned to the same class as the nearest three samples, but only providing that all three of these belong to the same class, otherwise the sample would be unclassified (Stekel, 2003). Shen et al. (Shen, et al., 2005) showed that using a *K*NN approach for predicting membrane protein types resulted in high success rates. Here they compared several predictive algorithms and the *K*NN predictor outperformed the other predictors, resulting in between 86-96 % of samples being successfully predicted. *K*NN was also the method of choice for Barrier and colleagues (Barrier, et al., 2005) for creating a classifier from gene expression measurements of Stage II and Stage III colon cancer patients. They built two classifiers which successfully predicted between 78-83 % of patients into their respective

groups. The main disadvantages of using this method are that in highly dimensional biological systems with many samples there are likely to be outliers present. As this approach is not particularly robust to outliers this may lead to incorrect classifications when such outliers are present in a given system. Additionally, this approach takes into consideration every attribute of every element when classifying a new sample. So if the target concept depends on only a few features in a highly dimensional data space the samples that are truly most similar may well be placed a large distance apart (Mitchell, 1997; Mylonas, et al., 2004).

1.5.2.2. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) calculates the straight line (or hyperplane) between two classes that best separates them. It does so by taking into consideration sample to sample variation within classes, as to minimise within class variance, and maximise between class variance. The class of any unknown samples is then determined simply by the side of the hyperplane it lies. Because LDA takes into account the variation within the sample population, this method sometimes performs better than other linear based approaches at classifying unknown samples, however, this approach does not extend naturally to data that is not linearly separable, so should be avoided with such datasets (Stekel, 2003). This approach was used with success by Xu and colleagues (Xu, et al., 2006) who aimed to detect serum proteomic patterns by using SELDI-TOF MS technology applied to the staging of colorectal cancer patients. They created several models and identified potential biomarker ions which could discriminate between disease stage with accuracies between 75-87 %. Gao and co-workers (Gao, et al., 2005) also utilised LDA where sera from patients with lung cancer and healthy controls were subjected to antibody microarray analysis. Here, it was concluded that although LDA correctly classified 17/24 of cancer patients correctly, and 52/56 of the control samples correctly, it did not perform any better than results obtained with other classifiers.

1.5.2.3. Support Vector Machines

Support Vector Machines (SVMs) function in a manner similar to LDA, in that they work by separating the data into two regions by constructing a straight line or hyperplane. The advantage SVMs have over other linear separators is that the data are first projected into a higher dimensional space (by a kernel function) before being separated by a linear method,

which allows for discrimination of nonlinear regions of space, and therefore separation of nonlinear data. The class of the unknown sample is then determined by the side of the hyperplane on which it lies (Crisianini and Shawe-Taylor, 2000). SVMs are a popular classification tool within the machine learning community, and there uses are widely documented (Chu and Wang, 2005; Eszlinger, et al., 2006; Wagner, et al., 2004; Warnat, et al., 2005; Yu, et al., 2005). They have been used with success in the classification of a wide range of biological data, for example 13 peaks were identified from a four group classification problem involving mass spectrometry data derived from prostate cancer patients. These peaks allowed for the correct identification of 87 % of samples (Wagner, et al., 2004). Meanwhile, Warnat et al. (Warnat, et al., 2005) analysed sets of gene expression data from leukaemia patients, and found the SVM based classifiers could predict in excess of 85 % of samples based upon training and testing cases being randomly chosen by cross validation. Eszlinger et al. (Eszlinger, et al., 2006) used a SVM based approach when analysing gene expression data derived from thyroid carcinoma tissue. They identified multiple gene markers which, when used as a classifier, produced high prediction accuracies. Similarly, Chu and Wang (Chu and Wang, 2005) applied SVMs for cancer classification of microarrays to subsets of genes identified by a t-test based gene selection approach. They successfully built classifiers capable of high classification accuracies with fewer genes than previously reported on the datasets used. Yu et al. (Yu, et al., 2005) developed strategies for pre-processing of SELDI-TOF MS datasets followed by SVM classification analysis. The models they generated led to sensitivities of 97 % and specificities of 93 %, again showing the potential for SVM in the analysis of this type of data. The major disadvantages associated with SVMs are that they are affected by speed and size, both in training and testing, and can be extremely slow when in the test phase (Burges, 1998). Furthermore from a practical point of view, for large scale tasks extensive memory is required due to the high complexity of the data (Burges, 1998; Osuna and Girosi, 1999).

1.6. Artificial Neural Networks

1.6.1. Introduction to Artificial Neural Networks

An ANN is an adaptive, non linear form of artificial intelligence inspired by the way the human brain learns and processes information. The most important facet of this paradigm is the fact that it is built from a potentially large number of interconnected processing elements, which work together to solve specific problems. A popular form of ANN is the multi-layer perceptron (MLP) and is used to solve many types of problems such as pattern recognition and classification, function approximation, prediction and forecasting. They learn in a fashion that is analogous to the way learning in the human brain is carried out, that is, by example. In humans, learning involves minor adjustments being made to the synaptic connections between neurons, in ANNs, learning is achieved by updating the weights that exist between the processing elements that constitute the network topology.

1.6.2. Historical background

ANN modelling began with the work of McCulloch and Pitts in 1943 (McCulloch and Pitts, 1943) who in an effort to understand how the nervous system might be functioning, introduced computing elements which were based on the properties of neurons and their connections. McCulloch and Pitts listed several assumptions reflecting the operation of neurons, and these assumptions described what became to be known as the 'McCulloch-Pitts' neuron. The assumptions can be summarised as follows; Neurons are binary devices, so can only be one of two possible states, an "all-or-none" process. Each neuron has a fixed threshold, and the structure of the net does not change with time. The neuron can receive inputs from excitatory synapses, all of which have identical weights. It can also receive inputs from inhibitory synapses, which absolutely prevent excitation of the neuron at that time, so if an inhibitory synapse is active, the neuron cannot turn on. In summary, the neuron responds to the activity of its synapses, and if no inhibitory synapses are active, the neuron then sums its synaptic inputs to determine whether it meets or exceeds its threshold. If it does, the neuron can then become active, if not, the neuron is inactive. The results of this model were simple logic threshold functions, such as INCLUSIVE OR, where a neuron would become active only if *a* OR *b* OR BOTH *a* AND *b* are active.

In 1958, Rosenblatt (Rosenblatt, 1958) then introduced the perceptron, which was the first computationally orientated neural network. He used a single layer of model neurons projecting to another layer of neurons by way of parallel connections. Learning was achieved by modifying the units or their connections in such a way that stimuli of a certain class would tend to encourage a stronger response in one unit from the others.

Widrow and Hoff (Widrow and Hoff, 1960) proposed an adaptive system related to the perceptron which could learn more quickly and accurately. This "adaptive neuron" calculated a weighted sum of activities on the inputs, multiplied by the synaptic weights. Rather than using the usual values of 1 or 0, if this sum was greater than a certain threshold, the output was +1, and -1 otherwise. They described this adaptive system of consisting of a "worker" and a "boss". The job of the worker was to predict, and the boss had the job of adjusting the worker. Essentially this adaptive system was able to form an error signal between the actual and computed outputs, and then was able to adjust the synaptic weights so that the error signal was zero. The system would learn something from each pattern and accordingly experience a design change, storing this "knowledge" in the values of the weights. Learning would continue until the search process converged and the error was exactly zero, with an output value of either +1 or -1, and not just a correct classification, where the sum of weights would merely have to be a value greater than the threshold. An important feature of this adaptive system was the existence of the worker and boss, which contained previous knowledge regarding the correct class the input pattern corresponded to. These algorithms are known as supervised, and presently back-propagation (BP) is the most commonly used supervised algorithm, and is basically an adaptation of the Widrow-Hoff rule.

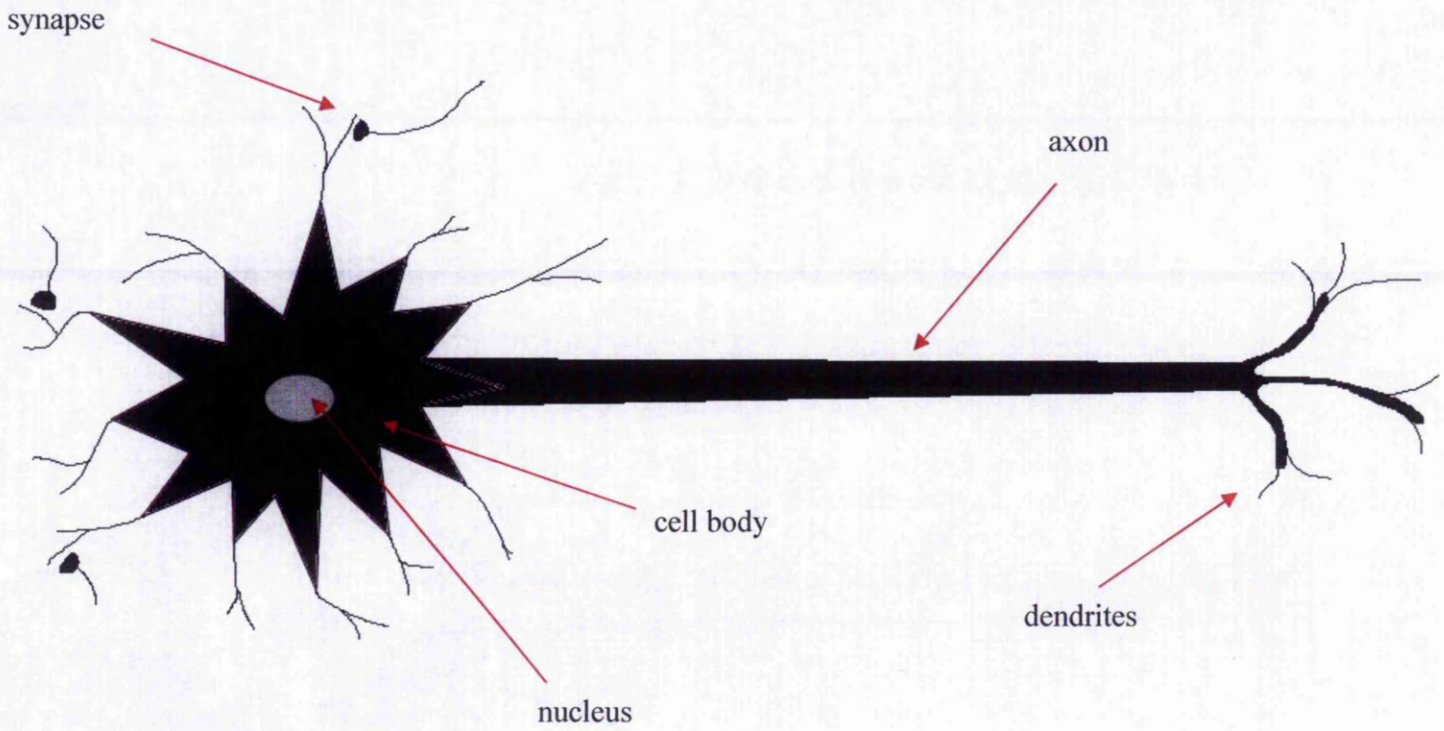
In the late 1960's, Minsky and Papert (Minsky and Papert, 1969) published a book in which they discussed the major limitations associated with perceptrons, and this led to a major setback in the use of neural networks in research. Some of these limitations were already known, but Minsky and Papert showed the importance and magnitude of these limits, such as the requirement for linear seperability of the data points. The most serious impact on future neural network research was the fact that the authors believed that the limitations that they had discussed would also be extended to variants of the perceptron, such as multilayered systems. To quote their statement, "...our intuitive judgement that the extension to multilayer systems is sterile".

When Rumelhart, Hinton, and Williams described the BP algorithm (Rumelhart, et al., 1986) interest was renewed in the use of neural networks, and this was to become the most popular learning algorithm for research with multilayer networks. BP is an adaptation of the Widrow-Hoff rule of error connection (Widrow and Hoff, 1960) and involves running the synapses backwards, so that the internal (hidden) unit(s) knows how large the error is, and can then modify the weights according to the strengths of the inputs received. So briefly, the algorithm involves a forward pass through the layers of the network in order to estimate error, and then a backward pass allowing for the modification of the synapses so that the error is decreased.

1.6.3. Biological neural networks

The basic structure and function of the human neurone will now be briefly discussed in order to understand the analogy between this and the way in which ANNs function. Neurons are the cells with the ability to learn and process information, and are primarily located in the brain. Figure 3 shows a schematic of a biological neuron, and shows that the neuron is mainly composed of a cell body, known as the soma, dendrites and an axon. The cell nucleus has a cell body which contains the information regarding hereditary traits, and a plasma containing the molecular equipment for the production of materials needed by the neuron. Neurons receive signals from other neurons by way of dendrites, and transmit these along the axon. The axon branches into several strands, and at the end of these strands are the synapses, which is a place of contact between two neurons. This contact occurs between the axon strand of one neuron and a dendrite strand of another, and upon this contact, chemicals known as neurotransmitters are released (Jain, et al., 1996). These neurotransmitters diffuse into the dendrites of neighbouring neurons forcing them to generate a new electrical signal, which then passes through the second neuron. The amount of signal passing through a receiving neuron is dependent upon the intensity of signal originating from the feeding neuron, and these signals may either excite or inhibit the firing of the neuron (Basheer and Hajmeer, 2000). The effectiveness of a synapse can be adjusted by the signals passing through it so that they learn from the activities which they contribute to. It is this dependence on learning from previous activities which acts as a memory and the ability of the human brain to remember (Jain, et al., 1996). This mechanism of signal transfer constitutes the vital steps involved in early neurocomputing development, as they both learn by iteratively adjusting the magnitudes of the weights or synaptic strengths.

Figure 3. Schematic of a human biological neuron



1.6.4. Artificial neurons: Network structure and architecture

The simplest forms of ANNs contained two layers of neurons, or nodes. This constituted of an input layer connected to an output layer. The input layer of the network represents the variables in the data to be modelled, with an input node assigned to each variable. The output layer computes an output based upon the information it has received from the input layer.

MLP ANNs are most commonly composed of three layers. Firstly an input layer, which is connected to a hidden layer, that is in turn connected to output layer. The hidden nodes contained within the hidden layer represent the features from the input data, and these features are highlighted or subdued by the activities of the weights between the input and hidden units. The output unit and the output it computes is then dependant upon the information it receives from the weights between the hidden and output units.

1.6.5. The perceptron

Artificial neurons mimic biological neurons in that the connections between nodes represent the axons and dendrites, the connection weights represent the synapses, and the thresholds corresponds to the activity present in the soma (Jain, et al., 1996). The most primitive form of ANN was based on a unit known as a perceptron. (Figure 4). A perceptron functions by taking a vector of inputs, calculating a linear combination of these inputs, and then finally outputting a 1 if the result exceeds some threshold or a -1 if it does not (Mitchell, 1997). Given inputs x_i through x_n, the output $o(x_{i\ldots\ldots}x_n)$ computed by the perceptron is:

$$o(x_1\ldots..xn) = \begin{cases} +1 \text{ if } w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n > 0 \\ -1 \text{ otherwise} \end{cases}$$

Where each w_i is a weight that determines the contribution of input x_i to the output. w_0 is a threshold that the weighted combination of inputs $w_1x_1+\ldots+w_nx_n$ must be greater than in order for the perceptron to output a 1.

The perceptron can be viewed as representing a decision surface in an *n*-dimensional space of individual instances. The perceptron outputs a 1 for instances lying on one side of the hyperplane and outputs a -1 for instances lying on the other side. Figure 5a shows an examples of a linear decision surface, and those instances that can be separated by this hyperplane are called linearly separable sets of examples (Mitchell, 1997).

1.6.6. Advancing beyond simple neurons: The Multi-Layer Perceptron

As discussed earlier, single layer perceptrons are only capable of solving problems of a linear nature. However, because the weights of the interconnecting nodes are set in response to comparisons between the network and actual output, they are not adjustable during training. As such, perceptrons show an inability to model non- linear problems. Therefore in order to search a nonlinear hypothesis space, such as that shown in Figure 5b, a MLP is needed. Unlike the perceptron, this contains one or more hidden layers, each comprising of a number of hidden nodes. Figure 6 shows the structure of a MLP. A commonly used function used within these hidden nodes is the sigmoidal activation function, which computes a linear combination of its inputs, and then applies a threshold to the result. Thus the output unit is a nonlinear function of its inputs (Mitchell, 1997). The sigmoid unit computes its output *o* as:

$$o = \frac{1}{1+e^{-x}}$$

This is often referred to as the sigmoid, or logistic function. In this function, the output ranges between 0 and 1, and is sometimes called a squashing function because of its capability of mapping a very large input set into a small output range. In addition to the sigmoid, other activation functions may be used. For example the hyperbolic tangent function (tanh) is similar to the logistic function, but the output range lies between -1 and +1. The output *o* for this function is defined by the following:

$$o = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Figure 4. The Perceptron

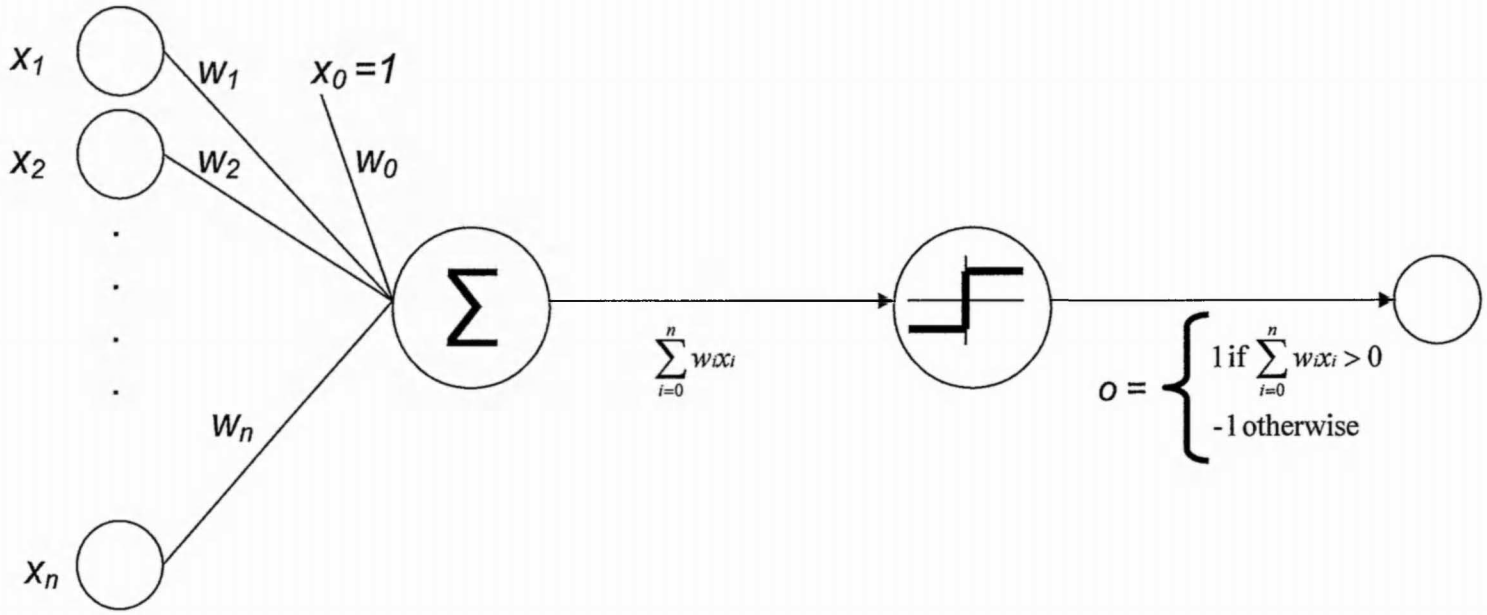


Figure 5. Examples of (a) linear and (b) curved decision boundaries

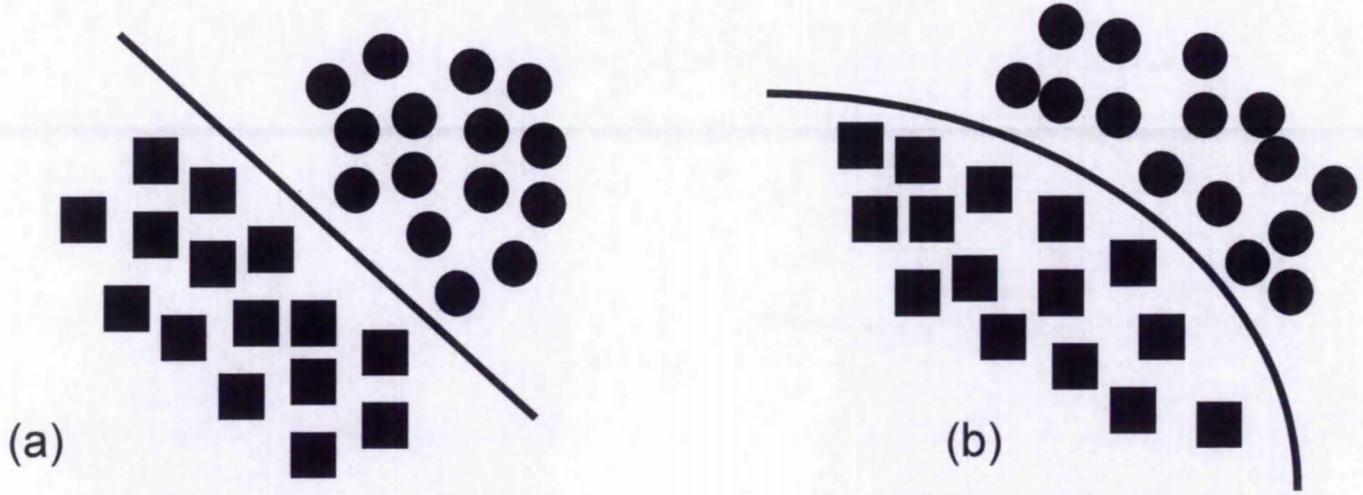
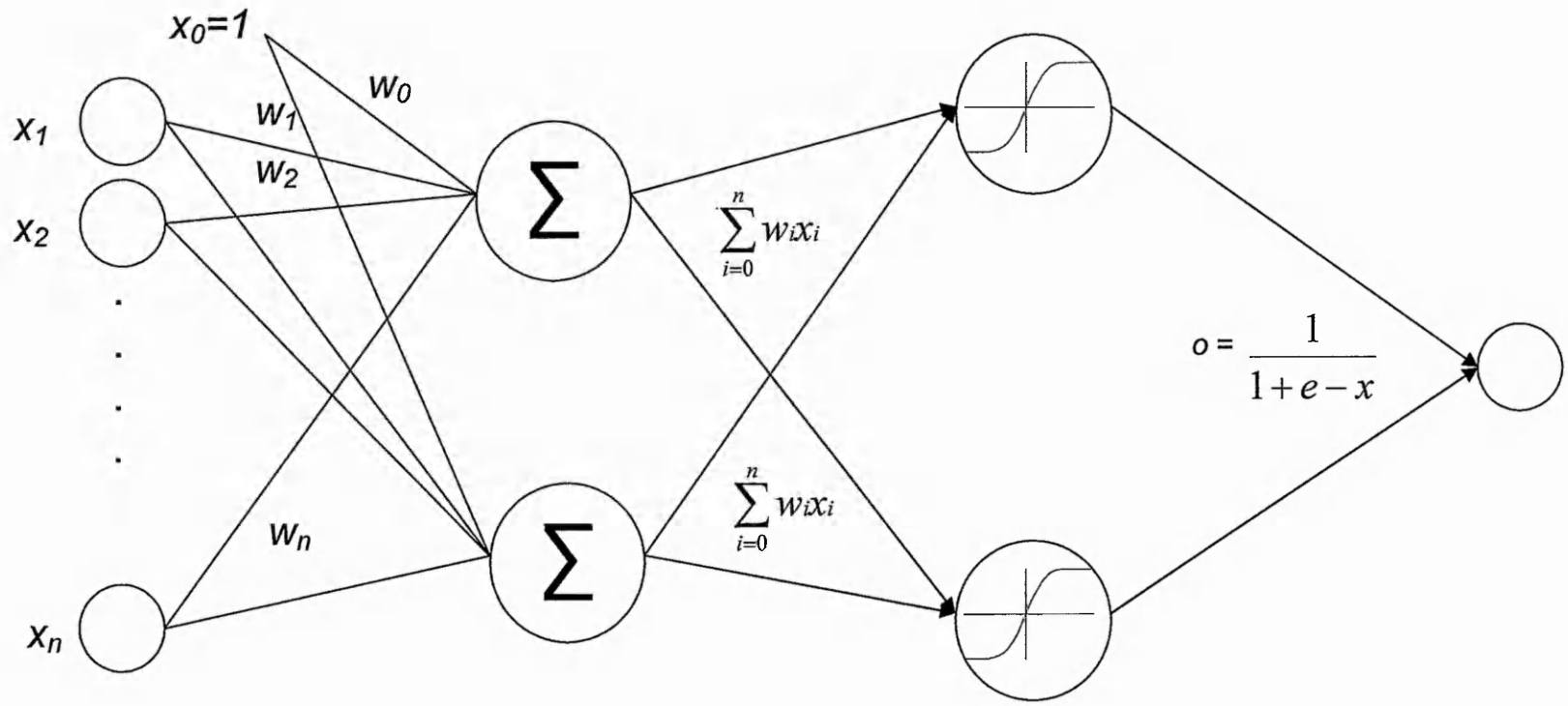


Figure 6. The Multi Layer Perceptron with sigmoidal activation function.



1.6.7. Learning

1.6.7.1. Perceptron training

The learning goal in a single perceptron is to determine a weight vector that causes the perceptron to produce the correct +1, -1 output for each of the given training examples. In order to learn an acceptable weight vector the network may begin with random initial weights, then apply the perceptron to each training example, iteratively modifying these weights whenever an example is misclassified (Mitchell, 1997). This process is repeated, iterating through the training examples until all are correctly classified. Weights are modified at each step according to the perceptron learning rule, which updates the weight w_i associated with input x_i according to:

$$w_i = n(t - o)x_i$$

Here t is the target output for the current training example, o the output generated by the perceptron, and n is a constant known as the learning rate (Mitchell, 1997). The weight change of a neuron is proportional to the influence an input had on the error during training and the learning rate is a constant which controls the size of these weight changes. Each time a pattern is presented to the network, the weights leading to an output node are modified slightly during learning in the direction required to result in a smaller error the next time the same pattern is presented. The larger the learning rate, the larger the weight changes, and the faster the learning will proceed. If the learning rate is too small, training will be slowed down, however, oscillation or non-convergence can occur if the learning rate is too large (Basheer and Hajmeer, 2000).

1.6.7.2. Gradient descent and the delta rule

The perceptron rule discussed above is designed to converge when the training examples are linearly separable, however it may fail if they are not. To overcome this, an alternative rule known as the delta rule may be used. This rule works on the principle that if the training examples are not linearly separable the rule will converge towards a best-fit approximation of the target values. The main principle behind this rule is to use gradient descent to search a hypothesis space of possible weight vectors to find the weights that best

fit the training examples (Mitchell, 1997), and will converge regardless of whether the training data are linearly separable. This rule is important because it forms the basis for the BP algorithm which will be discussed in the next section.

1.6.7.3. The Back-Propagation algorithm

The BP algorithm learns the weights for a network consisting of multiple layers, and utilises gradient descent to minimise the error between the network output, and the target output value. The goal of learning in BP is to use gradient descent to search a hypothesis space in order to find the optimal weight values for all inputs in the network which minimise the error. Unlike in single layered networks, with an error surface containing a single minimum, multilayer networks may face error surfaces containing multiple local minima. As such, these are only guaranteed to converge towards some local minimum error, which may not necessarily be the global minimum (Mitchell, 1997). The algorithm for a feedforward network, in which all units present in each layer are connected to all units in the next layer, can be summarised as follows:

1. Construct a network with the desired number of hidden and output units.
2. Initialise all network weights to small random numbers.
3. Propagate the input forward through the network.
4. Propagate the errors backward through the network.
5. The network is applied for each training example, and the error of each network output unit is calculated, together with the error of each hidden unit. The gradient with respect to the error for this example is computed, and the weights are updated according to:

$$\eta \delta_j x_{ji}$$

6. Repeat this gradient descent step using the same training examples until the network performs to an acceptable level.

The weight update rule used is similar to the delta training rule. Each weight is updated in proportion to (i) the learning rate η, (ii) the input value to which the weight is applied x_{ji}, and (iii) the error in the output of the unit δ_j. The difference between the two update rules is that the error term used in BP (δ_j) is more complex than that used in the delta rule ($t - o$)

as it is multiplied by a factor which is the derivative of the sigmoid squashing function. In addition to the error being calculated for each output unit, the error term must also be calculated for each hidden unit. This is achieved by summing the error terms for each output unit δ_k that is influenced by the hidden unit h, weighting each δ_k by the weight w_{kh}, from hidden unit h to output unit k. This essentially defines the influence each hidden unit has on the output(s) it feeds into.

The algorithm updates weights in an iterative manner, following the presentation of each training example (Mitchell, 1997). Training is typically performed in an iterative fashion and is terminated with respect to a given condition, for example a pre-determined number of iterations, or once the error is below a pre-determined threshold for a separate validation set of training examples. During training, different cost functions may be used such as the sum of squared errors for regression problems, and entropy based methods for classification.

1.6.8. Advancing the Back Propagation algorithm

1.6.8.1. Escaping local minima with momentum

One of the most common variations of the BP algorithm is to make a slight alteration to the weight update rule by making the weight update on the nth iteration depend on the update that occurred during the (n-1)th iteration. This is achieved by using a momentum term, and this can help in the prevention of the network from becoming trapped in local minima, or being stuck along flat regions in error space. The momentum term helps to speed up the time it takes for the network to reach convergence by gradually increasing the step size of the search in regions where the gradient is not changing. In addition to this, when all weight changes are in the same direction, the momentum term amplifies the effect of the learning rate, again leading to faster convergence. The weight update with momentum occurs as follows:

$$W_{ji(n)} = \eta \delta_j x_{ji} + \alpha W_{ji(n-1)}$$

Where $w_{ji(n)}$ is the weight update that is performed at the current (nth) iteration in the algorithm, and α is the momentum term.

1.6.8.2. Generalisation and avoiding the problem of overfitting

A common problem when training with ANNs is ensuring that they are capable of generalising to future cases, by providing an estimation as to their likely performance on new data. Given the fact that with nonlinear modelling one may not have prior information regarding the complexity of the data, it is essential to estimate the performance of these models on new data, in order to be confident that overfitting has been avoided. As briefly mentioned previously, the BP algorithm should stop training once the network has achieved an acceptable error. However, the question remains as to what is considered to be an acceptable level, and what can be done to ensure that the model will be capable of generalising to additional future cases. If training is terminated solely on the basis of a set number of iterations, then the model is at risk of overfitting, which is to all intent and purposes a memorisation of the training data.

The most universal approach to address this problem is resampling. Typically in resampling approaches, the dataset is split into different subsets, training and test. The neural network is trained and optimised using the training set, whilst the network error is monitored with respect to the test subset. Subsequently the trained weights which produced the lowest error over the test data split are used in the final model. The training is stopped once the error has not improved on this test subset for a predetermined number of iterations, known as early stopping. This helps to avoid over-training and therefore improves the ability of the model to generalise well to new data. This can be enhanced further by splitting the data into three subsets as opposed to just two; these are known as training, test and validation. Training performance is monitored as in the previous method, but here the network is further validated once the model has been trained using the validation data split, which gives an unbiased estimation of the networks likely performance on future cases.

A number of approaches to validation are commonly used, such as Monte Carlo resampling, bootstrapping, *k*-fold validation, and random sample cross validation (RSCV). Monte Carlo resampling is perhaps the simplest method, where a training, test and validation set are selected at random, with an equal number of cases in each subset. Alternatively the validation subset may be kept constant, with the training and test sets drawn at random, to enable comparison between models for validation data (Bishop, 1995). Bootstrapping has been shown to be an effective measure of estimating the error of

predictive values in neural network models, and therefore is a reliable approach in determining generalisation of the network (Tibshirani, 1996). In bootstrapping, subsamples of the data are analysed, where many "pseudo-replicates" are created by resampling the original data. Here, cases are drawn at random from the data set, with equal probability, in order to replicate the process of sampling multiple datasets. *k*-fold validation is an effective approach when the number of samples is not efficient enough to split the data into three subsets. A widely used version of this is called leave one out cross validation (Braga-Neto and Dougherty, 2005; Hu, et al., 2005), where *N* divisions are made (where *N* is the total number of cases in the dataset) and in each division the network is trained on all of the samples except one, which is set aside for test purposes. This process is repeated so that all of the samples are used once for testing. Finally, in RSCV, the training, test and validation data splits are randomised a number of times, so that each sample is represented in the validation split on numerous occasions, enabling confidence to be determined for the predictions on blind (validation) data.

Other methods are also available to address the overfitting problem, one example being weight decay regularisation. Here, the error function includes a penalty term, for example the sum of squared weights multiplied by a decay constant. This approach focuses on penalising large weights, in order to keep weight values smaller than they naturally otherwise would converge at (Bishop, 1995). The rationale being that large weights can affect generalisation by causing excessive variance in the output, leading to output values far exceeding the range of the data (Geman, et al., 1992).

1.6.9. Advantages and disadvantages of Artificial Neural Networks

As ANNs are loosely based on the way a biological neuron is believed to organise and process information, they have many advantages in their ability to derive meaning from large complex datasets. Firstly they are essentially nonlinear so are able to process data containing complex interactions that are therefore usually difficult or too complex to interpret by conventional methods. Another advantage is that they are fault tolerant, *i.e.* they have the ability of handling noisy or fuzzy information, whilst also being able to tolerate data which is incomplete or contains missing values. In addition to this they are capable of generalisation, so they can interpret information which is different to that of the training data, thus representing a "real-world" solution to a given problem by their ability to predict future cases or trends based on what they have previously seen. Thus, trained

ANNs can be used as stand alone executable systems used to predict the class of an unknown case of interest. Finally, the importance of individual variables in the trained ANN model can be easily recovered using various methods such as the analysis of interconnecting network weights and sensitivity analysis, which from a biological perspective is perhaps one of the most useful aspects of ANN modelling, and will be discussed in greater detail in later chapters.

Unfortunately, ANNs also have their limitations. Training of ANNs can potentially be time consuming depending on the complexity of the data being modelled, and as the number of hidden layers required to capture the features within the data increases, so does the time taken for training to complete. Additionally, training is not guaranteed to converge, and only partial solutions to the problem may be reached. This is usually due to the ANN being stuck in local minima along the error surface, which leads to a failure of the model to reach the global minima and failure of the model to generalise well. This problem of failing to generalise to new cases is also associated with overtraining, where the model essentially memorises the training data and any noise associated with it. Figure 7 shows a hypothetical error surface, and the problem of being stuck in local minima and overtraining. ANNs also need to be trained on as large a dataset as possible because when predicting future cases they cannot extrapolate particularly well to those which are widely outside of the range of initial training. The one major barrier which researchers usually associate with ANNs is that the ability to interpret the contributions of input variables and how they reach a conclusion is often perceived to be difficult, and because of this they are sometimes referred to as "black boxes" " (Duh, et al., 1998; Smith, et al., 2003; Tung, et al., 2004; Wall, et al., 2003).

1.6.10. Artificial Neural Networks in medical practice

ANNs have enjoyed a rapid increase in popularity in the medical arena since the early 1990's, with the number of publications cited in PubMed showing an almost exponential rise. A similar trend is seen in the field of cancer. In 1990, no papers were published detailing the use of ANNs in cancer research, whilst in 2005, this had risen to 87, more than one per week. The largest area of research with ANNs appears to be focused towards breast cancer, which amasses approximately 25 % of all published papers concerning cancer research and neural networks each year, followed by prostate cancer with around 15 %. Figure 8 shows the annual publications cited in PubMed by performing a search with

firstly the keyword terms "neural networks" and secondly, "neural networks" and "cancer". Lisboa and Taktak (Lisboa and Taktak, 2006) conducted a review assessing where ANNs had been involved in an actual clinical setting in the field of cancer. The authors found that of all the studies using ANNs in cancer research, only 27 were involved in clinical trials or randomised controlled trials. Of these, 21 showed an increase in benefit to healthcare provision and the remaining 6 did not. Perhaps the most widely known application of neural networks in medicine is the PAPNET system used in the automated cytological screening of cervical smears (Boon, et al., 1994). This system aids in the assisted screening of Pap smears, which examine cells taken from the uterine cervix for signs of precancerous and cancerous changes. As it is extremely difficult to detect all cases of early cancer with the human eye, abnormal Pap smears were missed. The PAPNET system showed to be able to reduce the numbers of false-negative cases, as such increasing the efficiency of screening and detection of pre-cancerous and cancerous cells in the cervix. Examples of some recent applications of ANNs, not necessarily limited to cancer studies include; detecting coronary artery disease (Allison, et al., 2005), the diagnosis of lung disease (Abe, et al., 2004), developing diagnostic and prognostic models in various cancers from proteomic (Chen, et al., 2004; Guo, et al., 2005; Hu, et al., 2005; Hu, et al., 2005; Yu, et al., 2004; Yu, et al., 2005) and genomic (Ando, et al., 2003; Berrar, et al., 2003; Bicciato, et al., 2003; Narayanan, et al., 2004; O'Neill and Song, 2003) data, the detection of subclinical brain electrical activity changes in Huntington's disease (de Tommaso, et al., 2003) and the discrimination of patients with Type 2 diabetes from healthy controls (Yang, et al., 2004).

Figure 7. Hypothetical error surface. This highlights the problems associated with local minima and overtraining.

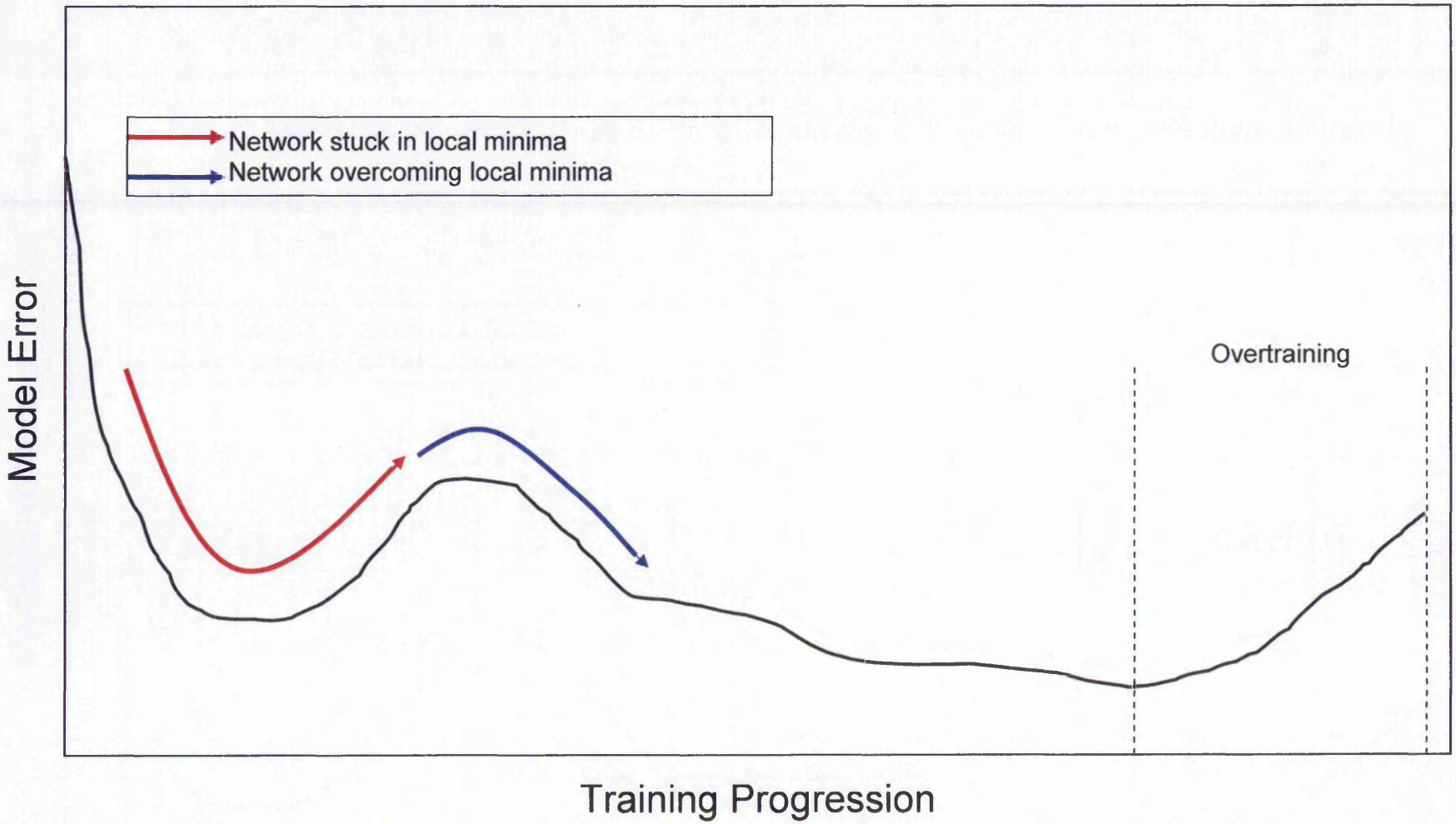
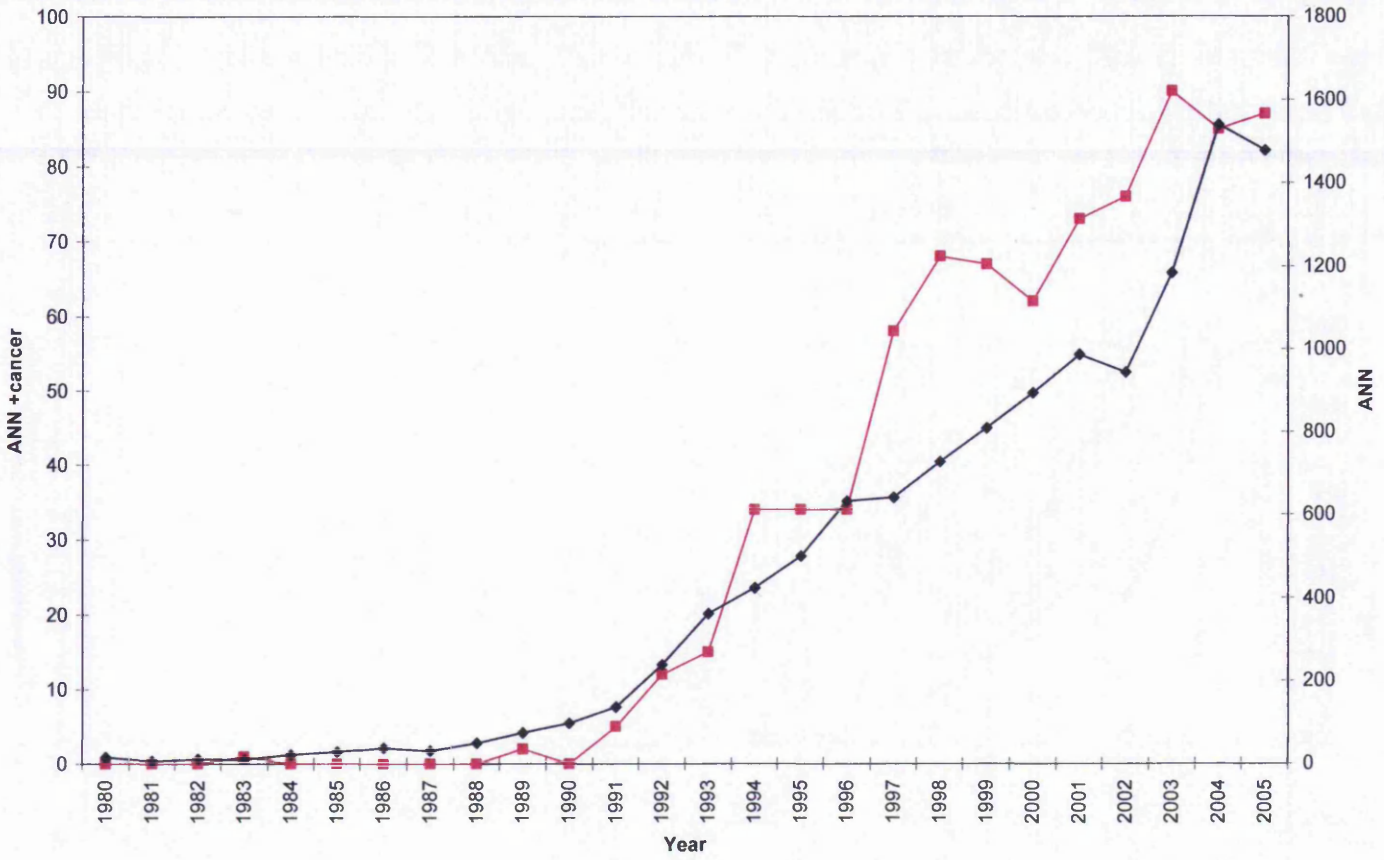


Figure 8. The annual publications cited in PubMed. This was based upon a search for the term "Neural Networks" and then "Neural Networks"+"Cancer". Blue line shows publications citing Neural Networks, whilst pink line indicates Neural Networks and Cancer publications.



1.6.11 Other Artificial Neural Network architectures and algorithms

Adaptive Resonance Theory (ART) neural networks, first described by Carpenter and Grossberg (Carpenter and Grossberg, 1988), perform unsupervised learning. The structure of the ART model consists of an input processing field, a clustering field, and a reset subsystem. There are two sets of connections (each with its own weights) between each node in each layer. The connection weights between the layers can be modified according to different learning rules. The node in the clustering layer with the largest net input becomes the candidate to learn the input pattern. Whether this candidate will learn the input pattern is dependant upon the reset mechanism, which controls the degree of similarity between the patterns placed in the same node (Cao and Wu, 2002). During this learning method, learning is fast and is guaranteed to converge in 3 passes of any set of patterns (Palmer-Brown, et al., 2003).

Kohonen networks (Kohonen, 1989), also known as self organising feature maps, consist of just two layers, an input layer and an output layer. The output layer of these networks may be two-dimensional (Ultsch and Roske, 2002), so that these networks may be used to map a three-dimensional surface onto a two-dimensional map (Barlow, 1995). The training patterns are presented to the input layer, then propagated to the output layer and evaluated, with one output neuron being labelled as the "winner". The network weights are adjusted during training and this process is repeated for all patterns for a pre-determined number of epochs, forming clusters within the data (Ward Systems Group, 1993). These networks are unique in that they autonomously self-organise themselves and converge into a stable structure representing the information that has been learnt (Nour and Madey, 1996).

Radial Basis Function (RBF) neural networks were first introduced by Moody and Darkin (Moody and Darkin, 1989). RBF networks are linear in their parameters, therefore once suitable basis function parameters have been chosen, they can be trained using a fast linear supervised training scheme. The most common nonlinear function used in RBF networks is the Gaussian function (Cowper, et al., 2002), and training in RBF networks is essentially a two stage process. In the first stage, the parameters governing the basis functions (hidden units) are determined using unsupervised methods. The second stage of training involves the determination of the weights of the final layer, which requires the solution of a linear problem, and is therefore a fast process (Bishop, 1995).

The Hopfield neural network model was first described by Hopfield (Hopfield, 1982), and like all neural networks, comprises of a set of nodes and their connections. Hopfield networks have been described as "general content-addressable memories", because they can be trained to recall a unique pre-determined state when they are presented with information associated with that state. This enables them to be trained to distinguish between two closely related states (Pritchard and Dufton, 2000). Hopfield networks in their original form, suffer from being a gradient descent technique incapable of escaping local minima, and are further limited by the fact that their penalty parameter approach for solving optimisation problems often results in poor quality solutions (Smith, et al., 2003).

A Recurrent Neural Network (RNN) is a neural network with feedback connections. From training examples, RNNs can learn to map input sequences to output sequences. In principle, they can implement almost arbitrary sequential behaviour. A recurrent network may respond to the same input pattern differently at different times, depending upon the patterns that have been presented as inputs just previously. Thus, the sequence of the patterns is as important as the input pattern itself. The patterns must always be presented in the same order, therefore random selection is not allowed. An extra slab is present in the input layer that is connected to the hidden layer just like the other input slab. This extra slab holds the contents of one of the layers as it existed when the previous pattern was trained. In this way the network sees previous knowledge it had about previous inputs. This extra slab is sometimes called the network's "long-term" memory (Ward Systems Group, 1993).

1.6.12. Comparison of Artificial Neural Networks with other methods

There have been a number of studies comparing ANNs with other statistical and computational approaches to data analysis. Some of these will now be reviewed, outlining how ANNs have performed compared to other methods when applied to biological data. Song and co-workers (Song, et al., 2004) compared various machine learning techniques to more classical statistical approaches in the prediction of outcome in two datasets. They used ANNs (single and multi-layered), logistic regression, least squares linear separation and SVMs to determine the risk of death in a population of patients with cardiac problems. They found the multi layered ANN to be consistently better than the other approaches, suggesting that the ability of the ANN to model nonlinear data was providing additional information regarding the datasets leading to higher predictive capabilities.

(Chernushevich, et al., 2001). Dreiseitl et al. (Dreiseitl, et al., 2001) compared the ability of *K*NN, logistic regression, ANNs, decision trees, and SVMs in classification of skin lesion data. The authors found logistic regression, ANNs and SVMs to give almost identical results, with *k*-nearest neighbours and decision trees performing the worst. Interestingly, even the worst of the five methods (decision trees) achieved sensitivity and specificity values comparable to human experts indicating these approaches may be of use to assist human decisions in the medical arena. On the contrary to this, Delen *et al.* (2005) found decision trees to be the best predictors when compared to ANNs and logistic regression when building classifiers to predict breast cancer survival. Decision trees predicted with 93.6 % accuracy, whilst ANNs and logistic regression performed with accuracies of 91.2 and 89.2 % respectively. Eftekhar and colleagues (Eftekhar, et al., 2005) made a comparison between ANNs and logistic regression models to study patients with head injury trauma. They built 1000 models and compared them using receiver operating characteristic (ROC) curves and T-tests. It was reported that ANNs significantly outperformed the logistic models in discrimination and calibration (goodness of fit) in 77.8 % of cases but underperformed in 68 % of cases when comparing model accuracies (using a 0.5 cut-off value). The authors concluded that further model validation was required with larger datasets to conclusively determine which approach out-performs the other. In 2001, Sargent (Sargent, 2001) carried out a review on 28 cases comparing ANNs with other statistical approaches when applied to medium and large data sets with more than 200 cases. ANNs outperformed regression in 36 % of the studies, and was outperformed in 14 % of the studies, with the results being similar in the remaining cases. The author stated that the results were inconclusive, and that both ANNs and regression should continue to be used and explored in a complementary manner. Thus, to summarise, it appears from the published literature that there does not seem to be any clear indication of which method is best when examining medical data, and that researchers should continue to experiment with whatever they are most knowledgeable about, or feel most confident about retrieving the maximum amount of information possible from the data of interest.

CHAPTER 2. ARTIFICIAL NEURAL NETWORKS IN PREDICTIVE MODELLING

2.1. Development of ANN protocols for biological data analysis

The careful adjustment of the particular parameters to be used when developing an ANN model is a key issue in order to create models which are capable of generalisation. There are various parameters which need to be considered before training of the ANN begins. These will now be discussed in more detail and are as follows; (i) partitioning of data, (ii) initial values of interconnecting weights, (iii) learning rate, (iv) momentum value, (v) training convergence criteria and (vi) the hidden layer size. Prior to the main analyses conducted in this thesis, parameter optimisation was carried out on three datasets from three different sources all using different analysis procedures. These were firstly a SELDI-TOF MS dataset generated from bacterial pathogens, secondly a MALDI-TOF MS dataset generated from control patients and those suffering with Stage IV melanoma, and thirdly a microarray dataset derived from gene expression analysis of patients with breast cancer. Some of these results will also be discussed in this section.

2.1.1 Data Partitioning

The quality of the results obtained using ANN models is highly dependent on the quality (and to a certain extent the size) of the data set used in model building, and therefore data which is used in the training process should be large enough to be representative of any variation contained within the real problem as a whole. During the development of ANN models, it is generally accepted that the data be split into different subsets, usually consisting of a training set, a test set, and a validation set. The training data should constitute the majority of the data and should therefore be representative of the system as a whole as this data set is used during the learning process to update the weights in the network. During training, the ANN model is trained with the training data, and continually optimised against the test data set so that the network performance can be assessed by using the error from predicted values assigned to these test samples. The validation data set is used to assess the ANN model performance on unseen data once the model has been developed. Various data splits have been proposed by several groups. For example, Bourquin et al. (Bourquin, et al., 1998) used a data split of 68 samples for training, 12 for test purposes and 22 for validation, which is approximately a ratio of 70:20:10. Manel et

al. (Manel, et al., 1999) used 80 % of the data for training and the remaining 20 % for test. Khan et al. (Khan, et al., 2001) and Jerez-Aragnes et al. (Jerez-Aragones, et al., 2003) preferred to split the data into 67 % for training and 33 % for validation. So it appears that some groups prefer two, whilst others use three different data splits. During this study partitioning the data into approximately 60 % training, 20 % test and 20 % validation appeared to be most effective in generating generalised models for new previously unseen data. In addition, if any new samples are introduced, these can be built into the model and used for second order validation purposes.

2.1.2. Randomisation of initial network weights

The neural network BP training algorithm begins by initialising the weight values in the network, which is usually done by randomly assigning values to them. Therefore a suitable choice of ranges for these weight values is potentially important in allowing the network to train to produce an appropriate set of weights, and thus may lead to improvements in the quality of training. Initial weight values should be small so that the sigmoidal transfer functions are not driven into saturation regions where a flat error surface would result, however if these are too small the sigmoidal activation functions will be approximately linear (Bishop, 1995). Studies by Kim and Park (Kim and Park, 2001) reveal that for their data, model performance was constantly degraded with an increase in initial weight distribution, whilst Kang-Ning and Perez (Kang-Ning and Perez, 1996) claimed that the rate of model convergence is seriously affected by the initial weight parameter. Kuo (Kuo, 2001) proposed an interesting approach which involved using ANNs with initial weights generated and optimised by using a genetic algorithm based method. Prior to the studies in this thesis a range of initial weight values were tested on the three datasets described above. These weights were initialised to a random value, within a range whose minimum and maximum values were between 0 and 0.001, incremented upwards to between 0 and 10. It was found that the mean squared error (MSE) and classification performance of the models on the validation data sets were not significantly different between each initial weight value (Figure 9), therefore an arbitrary value, initialised to a uniformly distributed random value between 0 and 1 is generally used.

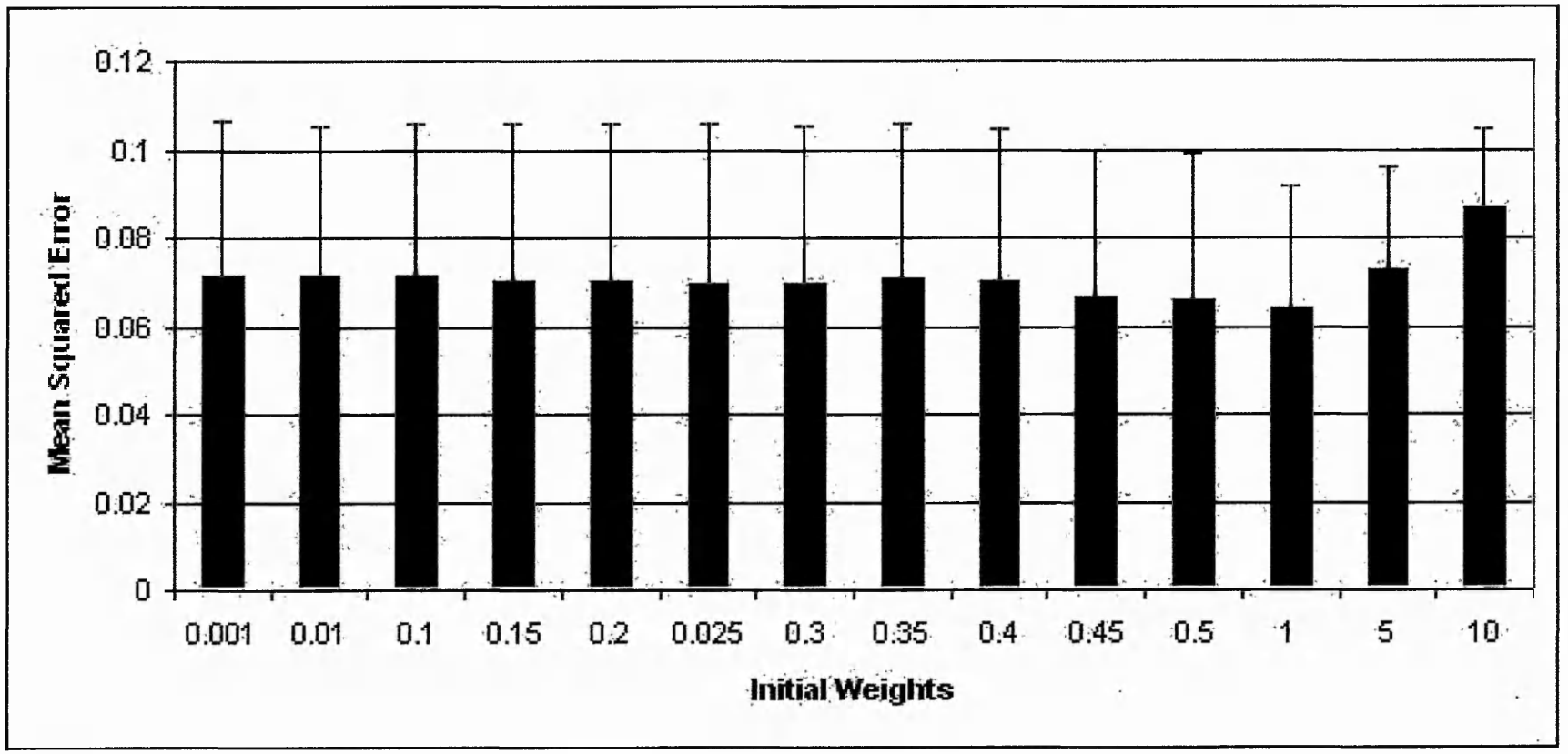
2.1.3. Learning rate

As previously stated (see introduction) the weight change of a neuron is proportional to the influence an input had on the error during training and the learning rate is a constant which controls the size of these weight changes (Dayhoff, 1990). The amount of weight modification is the learning rate times the error. For example, if the learning rate is 0.5, the weight change is one half the error value. So, it is clear that the careful selection of this value is important in order for training to reach the most optimum solution to the problem being modelled. Maier and Dandy (Maier and Dandy, 1998) found that when a lower learning rate was used (0.005), the error decreased slowly until a local minimum in the error was reached, and here it remained. When a learning rate of 0.1 was used, better predictions were found with faster training, as the network is more likely to escape from areas of local minima, and find a more global solution to the problem. These findings are comparable to those found in these studies, where learning rates between 0.1 and 1.0 were trialled, with 0.1 producing the lowest predictive error. This is because the learning rate values above 0.1 were causing the model to take steps that were too large, thus skipping and oscillating over the global minima so that convergence could not occur.

2.1.4. Momentum

As mentioned, large learning rates may often lead to oscillation of weight changes which results in either the learning process never completing, or the model converging towards a non-optimal solution. Adding a momentum factor to the BP learning algorithm is one technique that may be applied for a faster convergence towards minimum error, whilst also smoothing out the oscillations that may occur with a high learning rate. This momentum factor speeds up the training process by adding a proportion of the previous weight changes to the current weight changes. A high momentum will reduce the risk of the network being stuck in local minima, but risks overshooting the solution, whilst a small momentum value leads to slower training (Basheer and Hajmeer, 2000). Generally a trial and error procedure is used when seeking out the best learning rate and momentum combinations for any particular data.

Figure 9. Testing of initial weights prior to training



Attoh-Okine (Attoh-Okine, 1999) found that a learning rate in the region of 0.2 to 0.5 with a momentum factor between 0.4 and 0.5 appeared to be the appropriate combination, whilst Maier and Dandy (Maier and Dandy, 1998) came to the conclusion that although altering the momentum factor had no significant affect on the error, the learning count was greatly reduced with a higher momentum of 0.9. Raimundo and Narayanaswamy (Raimundo jr and Narayanaswamy, 2003) found several combinations of learning rate and momentum to be efficient during their study. Varying the values of learning rates between 0.05 and 0.4 and combining with a momentum value of either 0.3 or 0.5 produced the most effective models. In two separate studies, Mittal and Zhang (Mittal and Zhang, 2000; Mittal and Zhang, 2000) tested various combinations of learning rates and momentum factors and found that in one study a learning rate of 0.7 and a momentum of 0.5 achieved the best prediction results, whilst in another, setting both the learning rate and momentum factor to 0.7 was more effective. This showed that different values are appropriate for different data sets. In this study, in combination with varying learning rate values, momentum factors between 0.1 and 0.9 were trained and tested on the three datasets described previously. The results from these models can be seen in Figure 10 and it is evident that for the type of complex datasets used throughout this thesis, a learning rate of 0.1 combined with a momentum factor between 0.5 and 0.9 resulted in the network converging to the most optimal solution.

2.1.5. Training to convergence

Training convergence can be determined by a number of factors, depending on whether the training or the test set is to be used as an indicator of when training should be stopped. Examples of different criteria (computed at the end of each epoch) which may be used are; (i) average error below a pre-defined level, (ii) epochs since the minimum average error exceeding a specified number, (iii) largest error below a predefined level and, (iv) learning epochs exceeding a predefined number. The error on training data generally decreases with increasing number of epochs, with an initial large drop in error which slows down as the network begins to learn the patterns representing the data set. However, if training is allowed to continue beyond the point at which the error reaches the global minima, overfitting (or overtraining) may arise, where memorisation of the training data occurs. Because of this overfitting, if a network performance is monitored by training data alone, the network will perform with little error on the training data but will not be able to generalise well for new data. For this reason, during this research model convergence was

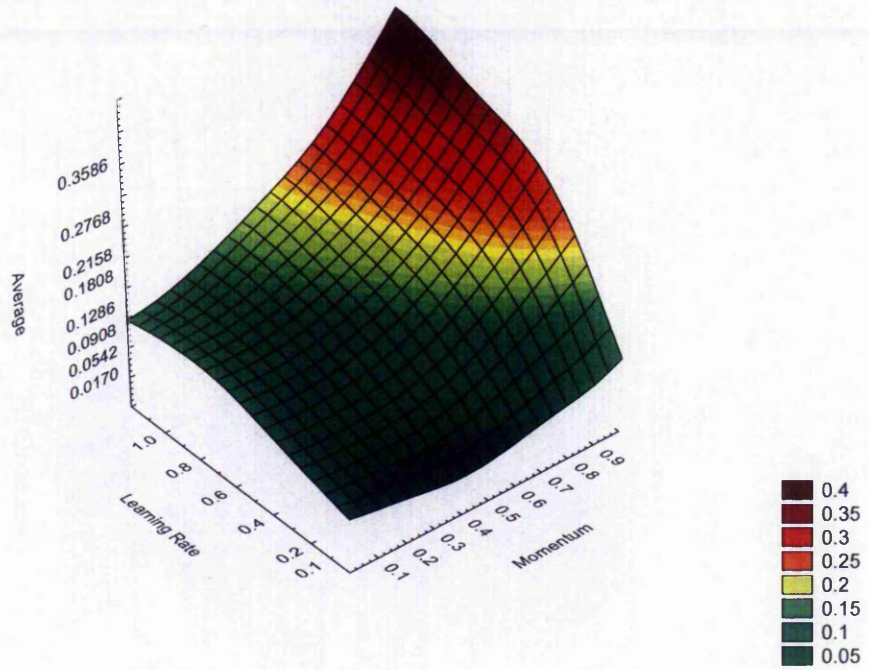
determined by a failure of the model to improve on the minimum MSE on the test data for a window of 20,000 epochs. This was enhanced further by using multiple resampling with RSCV to a large number of different training/test/validation data splits, thus creating several sub-models so that all data points are treated as unseen data a number of times, and the global data set is presented to the model in a number of different forms. This repetition allows confidence intervals to be determined for any given model, and results in robust generalised models.

2.1.6. Hidden layer size

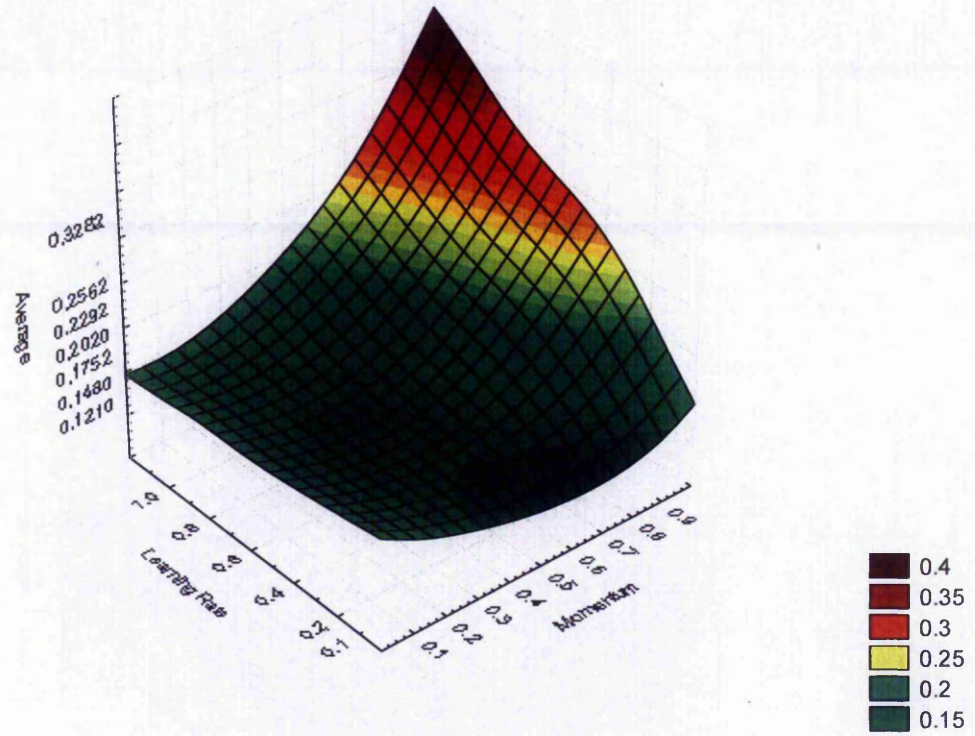
The purpose of the hidden layer is to enable the ANN to classify input data with nonlinear characteristics. A network with too few hidden nodes in this layer will result in only a linear estimate of the solution to the problem due to fewer nonlinear components being present. Conversely, too many hidden nodes will result in over-training where the algorithm will model the noise in the data, resulting in poor generalisation (Basheer and Hajmeer, 2000) and increased training times. Generally a trial and error approach is used in order to find the optimum number of hidden nodes, where a low number of hidden nodes are used to begin with and this number is gradually increased until a minimum error is reached, as shown by Srecnik et al. (Srecnik, et al., 2002). Other heuristics have also been proposed for determining the number of hidden nodes, such as using the $2n+1$ rule where n is the number of nodes in the input layer (Fletcher and Goss, 1993). One technique which may be used in order to optimise the number of hidden nodes by using a more intelligent means is correlated activity pruning (CAPing) (Roadknight, et al., 2001). This approach begins with a large number of hidden nodes, and removes units with constant outputs over all of the training patterns (as these are not contributing in the solution). Hidden nodes with identical or opposite (correlated) activation energies for all patterns can be combined, thus reducing the number of units until the optimum number is found. The process involves monitoring activation strengths at each hidden node and calculating the correlation coefficient for each pair of hidden nodes. The pair of nodes with the correlation coefficient nearest 1 (or -1) are replaced by one node and the weights are consequently changed. This process is then repeated, until generalisation is lost and therefore the minimum number of hidden nodes can be found whilst maintaining good generalisation.

Figure 10. Optimisation of learning rate and momentum factor values. Architectures with various momentum factor values and learning rates were trained in order to deduce the optimal ANN model over three different datasets for (a) SELDI-TOF MS, (b) MALDI-TOF MS, and (c) Gene expression data.

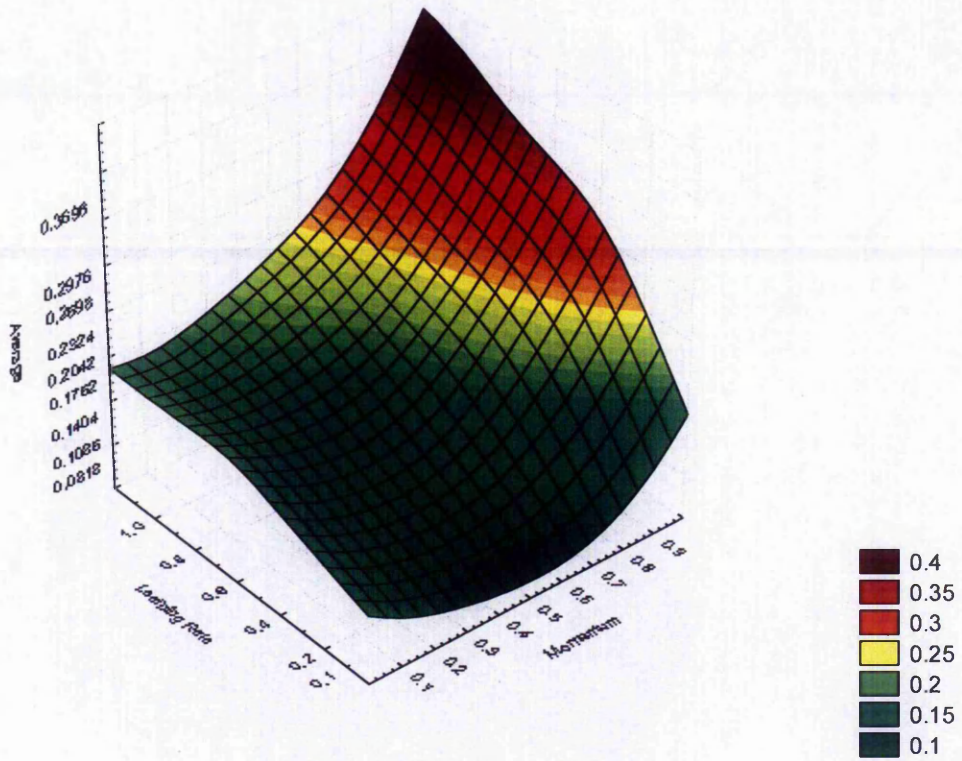
(a)



(b)


Average
0.3282
0.2562
0.2292
0.2020
0.1752
0.1480
0.1210
Learning Rate
0.1
0.2
0.3
0.4
Momentum
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.4
0.35
0.3
0.25
0.2
0.15

(c)


0.3696
0.2976
0.2658
0.2324
0.2042
0.1762
0.1404
0.1086
0.0818
Learning Rate
Momentum
0.4
0.35
0.3
0.25
0.2
0.15
0.1

Using the three datasets used in this section, different numbers of nodes in the hidden layer were tested on each dataset, firstly using the CAPing approach (Figure 11), and secondly by testing between 1 and 50 hidden nodes, as can be seen in Figure 12. Both approaches showed that whilst 1 hidden node performed poorly for unseen data (due to it essentially providing a linear estimation of the problem), no significant increase in performance was gained by increasing this number beyond between 2 and 5 hidden nodes. Furthermore, as the number of nodes is increased, training times increase almost exponentially, adding support to the suggestion that the number of hidden nodes should be kept as small as possible.

2.1.7. Summary

Throughout this thesis for all data analyses conducted, a three-layer MLP ANN with a feed forward BP algorithm with a sigmoidal transfer function has been used. Prior to training, data were scaled linearly between 0 and 1 using minimums and maximums. Here, the raw values were scaled linearly, so that the smallest value for each variable in the dataset was scaled to the minimum value, and the largest value in the dataset was set to the maximum value. This scaling method ensures that all relationships amongst the data values are kept identical, therefore not introducing any potential bias into the data.

Based upon the findings in this section, the network parameters used for all analyses (unless otherwise stated) were as follows:
Learning rate of 0.1.
Momentum value of 0.5.
Hidden layer size of between 2 and 5 nodes.
Data randomly split into subsets of 60 % training, 20 % test, and 20 % validation.
Initial weights of the network randomised between 0 and 1.
Training terminated once the network fails to improve on the test error for 20,000 epochs.

Figure 11. CAPing of hidden nodes. A CAPing approach was used to determine the minimum number of hidden nodes that could be used whilst maintaining good generalisation. The number of hidden nodes are presented on a logarithmic scale, with points representing 1000, 16, 8, 2 and 1hidden node(s).

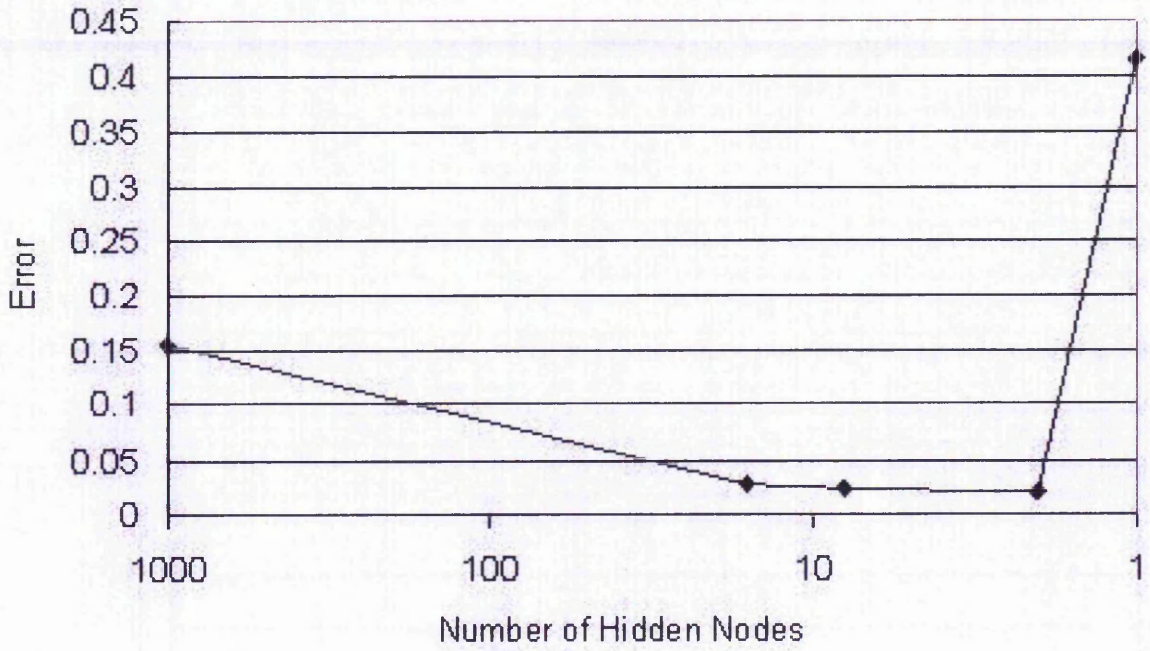
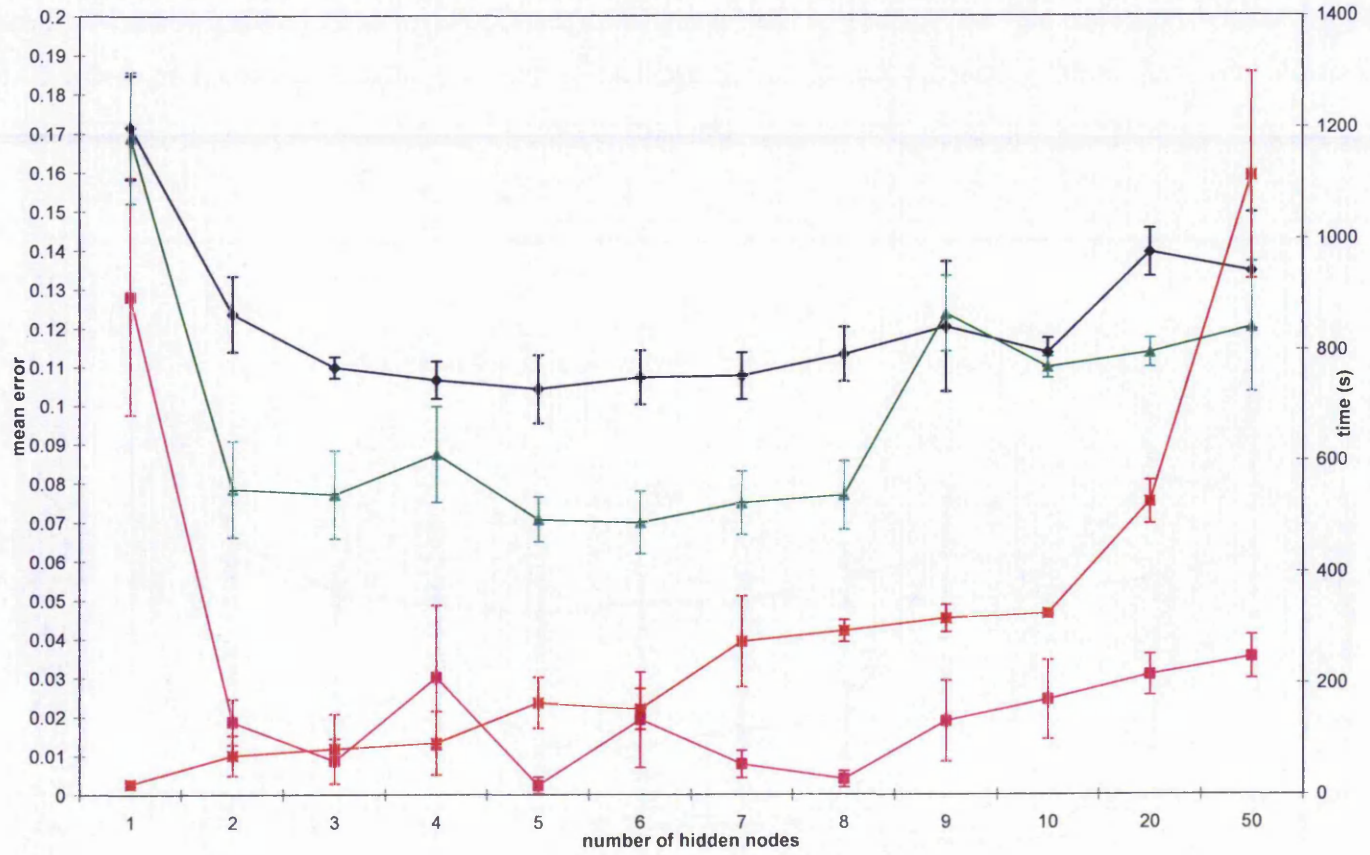


Figure 12. Mean error and mean training times with differing numbers of hidden nodes. Pink line indicates the SELDI-TOF MS dataset, green line the MALDI-TOF MS dataset, and blue line the gene expression dataset. Red line shows training times. Error bars indicate 95 % confidence intervals.



2.2. Basic parameterisation: Parameter reduction in microbial diagnostic models

2.2.1. Introduction

As mentioned previously data from proteomic and genomic sources is extremely complex. Therefore in studies aimed at analysing datasets containing large sample sizes and high dimensionality, advanced computer algorithms must be utilised to identify biomarkers capable of high classification accuracies by explaining the variation within the dataset, and determining any population characteristics. Here, a parameterisation approach was used based on the analysis of the weights of the trained ANN models in order to reduce dimensionality. The principle here is that the greater the absolute value of an inputs weighting to the hidden and output nodes, the more important this input was in the development of the model and sample classification (Garson, 1991). This approach was applied to a proteomic (SELDI-TOF MS) analysis of a host of bacterial pathogens, provided by the Health Protection Agency with the aim of generating predictive models in order to rapidly classify and identify samples based upon their proteomic profiles.

The current problem in the diagnosis of infectious diseases is that in some cases traditional microbiological tests (such as those outlined by Reddick (Reddick, 1975), D'Amato et al. (D'Amato, et al., 1978), Craven et al. (Craven, et al., 1978), Robinson and Oberhofer (Robinson and Oberhofer, 1983), and Janda et al. (Janda, et al., 1984) have performed poorly in the classification of bacterial species and strains. For example it is often difficult to determine a correct classification of a field isolate based on tests created for reference strains (type strains), because in many instances the field isolates having counterpart reference strains have undergone evolutionary changes resulting in intermediaries within the population. This variation makes clinical diagnosis difficult and pathogens may be misidentified when diagnosis is based purely on traditional microbiological tests. If computer algorithms are capable of identifying phenotypes through expression profile analysis, and ultimately identifying candidate biomarkers which correlate strongly to a pre-determined observation, then this would provide the potential for the development of decision support tools which may be utilised to supplement human judgment and potentially reduce the turn-around time of identification or diagnosis compared with existing clinical and microbiological methods.

Therefore because of the current limitations in strain identification the aim here was to determine if these parameterisation approaches using the weights of trained ANNs could be used to identify biomarkers which are capable of predicting class from such complex biological datasets. Members of the genus *Neisseria* and other closely related species were utilised with the ultimate aim being the reduction in complexity of the system in order to accurately discriminate the pathogen *Neisseria meningitidis* from other closely related species. This could potentially facilitate the development of novel rapid identification systems. The genus *Neisseria* contains a number of species and these may be both normal flora, and pathogens of humans and animals. Data used were generated from standard NCTC (National Collection of Type Culture) strains. Of these species, *N. meningitidis* and *N. gonorrhoeae* have been studied widely because of the severity of the infections they cause. *N. meningitidis* and *N. gonorrhoeae* exhibit over 90% homology between their genomes, however their respective site of infection, disease picture and antibiotic therapy vary markedly, highlighting the need for new, more accurate, rapid diagnostic tests to aid in the identification of these pathogens.

In molecular microbiology, hierarchical clustering is a method which is commonly used in order to study and determine the relationships between different bacterial strains, and is often used to place new strains into a particular taxon (Andrighetto, et al., 1998; Rebuffo, et al., 2006). However this approach is not always satisfactory when analysing highly dimensional datasets such as that seen in MS (Goodacre, et al., 1998). Bearing this in mind it still seemed appropriate that this study would incorporate hierarchical clustering into the analysis in order to study whether any possible outliers misclassified by the ANN within the two populations were more related to their original identification by traditional methods, or by their actual ANN classification. PCA was also used in a similar manner to provide a visual representation of how the samples related to one another.

2.2.2. Methods

For methods concerning bacterial culture preparation, and SELDI-TOF MS analysis, please refer to Lancashire et al. (Lancashire, et al., 2005).

Network parameters used were as detailed in section 2.1. The raw data obtained from the SELDI-TOF MS instrument consisted of individual *m/z* values with their corresponding relative abundance values. It is these relative abundance values for each *m/z* value that

were used as inputs in the input layer. The network utilised a constrained approach to maximise the efficiency of the analysis whereby two hidden nodes were used in the hidden layer. Two hidden nodes were used in order to amplify the importance of key ions within the mass spectrometry data, while producing accurate predictions and maintaining model generalisation. This approach was adopted with success on earlier MS data (Ball, et al., 2002; Mian, et al., 2003). Increasing the number of nodes in the hidden layer did not result in an increase in the ability of the model to predict strain (data not presented). The output layer consisted of a single node where strains known to be *N. meningitidis* were represented by 1 and other strains represented by 2.

Subsequent to proteomic analysis, the raw data from the SELDI instrument details the intensity of ions in the range of 0 to 98 kDa., giving a total of 33886 individual data points. Prior to ANNs model development and analysis, data points with *m/z* values below 3kDa. were removed, as anything below this was deemed to be noisy and unimportant due to the presence of matrix signatures. Due to the limitations in accurate mass resolution beyond 30 kDa., everything above this mass value was also removed (as shown in the paper by Ball and colleagues (Ball, et al., 2002)), leaving a total of 12822 ions to be used as inputs.

206 samples were used to train, test and validate the model. The data utilised was evenly split with 103 as *N. meningitidis* and 103 as "other" strains to prevent the predictive performance being in favour of one output class. Of the "other" strains group, the majority belonged to the genus *Neisseria*, with the addition of a few other closely related taxa such as *Kingella* and *Moraxella*. An additional 188 samples (60 of which were *N. meningitidis* and 128 were "other" species) were kept completely separate and brought in at a later date in order to provide a second order of validation utilising blind data for the final optimal model.

An initial model was tested using all 12822 ions as inputs resulting in a classification accuracy of 76 % (results not shown). Whilst this is relatively high, this model is of very high dimensionality, resulting in long training times due to its high complexity. Therefore parameterisation steps ensued with the aim of identifying which ions were the most important within the data set, allowing the model complexity to be reduced and the predictive capabilities to be markedly increased.

This was achieved by a "rolling input subset" approach which involved training the data containing inputs across a 3 kDa. mass range, and then shifting this along 1 kDa. at a time, in order to create a new data block. For example, the first block contained data within the 3-6 kDa. mass range, the next block ranged from 4-7 kDa., the next from 5-8 kDa. and so on up to 30 kDa. Each model was then trained over 50 random training/test/validation sub-models and relative importance values for each individual input were recorded so that they could be ranked according to their influence upon correct sample assignment. Relative importance values were calculated for each input by:

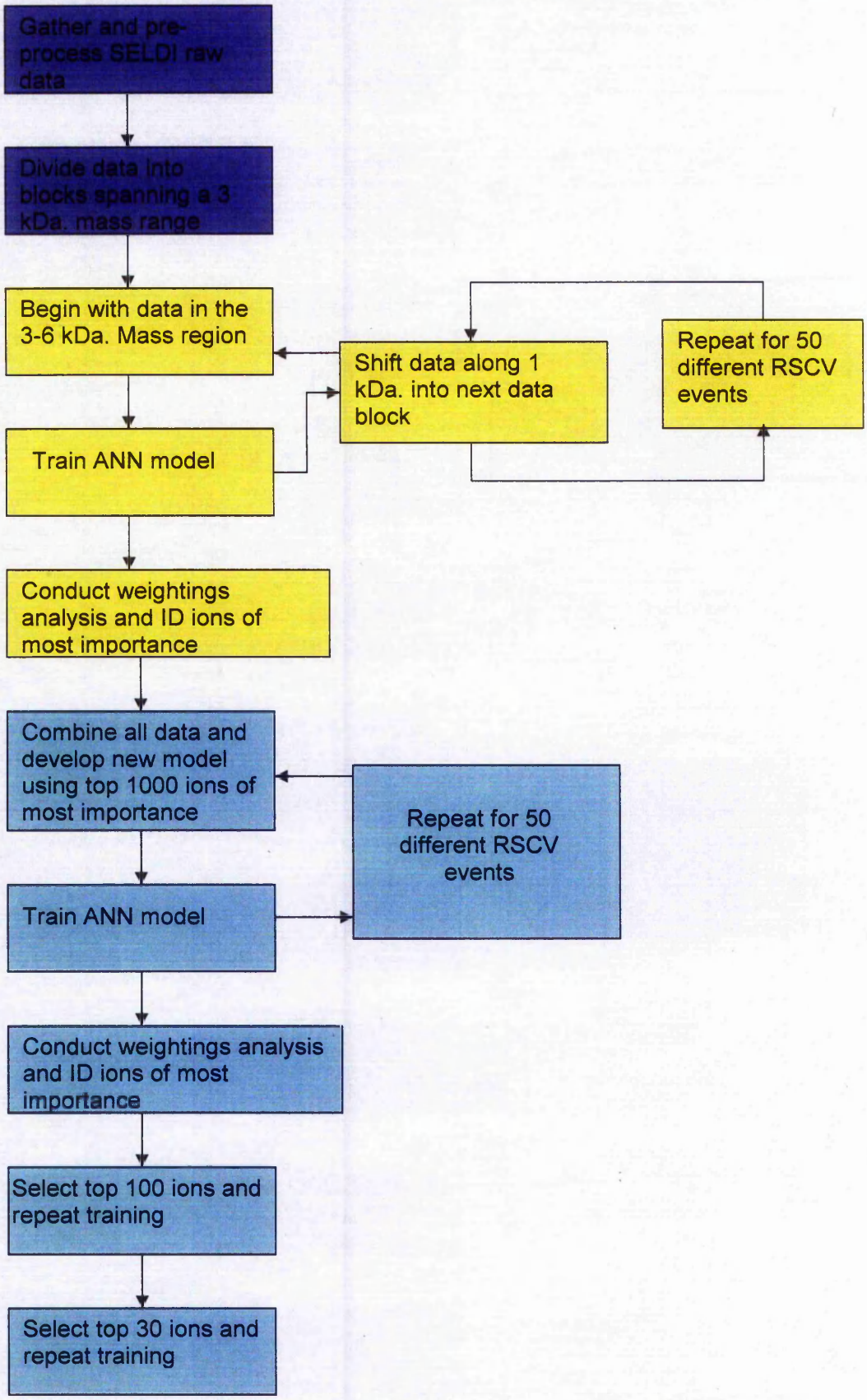
(i) Multiplying the absolute weight values of the network from the input node to the first hidden node with the absolute weight values of the network between the first hidden layer node to the output node.
(ii) Calculating this same value using the weightings to the second hidden node.
(iii) These two values were then summed to give a relative importance value for the input.
(iv) This value was then scaled relative to the total number of inputs.

The process was repeated for all of the remaining inputs to give the relative importance of each input with respect to all of the other inputs over a number of sub-models.

Once this initial analysis was completed, ions with the greatest importance were selected from the data in order to reduce model complexity. This was accomplished by selecting the top 1,000 inputs with the greatest relative importance values and repeating the training process. Relative importance analysis was again used to determine the top 100 inputs from these 1,000. This was repeated again to deduce the top 30 inputs, at which point the process was stopped due to no further significant improvement in the model.

In addition to ANN analysis at each parameterisation step, a cluster analysis and PCA was performed. The purpose here was to firstly look at the relationships between the samples within the population and secondly, to provide additional information regarding samples which were misclassified by the ANN as a means of providing a possible explanation for these misclassifications. It is important to note that PCA and cluster analysis were not used as predictive tools, but purely as a means to understand and visualise the data structures being analysed. Figure 13 shows a summary of the methodology steps taken in the analysis.

Figure 13. Flow diagram representing methodology overview of ANN analysis resulting in the identification of potential biomarkers. Parameterisation methods were conducted in order to deduce the top 30 molecular ions of greatest importance.



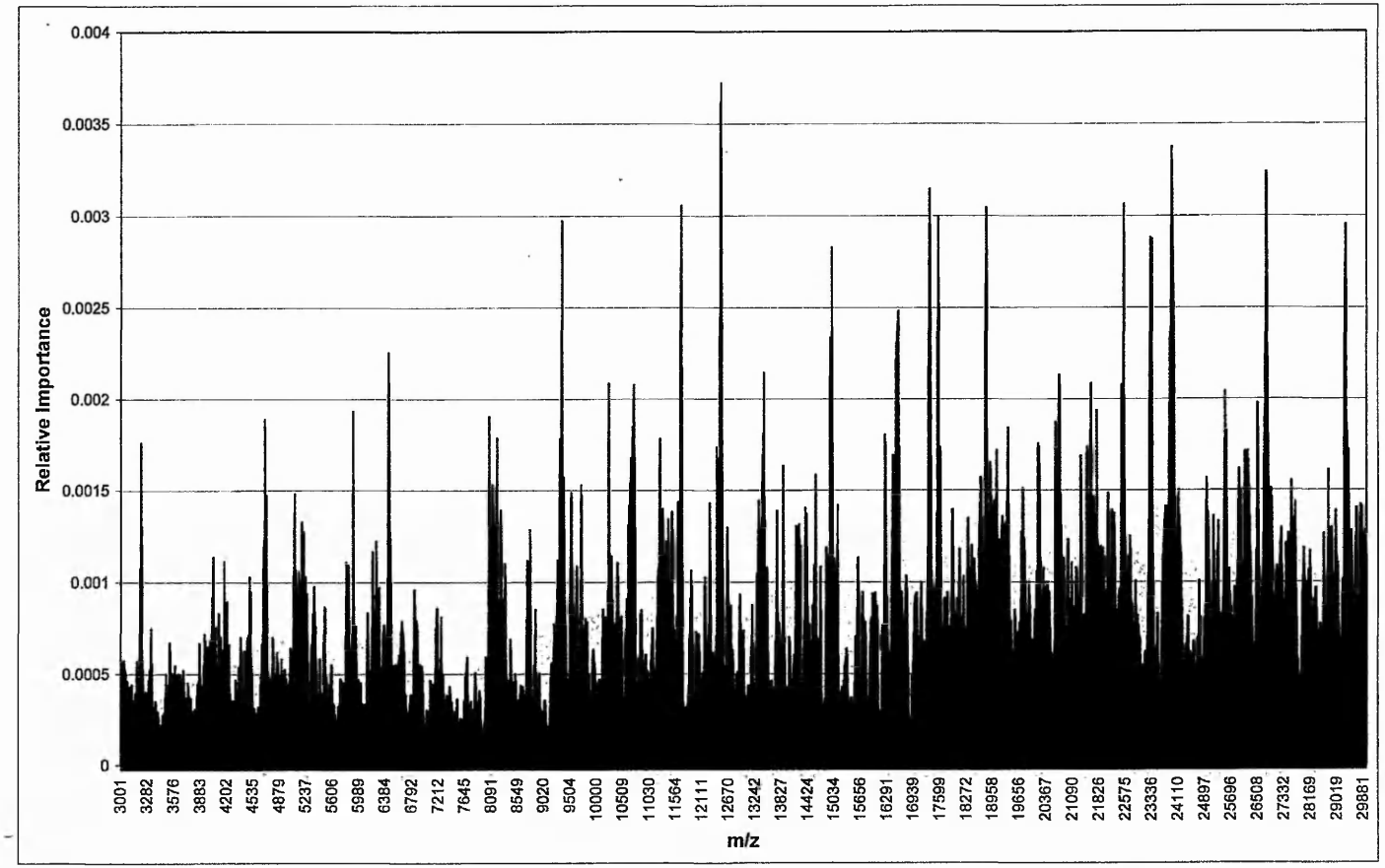
2.2.3. Results

2.2.3.1. ANN analysis

Ion mass intensity profiles generated from SELDI-TOF MS analysis were analysed in 3 kDa. blocks in order to determine their relative importance in classifying the samples into their respective groups. This enabled a proteomic profile to be produced for the whole mass range that was being analysed. Figure 14 shows the mean relative importance profile generated from the 50 sub models trained from each 3 kDa. block over the whole mass range of the SELDI data available. In order to reduce the complexity of the model and determine which of these ions were the most significant in strain prediction, all ions were ranked in descending order of relative importance and an arbitrary value of the top 1,000 ions of most importance were selected for further analysis. This reduction was necessary in order to determine any potential biomarkers whose intensities correspond with strain identification and which are capable of accurate discrimination between the two groups. Different randomly extracted training, test and validation sets were utilised to develop each sub-model. This repeated RSCV enables one to calculate the probability, or percentage chance, that a sample belongs to one group or another. The results from this 1,000 input model showed that for validation data, 203 out of the 206 samples were correctly classified, with 100 % sensitivity (percentage of *N. meningitidis* samples correctly classified) and 97 % specificity. It should be noted that an error threshold value of 0.5 was used to designate sample destination, *i.e.,* the actual output for a *N. meningitidis* was 1, and an output of 2 would represent "other" species. So if a sample was predicted at anything between 1 and 1.5, then it was classed as *N. meningitidis*, and if it was predicted at between 1.5 and 2, then it was grouped into the "other" strain category. The magnitude of this error may be used to identify strains of a similar nature.

Although these results were based on data from the whole profile and showed high prediction rates, if the system is ultimately to be used as a rapid diagnostic aid then it would need to be much more parsimonious to facilitate ease of data gathering and

Figure 14. Relative importance values for ion masses between 3,000 and 29,999 Daltons. These values represent a value obtained from multiple sub-models in which random initial weightings were applied to each.



minimise potential sources of error. Therefore, the relative importance values of these 1,000 inputs were again ranked and the top 100 were selected for further training. Training was again repeated as with the 1,000 input model and results from this model showed that 204 out of 206 samples were correctly classified, with 100 % sensitivity and 98 % specificity.

To deduce whether the number of inputs could be reduced further still, the top 30 inputs were chosen to be taken from these 100 and training was repeated. Predictive performance showed a further increase, correctly classifying 205 out of 206 samples, with 100 % sensitivity and 99 % specificity. Therefore it was evident that the parameterisation method applied here led to models which were increasingly more parsimonious whilst maintaining the high predictive accuracy. Figure 15 shows how the model performance changes with increasing dimensionality reduction. Further input number reductions beyond 30 inputs resulted in a drop in predictive performance, so these 30 inputs were deemed as the optimal set and the most important for predictive performance. This model had an area under the curve (AUC) value of 0.9994 when analysed using a ROC curve (Goodenough, et al., 1974; Lusted, 1971; Metz, 1978). A ROC curve determines the number of true positives, true negatives, false positives and false negatives and produces a summary statistic for performance. It achieves this by plotting the true positive rate against the false positive rate at different possible cutpoints (in this case, prediction error thresholds). The AUC value measures discrimination, that is, the ability of the model to correctly classify the true positives and true negatives. A perfect ROC curve (and therefore a perfect test) would have an AUC value of 1, so the closer the curve follows the left hand border and then the top border of the ROC space, the more accurate the test. The errors associated with the individual sample predictions are shown in Figure 16. Additionally, the 30 ions identified, with their respective relative importance values are tabulated in Table 1 and it is interesting to note that the majority of these inputs appear to arise in clusters around specific *m/z* values.

Once these key ions were identified, they were then applied to an additional blind data set of 188 samples. Using this sample set the model correctly identified 184 out of 188 (97.9 %) of the samples correctly, with a sensitivity of 100 % and specificity of 96.9 %. The ROC curve for this analysis is shown in Figure 17, achieving an AUC value of 0.986.

Figure 15. Changes in model performance with increasing dimensionality reduction. Error bars indicate 95 % confidence intervals.

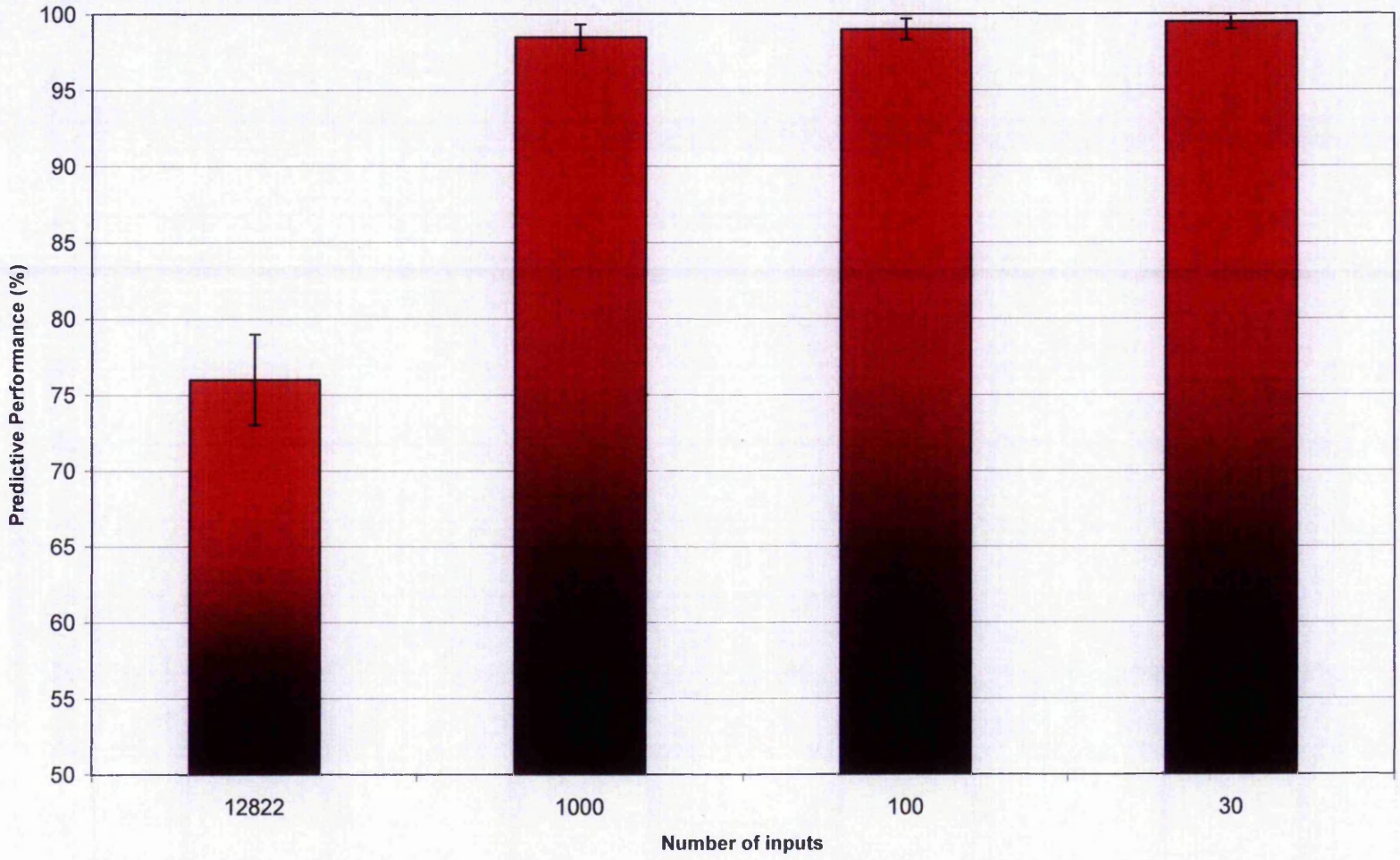


Figure 16. Error values demonstrating the ability of the top 30 ions to accurately predict bacterial strain. One sample was incorrectly classified, with an error value greater than 0.5. 95 % confidence intervals are represented by error bars.

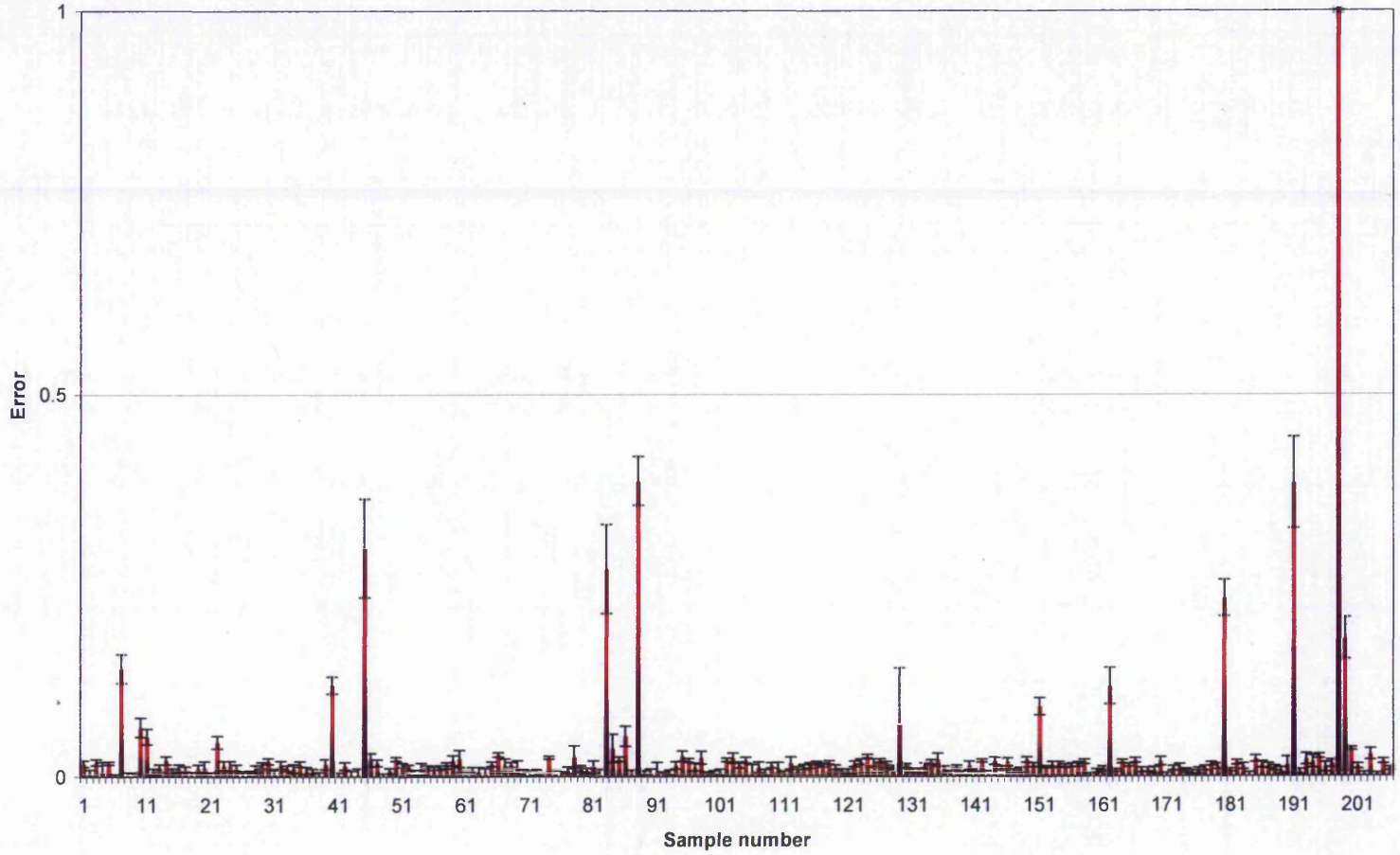


Table 1. The 30 ions with the highest relative importance with regards to strain prediction. These were ranked according to their molecular mass. Ions were grouped by colours if their masses clustered around specific mass ranges.

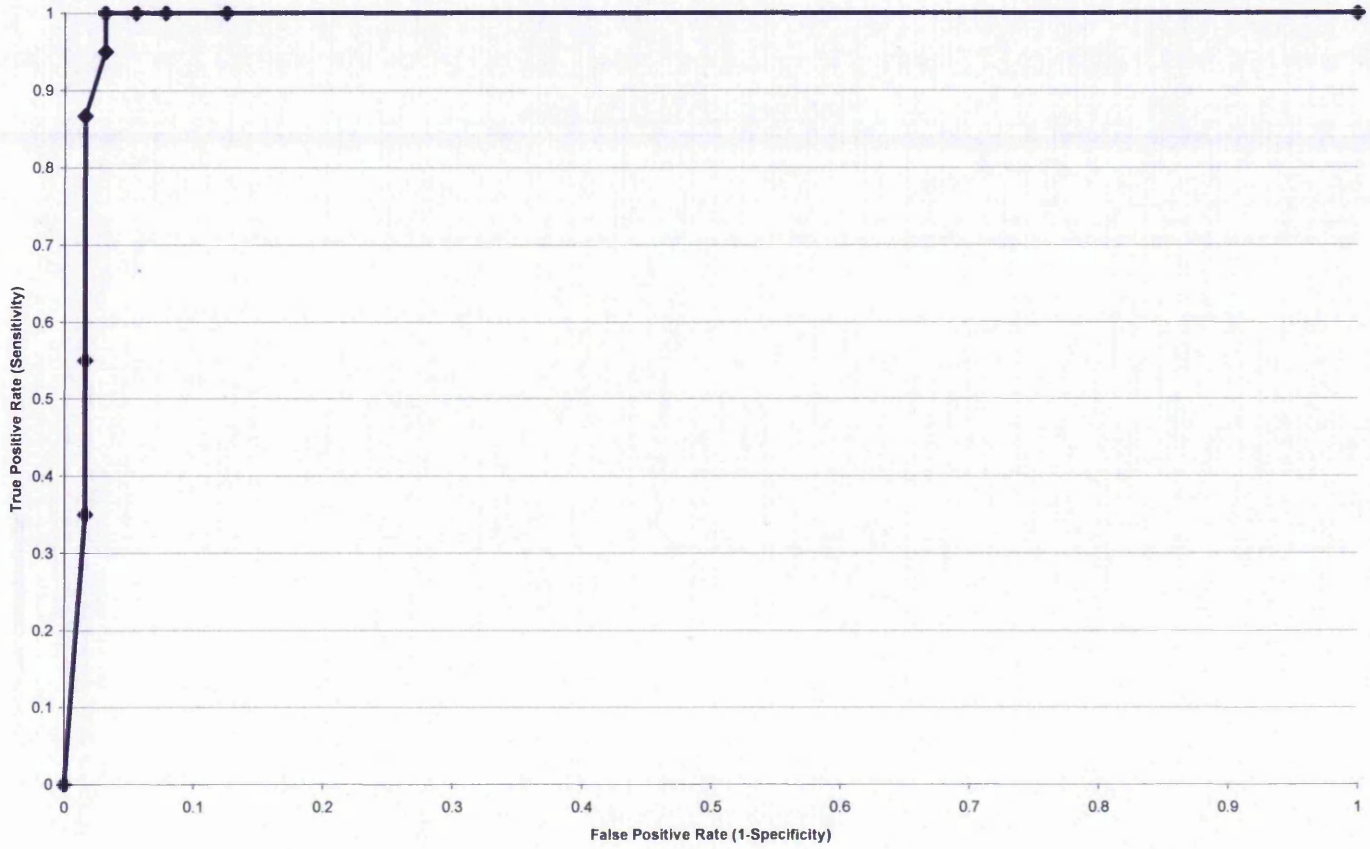
Input *m/z* value	Relative importance	Input *m/z* value	Relative importance
4709	0.2	19060	0.019
5085	0.074	19062	0.008
8181	0.008	19065	0.013
10663	0.2	19067	0.004
10665	0.09	19070	0.014
11672	0.003	19072	0.019
11674	0.015	23302	0.017
11676	0.029	23305	0.017
11678	0.022	23322	0.014
12515	0.008	23325	0.027
12517	0.007	23328	0.032
14609	0.042	23330	0.012
14611	0.014	23333	0.002
19055	0.035	23336	0.018
19057	0.021	29000	0.015

2.2.3.2. Cluster analysis, principal components analysis and similarity analysis

A clustering algorithm (complete linkage with distances measured by Euclidean distances) was applied to the data in parallel to the ANNs in order to measure consistencies between the various approaches and to visualise any possible outliers in the two populations. This may in turn lead to some explanations regarding the few samples which were misclassified by the ANN models. The results from this cluster analysis can be seen in Figure 18a-d.

It is evident from Figure 18a-d that this clustering approach, although not used to predict class, was capable of grouping samples according to their known identification reasonably well. Results here were fairly reproducible as the dimensionality of the data was reduced, although notably with some overlap between the populations. It is interesting to recognise however that of the three samples which the ANN incorrectly identified using the top 1000 ions (highlighted in Figure 18a by arrows), two of these are in clear clusters of the opposite group. For example, the sample on the extreme right of the incorrect classifications was identified using 16S rDNA methods as *K. denitrificans,* but clearly showed more similarities to the *N. meningitidis* samples than the "other" species. Furthermore, this trend continued, with the majority of the samples misclassified by the ANN being placed in clusters together with species more closely related to the ANN classification,

Figure 17. Receiver Operating Characteristic curve for the 30 ion model. This was a result of the model applied to a second validation data set consisting of 188 samples.



and not the initial pre-determined group. This is shown even more clearly in Figure 18d, which represents the cluster analysis of the second validation set of samples. Of the four samples which the ANN misclassified as being *N. meningitidis* (which by 16S rDNA were identified as *N. gonorrhoeae*), the cluster analysis, like the ANN, placed these in the cluster containing the majority of the *N. meningitidis* indicating that according to the proteomic profiles, these isolates are more similar to *N. meningitidis* than *N. gonorrhoea.* Thus by using these clustering approaches in concert with ANNs one can visualise the data using cluster analysis, and predict class with the ANNs. Furthermore we can begin to understand the nature of samples which may be outliers, and therefore may show characteristics which are not archetypal of that particular species which it is initially believed to belong to, and begin to understand why the ANN models that we have developed, predict some species to be one class and not another. To provide additional support to this, PCA and similarity analysis was conducted on this validation data (Figures 19 and 20).

This PCA matrix shows that all of the *N. meningitidis* samples (represented in dark grey) fall along one clear vector, and are all clustered closely together. Whilst it is evident that the other species belonging to the second population exhibit much more variation and fall along a vector far less clustered, it is interesting to see that the four samples which the ANN identified incorrectly as *N. meningitidis* are clearly placed along the same vector as the *N. meningitidis* by the PCA, indicated by arrows, in black.

A similarity analysis was also performed to determine which samples were the most similar to the incorrectly classified ones with respect to their mass spectra. From the 25 most similar samples to these four misclassified ones, 19 were *N. meningitidis*, and just 6 were "other" closely related species. This again provides reasoning to why the ANN classified them as *N. meningitidis,* and that these samples appear to actually be more related to what the ANN predicted than to what they were initially identified as using 16S rDNA analysis.

Figure 18a-d. Cluster analysis. Complete linkage using Euclidean distance measures for (a) Top 1000 ion model, (b) Top 100 ion model, (c) Top 30 ion model applied to original data set and (d) Top 30 ion model applied to second validation data set. The black blocks indicate samples which were *N. meningitidis* whilst the light grey blocks represent "other" species. The dark grey blocks show incorrectly classified samples (also highlighted with arrows).

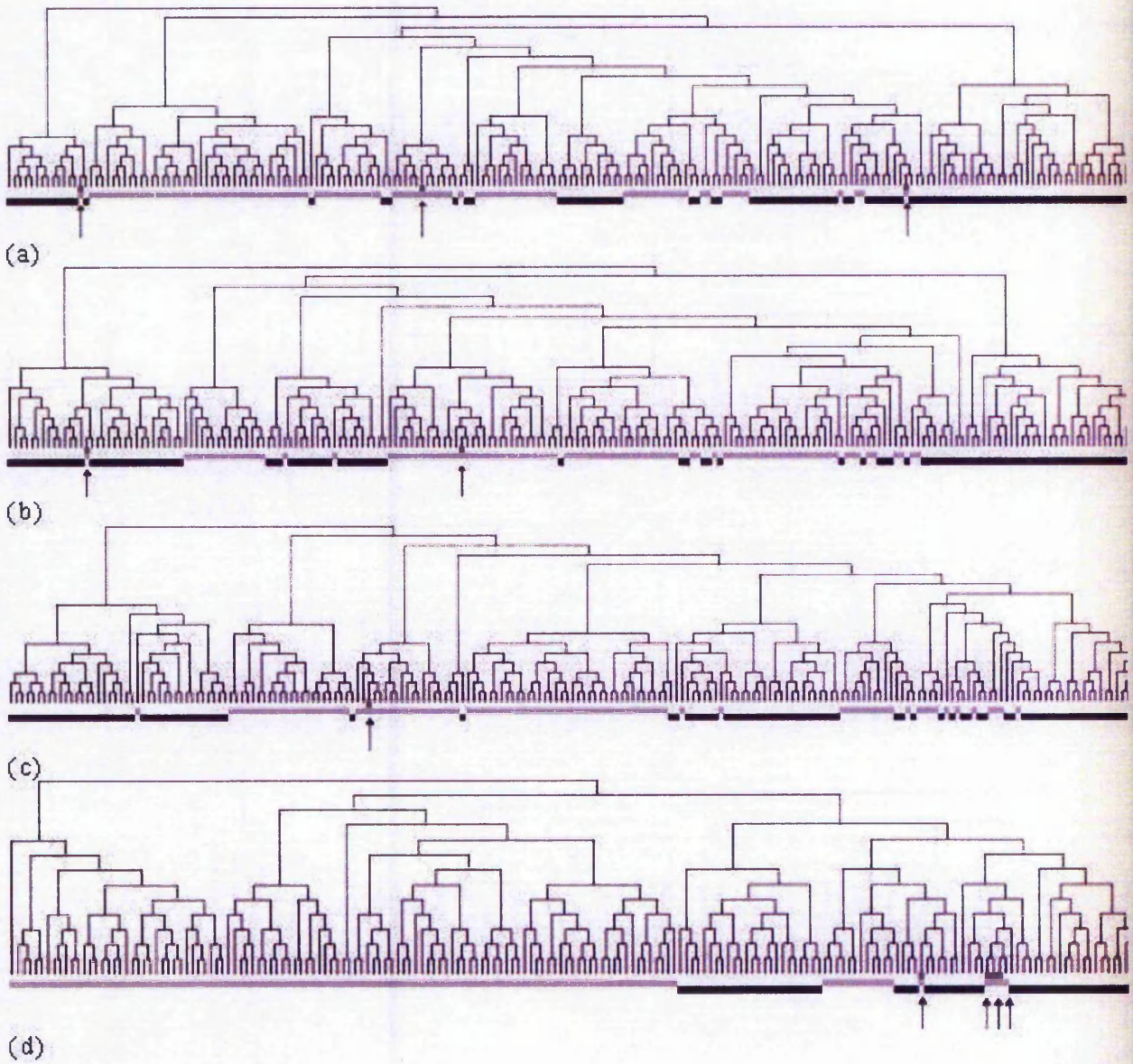


Figure 19. Principal Components Analysis. This was produced from the 30 ion model when applied to second validation data set. The dark grey blocks indicate *N. meningitidis* samples and the light grey blocks other samples. The black blocks and arrows indicate the samples that were misclassified.

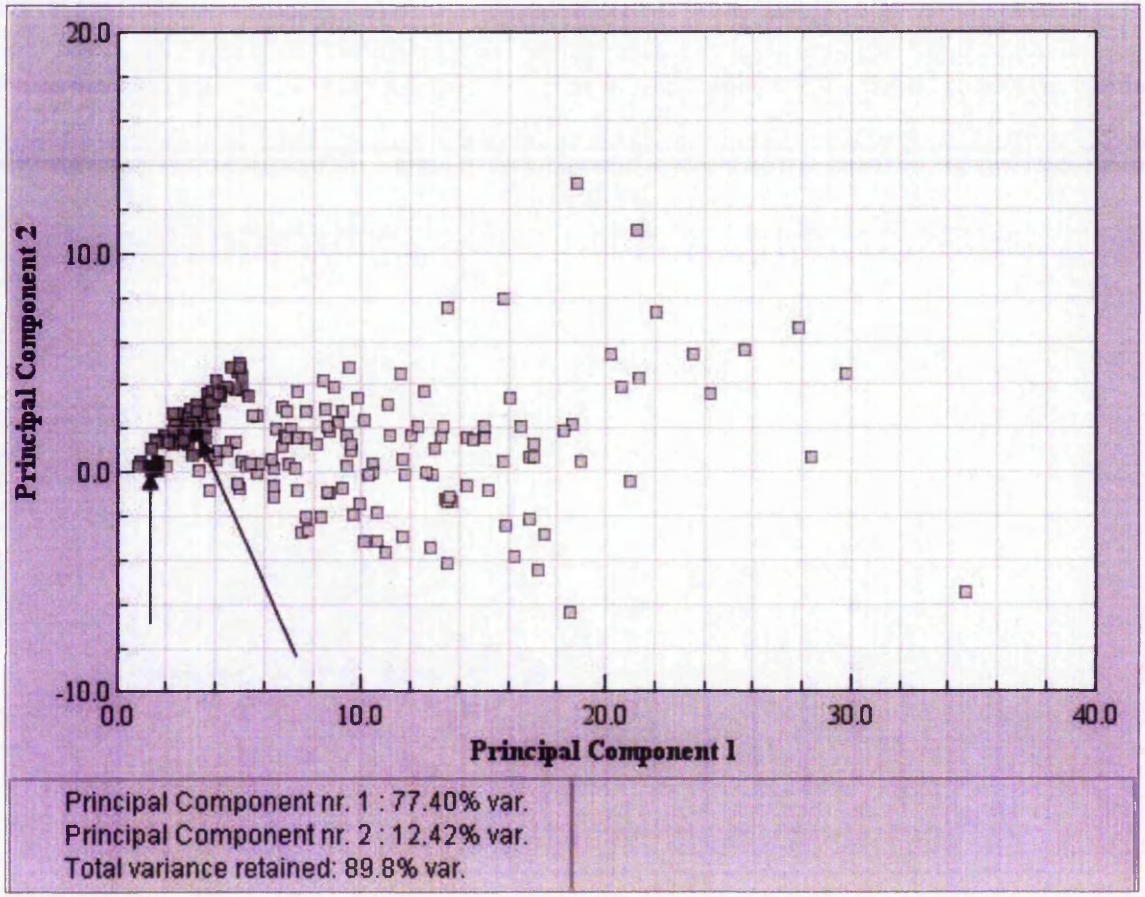
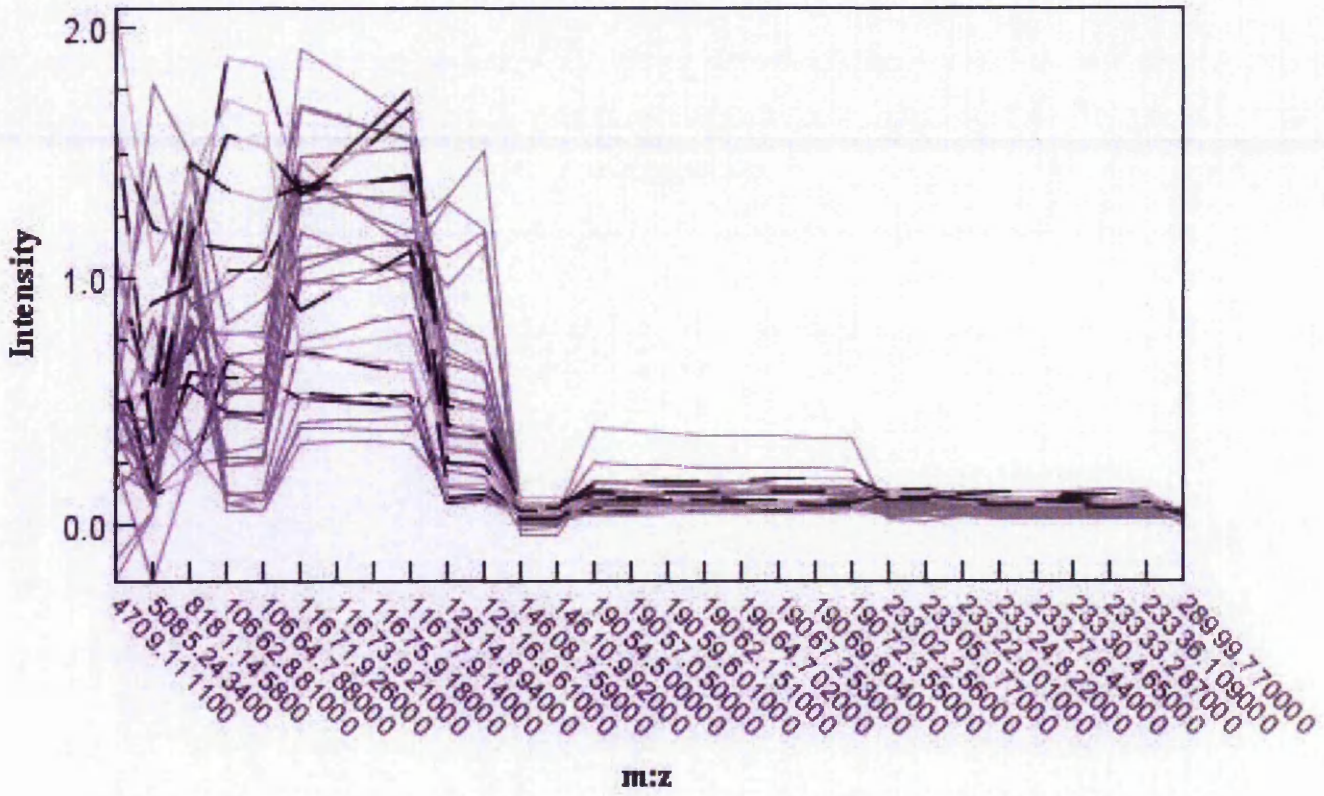


Figure 20. Similarity analysis. This was conducted to assess which spectra were most closely related to those samples incorrectly classified. Of the 25 most similar samples to the 4 which were misclassified, 19 were *N. meningitidis,* concurring with the ANN predictions rather than the 16S rDNA assignments.



2.3. Advanced parameterisation

2.3.1. Introduction

The previous section highlighted methods in which ANNs could be used in order to generate predictive models by using the weights of the trained ANN models to reduce the dimensionality of the data and identify potentially important markers. Here, this is taken a stage further by describing a method which generates parsimonious models, and also gives an indication as to the importance of each input that is used in the model.

To recap, computational data mining techniques are becoming increasingly important in the analysis of complex biological data. The analysis of biomedical data by proteomic means such as MS is extremely promising and provides an insight into the relative abundance of ions in a given sample. If methodologies are developed to analyse this data there is potential to identify proteomic patterns representative of a given state, such as healthy and diseased tissues. However, the problem arises within the data sets themselves. In a typical mass spectrum generated for a single sample, in excess of 30,000 individual data points are produced. Additionally, in order to interpret this data, as many samples as possible need to be analysed in order to provide an accurate representation of the possible sample to sample variation. Therefore with 30,000 data points per sample, the amount of data points requiring analysis can rapidly rise into the millions. In order to identify biomarkers representative of proteomic patterns which are capable of classifying samples into their assigned groups, advanced computational algorithms must be employed. These classifiers must be able to cope with nonlinear data which is of high dimensionality. They must be robust and create generalised solutions which not only model the data that is currently available to a high accuracy, but must also be capable of predicting the outcomes of future cases. One appropriate solution to this problem is to use an ANN modelling approach. However, perhaps the biggest limitation of ANNs is that they have been labelled as "black boxes". They are often looked upon in this way as even though their predictive capabilities may be good, they do not appear to give an explanation as to how they reach a solution. This is often undesirable if the inter-relationships between certain variables and their influence upon the outputs are being sought after.

Here we present a method which overcomes this problem by firstly simplifying the solution by reducing the complexity of the model. This is achieved by removing noisy

inputs which have little or even a negative influence upon the system (that is, inputs whose removal causes minimal changes to the network error function). This leads to a more parsimonious model with increased predictive performance which generalises better to new data that was not involved in the building of the model. The resulting model is less computer intensive and also aids in the identification of potential biomarkers which are indicative of the output in question. The method is based upon input pruning, and several common methods of pruning will be outlined and assessed followed by a novel meta-pruning approach which has been developed to increase the model performance further still. The methods described were tested on two sets of real data generated by SELDI-TOF MS and validated using RSCV. It is shown that by utilising these methods, the internal workings of ANNs can be elucidated, providing a full understanding of how they reach a particular solution.

In ANNs, the interchangeable weights between the different layers contain all of the information regarding the solution to the problem. The importance of each input depends upon the direction and strength of this weight value. Negative connection weights represent inhibitory effects on neurons, and decrease the value of the prediction, whilst positive connection weights increase the value of the prediction.

In order to determine which inputs are contributing the most to a given outcome, the influence and importance of each needs to be evaluated. The most commonly used methods to achieve this are (i) sensitivity analysis, (ii) analysis of connection weights and (iii) the analysis of absolute weight values, used in the previous section (e.g. Garson's algorithm (Garson, 1991)). These will now be covered in more detail, leading to the new proposed meta-pruning approach.

2.3.1.1. Sensitivity analysis

Sensitivity analysis is often used as a form of input pruning and if a number of models are studied, one can ascertain which variables are consistently of a high (and low) sensitivity. The basic principle behind sensitivity analysis is to run the network on a set of test samples, and determine the overall error for the predictions on these test samples. The network is then run again using the same test samples, but this time the observed value of a particular input, *i*, is replaced as if it were unavailable (Hunter, et al., 2000), and again the error is calculated. It is expected that if any important information is being removed from

the model there would be some deterioration in the error. The sensitivity ratio of input *i* is determined by the ratio of the error when the input was unavailable, to that of the original error. The importance of input *i* relative to all other inputs can then be calculated according to:

$$\mathbf{Ratio}_i = S_i / (1/n)$$

Where S is the sensitivity of input *i,* and *n* is the number of inputs.

Therefore, the more influential an input is on the model, the greater the ratio. If the ratio is one or lower, then eliminating that variable either has no effect on the performance of the network, or even improves it. So any input with a ratio of one or less, can essentially be removed without any detrimental effects to the predictive capabilities of the model.

2.3.1.2. Analysis of connection weights

This method examines the raw values of the weights leading from each input to the hidden node, and from each hidden node to the output. It takes the product of each input to hidden, and each hidden to output connection weights. In this study the method by which inputs were retained was to calculate the mean overall connection weight over all inputs and calculate thresholds based on standard deviations similar to that discussed by Narayanan *et al.* (2004). Inputs whose overall connection weights exceeded ± 1 standard deviation from the mean were retained, and inputs within this range were discarded.

2.3.1.3. Analysis of absolute weights

This analysis was proposed by Garson (Garson, 1991) and partitions the neural network connection weights in order to determine the relative importance of each input variable in the ANN model. This algorithm takes the product of the absolute values of the interconnecting weights to produce a weight value that is attributable to each input. The inputs can then be ranked by this weighting and from this it possible to calculate the relative importance ratio (as with sensitivity analysis). All inputs with a relative importance ratio greater than 1 can then be retained in the model. As this is the absolute value, this method does not provide information regarding the directional influence a particular input may have.

2.3.1.4. Meta-Pruning

Each of the different methods outlined above have their own advantages with regards to identifying the importance of input parameters in the ANN model. However, each approach consists of its own algorithm, and therefore has the potential to reach a different solution. This raises the question of which approach is identifying the most important input variables, and how can the user be confident that the inputs identified as the most important, actually are the most important?

Here we propose a novel "meta-pruning" approach. This combines these three input pruning methods in order to provide a consensus approach aimed at understanding the variable influences in ANNs, and identifying important inputs with a high degree of confidence.

With this approach, results from sensitivity analysis, analysis of connection weights and the analysis of absolute weights are obtained. All inputs with a ratio greater than one from the sensitivity analysis are retained. Likewise from the analysis of the absolute weights, all inputs with a ratio greater than one are kept. From the analysis of connection weights, inputs whose overall connection weights exceed ± 1 standard deviation from the mean are retained.

The resulting inputs can then be compared against one another and only those inputs which occur as highly important, *i.e.,* have been retained in all three approaches are subjected to further training. This method ensures that each input which remains in the model is consistently highly influential across various approaches, thus increasing the confidence in the model derived from these inputs, and further reducing the complexity of the solution.

2.3.1.5. Datasets used for analysis

Two separate datasets extremely different in their origins were utilised in this study as a basis for comparing the various methods. Both datasets were generated by SELDI-TOF-MS and as such raw data files were imported into the ANNs for analysis. The first data set consisted of 209 samples of the bacterium *Neisseria gonorrhoea,* the etiological agent of the sexually transmitted disease Gonorrhoea. This data consisted of two populations of *N. gonorrhoea*; the first population containing 81 samples was from one specific outbreak, and the second population consisted of 128 samples originating from another outbreak which occurred at a different location to the first. The aim here was to determine whether

ANNs could be combined with SELDI-TOF MS in order to potentially identify specific outbreak isolates, which would form the basis of using ANNs to pinpoint the origins of outbreaks of pathogenic organisms. This would also determine whether ANNs have the potential to be used in such a way that would aid in the understanding of the inter- and intra-population characteristics of organisms. This may in turn lead to an explanation of how pathogens change and adapt through time, by for example the development of antibiotic resistance in response to environmental stress.

The second data set contained 206 samples, 101 of which were diagnosed as Stage I melanoma, and the remaining were identified as Stage IV. The aim was to utilise ANNs to determine biomarker patterns which would have the potential to aid in the classification of tumour staging.

In order to compare the effectiveness of the different pruning methods, each method was tested in turn to compare the accuracy of the models developed. ANN architecture, parameters and training were as detailed in section 2.1. The inputs consisted of the raw data from the SELDI-TOF MS spanning the 3-30 kDa. mass range, therefore each input represented an *m/z* value along with its relative intensity.

Models were trained by randomly splitting the data up into separate training/test/validation data splits. This was repeated 50 times and models were trained for each data split ensuring that each sample is treated as validation (unseen) data a number of times

2.3.2. Results

The initial model for each dataset consisted of training with all inputs from the SELDI-TOF MS analysis within the 3-30 kDa. mass range. These models consisted of approximately 12,000 inputs per model, which is clearly a highly complex system. Performance results for the validation data using the melanoma data set were 51 %, and for the *N. gonorrhoea* data set was 77 %.

Input influences on the models were then calculated using the three methods outlined above. Sensitivity analysis identified 7093 inputs which were having a positive influence on the *N. gonorrhoea* model, and 7009 inputs in the melanoma data set. These were selected for comparison and the remaining inputs were discarded.

Weightings analysis using absolute weight values identified 5446 inputs of high importance within the *N. gonorrhoea* dataset and 5455 inputs within the melanoma data. Again, these were selected for comparison.

Finally, the analysis of the raw connection weights identified 4094 inputs which were having an important role in the correct classification of samples from the *N. gonorrhoea* data, and 4093 inputs of importance from the melanoma data sets. As before, these were selected for comparison.

The inputs identified as important using the various pruning methods were then compared to determine which of these were present across the three methods. These could then be used in a model to assess whether the proposed meta-pruning approach would result in a more parsimonious and powerful model, by only identifying inputs which were consistently of a high importance across a variety of methods. This would increase the confidence in inputs identified as potential biomarkers indicative of class, in this case, origin of bacterial isolate, or tumour stage. The method led to the reduction of the data further still, with 940 inputs remaining in the *N. gonorrhoea* dataset, and 964 inputs remaining in the melanoma dataset.

2.3.2.1. *Neisseria gonorrhoeae* data

Each model was trained as described earlier using RSCV over 50 random data splits. Using the inputs derived from sensitivity analysis, the model predicted 82 % of the validation set correctly. The inputs identified as important using weightings analysis based upon the absolute weight values predicted to an accuracy of 84 % and using the inputs from the overall connection weights method, 91 % of the samples were correctly identified. The model containing the inputs from the meta-pruning approach correctly classified 95 % of the samples correctly, a significant improvement over the other methods. Next, to determine if a smaller subset of inputs could be identified whilst retaining the model performance, this meta-pruning process was repeated using these 940 inputs. This led to the identification of 218 inputs which correctly classified 98 % of the samples correctly, improving further still on the previous model. Figure 21 shows these results from the validation data sets across the different methods, and Figure 22 details the performance results for the different methods for the training, test and validation data splits.

2.3.2.2. Melanoma data

As with the *N. gonorrhoea* data, each model was trained using RSCV. Inputs selected by sensitivity analysis resulted in a performance accuracy of 51 % for validation data. Weightings analysis by absolute values selected inputs which correctly identified 70 % of the samples, whilst using the actual connection weights identified inputs which predicted 77 % of the samples correctly. As with the bacterial data set, the meta-pruning approach significantly out-performed the conventional pruning methods, predicting 81 % of the samples into their respective groups. Once again, meta-pruning was repeated for a second step to determine if a smaller subset of inputs could be identified which would predict the samples into their respective groups even more accurately. This led to the identification of 237 inputs which now correctly classified 93 % of the sample correctly, showing a further improvement on the previous model. Figure 23 illustrates the prediction results for validation data, whilst Figure 24 shows the performance of the different pruning methods for training, test and validation data splits.

Figure 21. Overall median accuracy of the different pruning methods. Results are shown for validation data across 50 random sample cross validation data splits using the *N. gonorrhoeae* data.

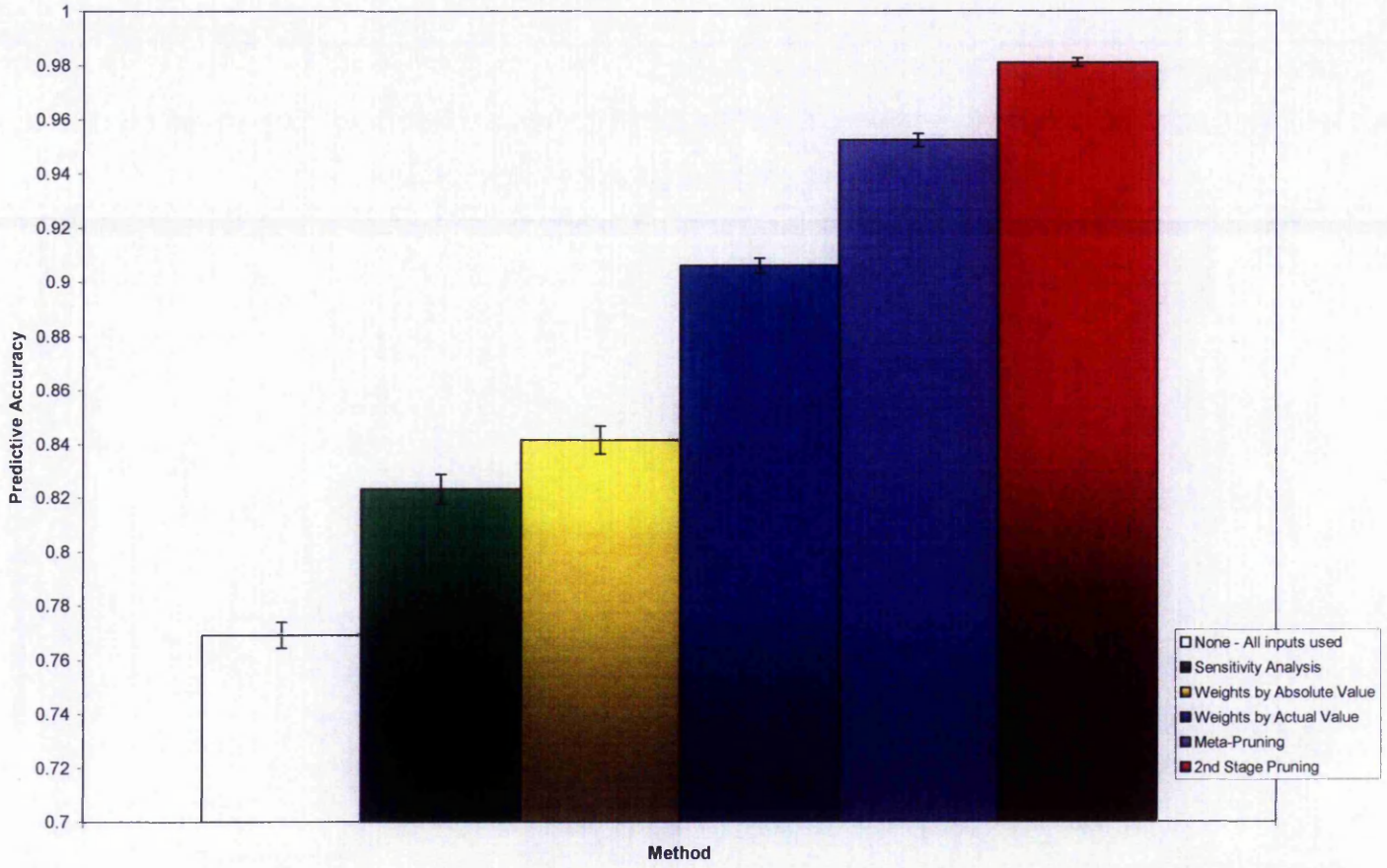


Figure 22. Performance results for the different pruning approaches across all three data splits using the *N. gonorrhoeae* dataset. Models were trained across 50 random sample cross validation data splits.

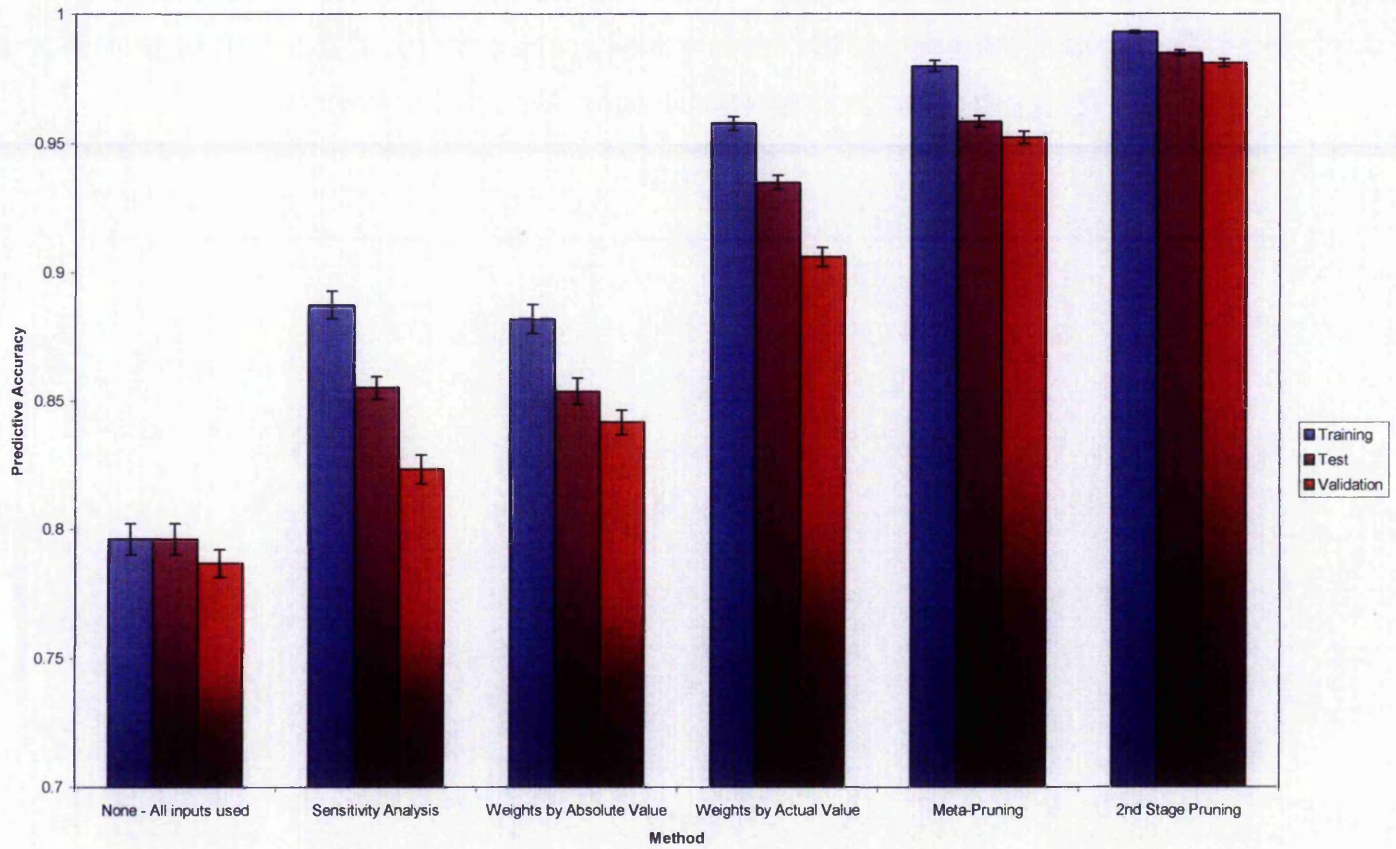


Figure 23. Overall median accuracy of the different pruning methods. Results are shown for validation data across 50 random sample cross validation data splits using the melanoma data.

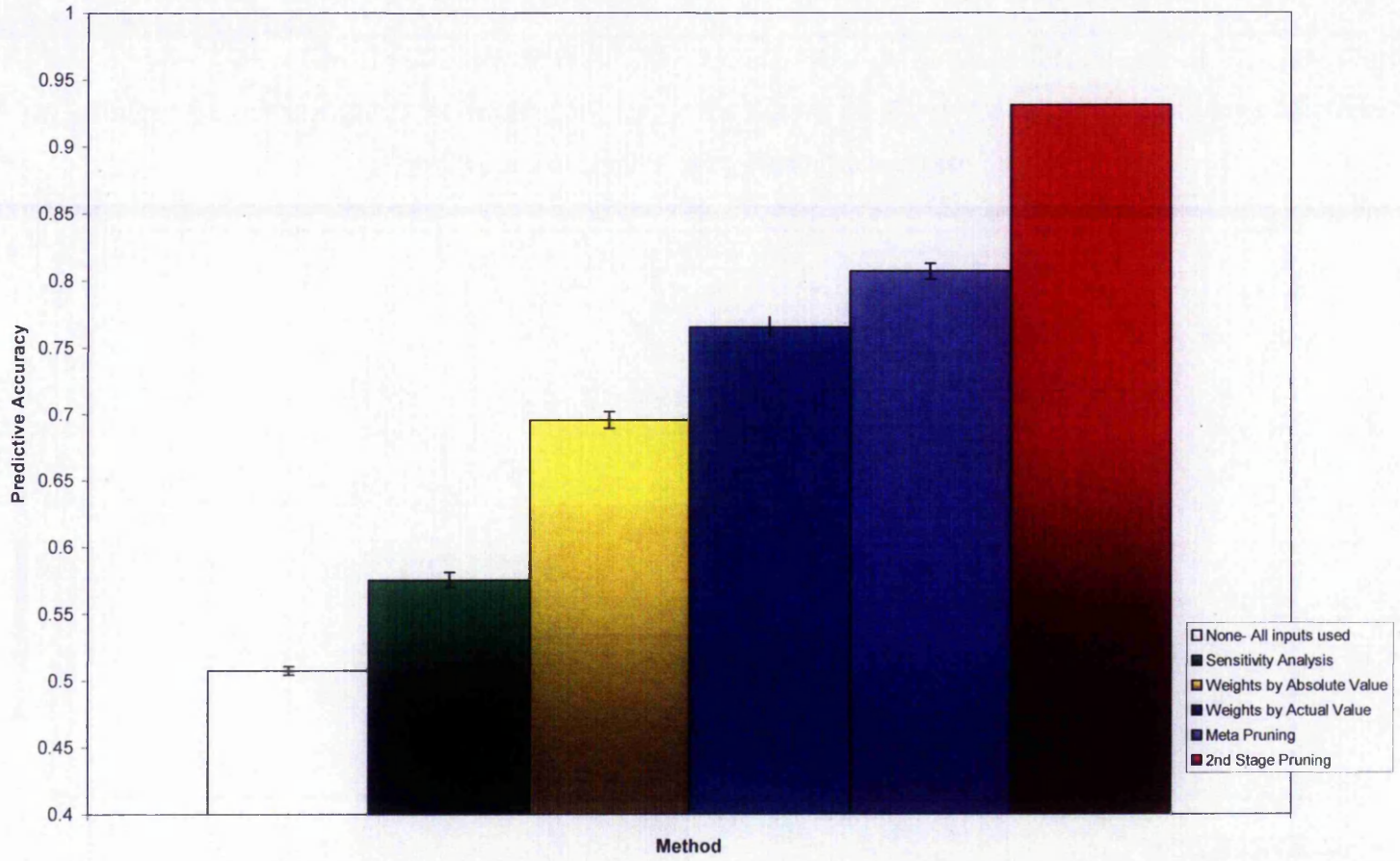
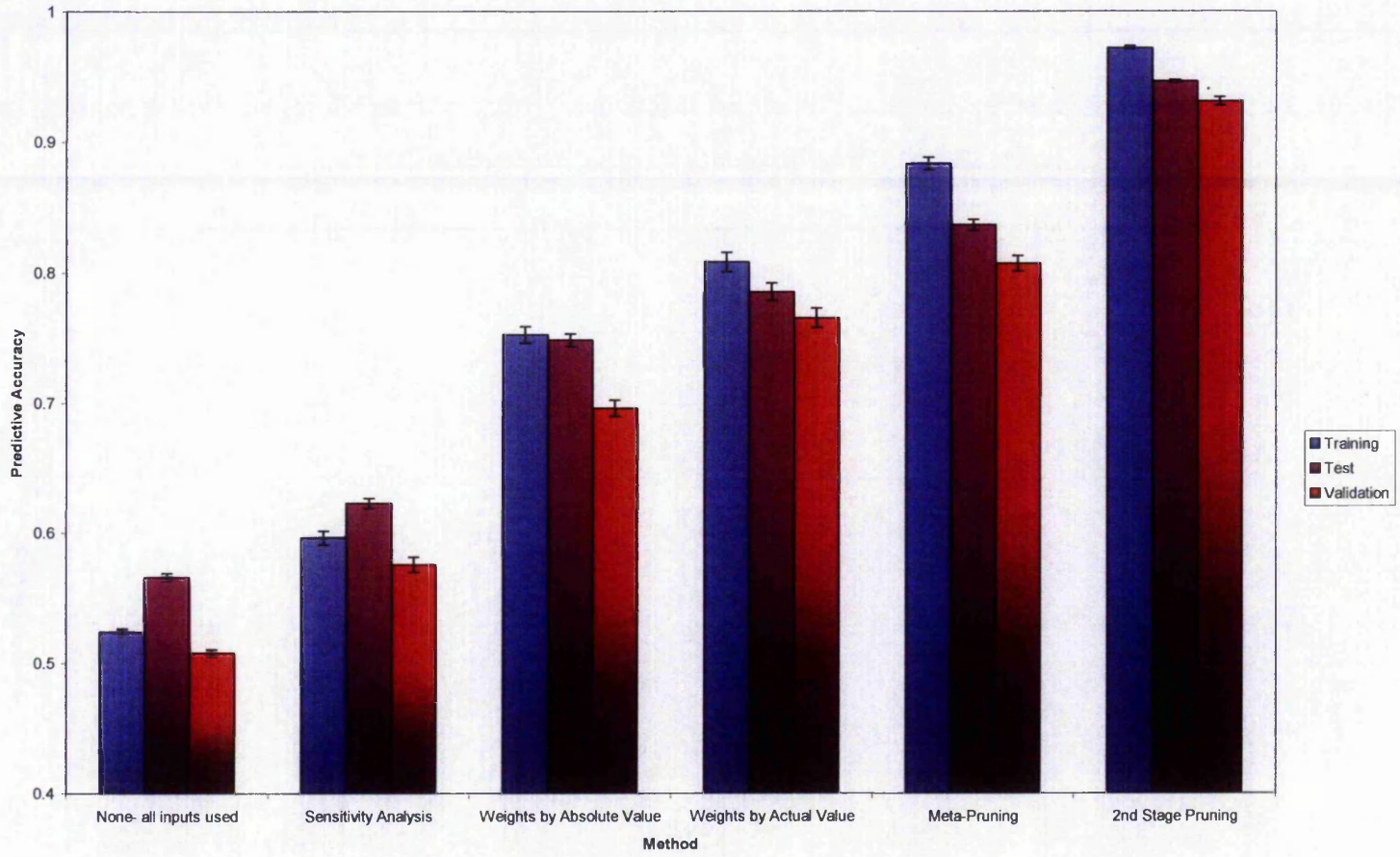


Figure 24. Performance results for the different pruning approaches across all three data splits using the melanoma dataset. Models were trained across 50 random sample cross validation data splits.



2.4. Summary and discussion

It is important to be able to reduce the complexity of datasets in order to identify components of importance which may then be used as biomarkers indicative of a specific class for identification, prognosis and diagnostic purposes. This chapter has assessed various methods in which to achieve this using ANNs. This was achieved firstly by examining the initial network parameter settings of the ANN architecture, and then by assessing whether the absolute weights of trained ANN models could be used for parameterisation purposes to identify subsets of ions whos phenotypic "fingerprints" correlated strongly to a particular class. Finally a comparison of several input pruning approaches leading to the assessment of a proposed "meta-pruning" approach to identify biomarkers which are consistently of a high importance across a number of methods.

2.4.1. Parameter reduction

The initial parameterisation methodology using weightings analysis was applied to a relatively large proteomic dataset derived from bacterial data, containing *N. meningitidis* and other closely related species. The ability to consistently and rapidly identify closely related bacterial species, such as those from the genus *Neisseria,* could pave the way for new technologies becoming more accessible to aid human judgment as an on-site laboratory based decision tool. This model parameterisation allows for the identification of molecular ions which are important in classifying the population into their respective groups. Thus, by determining which ions are contributing most in the classification, unimportant and noisy ions may be removed. Using the "rolling input subset" approach described earlier, the number of ions used as inputs in the ANN model were reduced from an initial set of approximately 13,000 (between 3 and 30 kDa.) to just the top 1,000 which could accurately predict species type. For a system to have practical application in a diagnostic laboratory where there is a high throughput of samples and a demand for rapid analysis, a simple application tool is required, therefore, it was decided that the number of ions needed to be reduced to the smallest number possible which could discriminate between the two species. In order to achieve this, the top 1,000 ions were ranked in order of importance in predicting a bacterial strain identity, and the top 100 were selected for further training. This resulted in a model which out-performed the previous. This process was repeated until the models ceased to improve in predictive capabilities resulting in a model containing just 30 molecular ions which could predict with a sensitivity of 100 %

and specificity greater than 96 % on a second order validation data set, which had not been used in any way in the training or development of the model. Therefore this approach of analysing the interconnecting weights of the trained network can be used as a rapid and efficient means to greatly reduce the number of inputs in a model in order to decrease complexity, whilst increasing predictive capabilities as a result of removing noisy inputs, which may be causing a conflict in the model and reducing predictive performance.

Prior to analysis, the ANN does not assume any prior relationships between any inputs used in the model, thus it was interesting that the ANN did not appear to identify single peaks around given mass values, but found clusters of ions around masses which were important in the classification. Because of the mass accuracy of the instrument (around 0.2 %) and data averaging, these clusters are highly likely to represent the same molecular species. However this could only be confirmed with further research. The scope of this study was then expanded to investigate why the ANN incorrectly classified the few samples that it did. This was done mainly by cluster analysis, but also incorporated PCA and a similarity analysis, in which samples were examined with regards to their raw mass spectra in order to identify which other samples closely matched these. From the clustering approach it was clear that the majority of the samples that the ANN misclassified in the various models were actually in fact grouped together with samples from their ANN predicted class and not their true class which 16S rDNA methods identified them as. This was supported further by conducting a PCA analysis on the validation data set, which again showed similar findings. The *N. meningitidis* samples were all found to lie along a clear vector, together with the non-*N. meningitidis* samples that the ANN misclassified. A similarity analysis also resulted in the same, with only 6 of the 25 most similar samples belonging to the "other" species group, whilst 19 of the 25 were *N. meningitidis.* This supports the assumption that the samples are actually more closely related to the ANN predicted class, than the class they were initially placed in, providing valuable information about the nature of the two populations and the outliers which are found to be exhibiting characteristics typical of both species. This advocates the proposition that there may exist a continuum between the two species, and serves to highlight the need for rapid, highly robust methods, which are capable of generalisation and can prove to be a reliable decision support tool for identification purposes.

It has been shown that parameterising ANNs based upon their trained weights can be successfully used as a system in which to develop predictive models by identifying

molecular biomarkers whose proteomic profiles correlate strongly to a particular class. 30 ions were identified which were capable of identifying 184 out of 188 samples from a separate validation data set correctly. Furthermore, it was shown using cluster analysis and principal components analysis that these few misclassified samples may actually show more resemblance to their ANN predicted class than to their actual class. Of these 30 ions identified, the majority of these appeared in clusters around certain mass values. As such there is a high probability that these belong to the same molecular species, suggesting that of these 30 ions identified, there may only be around 10 individual species present. All models were validated using a RSCV approach so that each sample was treated as blind a number of times. This was enhanced further by applying the models to an additional data set consisting of 188 samples that had not previously been used in the training of the model in any way. The high classification accuracies for this data set (greater than 96 %) reinforce the assumptions that a generalised, robust model has been created. This part of the study has therefore shown that this paramaterisation approach is powerful enough to model and classify for complex datasets and identify potential markers representative of different output classes.

2.4.2. Advanced parameterisation

Next, to build upon the findings outlined above, this section has compared this type of absolute weightings analysis to two of the most popular methods used for input pruning in ANNs. Here, all of these methods were shown to reduce the complexity of the models by eliminating unimportant inputs and reducing the dimensionality of the data space, whilst maintaining or even improving the overall predictive capabilities of the ANN. In addition to this, a novel meta-pruning approach has been proposed by which only inputs which are consistently important, *i.e.* only those which are classed as important across all methods are retained in the model, thus reducing the dimensionality of the model even further, and producing robust models which show a significant improvement in predictive performance over two very differing data sets derived from proteomic analysis.

In both datasets, using all of the initial inputs in the model resulted in the worst predictive performance, highlighting the vast complexity of the system and the need for data and noise reduction. This was followed by sensitivity analysis, then weightings analysis by absolute values. Weightings analysis using the actual connection weights values produced the more accurate models from the three methods compared, which concurs with findings

by Olden et al. (Olden, et al., 2004) who also concluded that the connection weights method to be the most accurate of the methods they compared. However, our meta-pruning approach significantly out performed the traditional methods in both datasets. It is interesting to note that out of the 940 inputs identified from the *N. gonorrhoea* dataset, a students t-test demonstrated that 55 % of these inputs showed a significant ($p < 0.05$) difference in their raw mass intensity profiles between the two classes. Similarly, within the melanoma dataset, 58 % of the 964 inputs identified as key predictors with the meta-pruning approach also showed statistically significant differences between groups. Moreover, this was extended to a second order meta-pruning analysis, where the inputs identified initially were subjected again to this parameterisation to further reduce the number of inputs, leading to an increase in model performance with fewer inputs. 57 % and 53 % of these for the *N. gonorrhoea* and melanoma datasets respectively showed statistically significant differences in their intensities between groups. These inputs may represent ions or proteins present in the system which are showing a true up- or down-regulation with regards to sample class, and may form the basis for future studies.

This study leads to the conclusion that ANNs are indeed not black boxes, and valuable information regarding variable importance is readily accessible by utilising the methods discussed. This also leads into the ability to understand the relationships between variables and how they contribute to the output. The usefulness of the meta-pruning approach developed here is apparent by the significant gain in predictive performance over the other methods described. This approach also leads to a more robust and parsimonious system which represents a solution applicable to new unseen data sets. The inputs selected in the model as the most effective predictors were derived from a consensus approach where they were classed as important across several methods therefore increasing the confidence that these inputs may represent key potential biomarkers whose proteomic profile which correlates to and is strongly representative of a particular class. Using this meta-pruning approach, a more parsimonious solution is found which also results in improved predictive performances for blind data. Additionally, false detection pitfalls are also avoided. Here, only inputs that are consistently important over a number of sub-models, over a number of methods are considered for further training. Therefore the likelihood of an input being considered as important when it truly is not is lower using this approach than when using just one method. In addition to this new development, ANNs in conjunction with mass spectrometry show their potential use in microbiology for the rapid analysis of bacterial

strains to locate the origin of pathogenic organisms from specific outbreaks, which has previously not been shown.

CHAPTER 3. BIOMARKER IDENTIFICATION

3.1 Introduction

Proteomic and genomic technologies have revolutionised the way in which biological research is carried out, having the potential to simultaneously map and analyse tens of thousands of genes or proteins in any given system. Computational analysis of such data may aid in the identification of genes, proteins or gene/protein signatures which correspond to prognostic outcome in patients with diseases such as cancer (Adib, et al., 2004; Beer, et al., 2002; Bhattacharjee, et al., 2001; Jiang, et al., 2002; Lee, et al., 2004; Masters and Lakhani, 2000; Michiels, et al., 2005; Porkka, et al., 2002; Rosenwald, et al., 2002; van 't Veer, et al., 2002; West, et al., 2001). Genotypic, and subsequently phenotypic traits, determine cell behaviour and, in the case of cancer, govern the cells' susceptibility to treatment. Since tumour cells are genetically unstable, it is likely that sub-populations of cells arise that assume a more aggressive phenotype, capable of satisfying the requirements necessary for invasion and metastasis (Fidler, 2003). Thus the detection of biomarkers indicative of tumour aggression should be apparent, and consequently their identification is considered of paramount importance for early disease diagnosis, prognosis and response to therapy.

However, there are problems associated with the computational analysis of such extreme volumes of data, known as the curse of dimensionality. This refers to the exponential growth of data as a function of its dimensionality. This curse of dimensionality often leads to an input space with many irrelevant or noisy inputs, subsequently causing predictive algorithms to behave badly as a result of them modelling extraneous portions of the space. Therefore ultimately, if computational power permitted, analysis based upon assessing all potential input combinations to determine an optimal subset of predictive biomarkers would potentially provide further understanding of biological systems in the disease or condition of interest.

Unfortunately this type of analysis is currently unfeasible in these datasets due to the processing time that would be required in analysing each potential input combination that exists in such a large input space. For example, a gene microarray dataset routinely contains in excess of 20,000 variables per sample and would therefore require assessment of over 10^{6000} possible input combinations to determine *the* optimum expression signature.

Given the limitations but obvious predictive advantages of this approach an alternative method was developed for determining the optimal genomic/proteomic signature within a realistic time period that does not require excessive amounts of time or processing power. This approach utilises ANNs and involves sequentially selecting and adding input neurons to the network (where each input neuron is equivalent to a single gene/ion from the experiment together with its expression value or intensity) to identify an optimum biomarker subset based on predictive performance and error. Three datasets were used to test and validate this approach. The first interrogates human serum samples with varying stages of melanoma. The samples were analysed by MALDI-TOF MS by colleagues at Nottingham Trent University from samples collected by collaborators at the German Cancer Research Centre. The remaining two were publicly available datasets which both originated from gene expression data derived from breast cancer patients.

The first dataset was produced by MALDI-TOF MS analysis of protein and peptide data derived from clinically defined melanoma serum samples. The aims here were to firstly compare healthy control patients with those suffering from melanoma at the four different clinical Stages, I, II, III and IV, in order to identify biomarker ions indicative of stage. Secondly, adjacent stages were to be analysed comparatively with in the aim of identifying potential biomarkers representative of disease progression.

The second dataset, published by van't Veer et al. (van 't Veer, et al., 2002), used microarray technology to analyse primary breast tumour tissue in relation to development of metastasis. The authors generated data by gene expression analysis in a cohort of 78 breast cancer patients, 34 of which developed distant metastases within five years, and 44 which remained disease free after at least five years. Each patient had 24,482 corresponding variables specifying the expression ratio of a single known gene or expressed sequence tag (EST).

The third dataset publish by West et al. (West, et al., 2001) used microarray technology to firstly analyse primary breast tumours in relation to oestrogen receptor (ER) status and secondly to assess whether the tumour had spread to the axillary lymph node (LN), providing information regarding metastatic state. This dataset consisted of 13 ER+/LN+ tumours, 12 ER-/LN+ tumours, 12 ER+/LN- tumours, and 12 ER-/LN- tumours. Each sample had 7,129 corresponding gene expression values. The approach described here was then validated using a second dataset (Huang, et al., 2003) which was made available by

the same group as the first, and contained a different population of patients, ran on a different microarray chip.

3.2 Stepwise approach methodology

3.2.1. ANN architecture

As previous, the ANN modelling used a supervised learning approach, MLP architecture with a sigmoidal transfer function, where weights were updated by a BP algorithm. Learning rate and momentum were set at 0.1 and 0.5 respectively, as previously defined. Prior to training, the data were scaled linearly between 0 and 1 using minimums and maximums. This architecture utilised two hidden nodes in a single hidden layer and initial weights were randomised between 0 and 1.

3.2.2. ANN model development

The same approach was applied across all datasets, with the only differences being the number of samples and input variables. Here, as an example the methodology as applied to the van't Veer dataset will be described. Data from the microarray experiments was taken in its raw form. This consisted of 78 samples each with 24,482 corresponding variables specifying the expression ratio of each single gene. Prior to training each model the data was randomly divided into three subsets; 60 % for training, 20% for testing, and 20 % for validation, so that RSCV could be carried out.

Initially, each gene from the microarray dataset was used as an individual input in a network, thus creating n (24,482) individual models. These n models were then trained over 50 randomly selected subsets and network predictions together with MSE values for these predictions were calculated for each model with regards to the separate validation set. The inputs were ranked in ascending order based on the mean squared error values for the test data and the model which performed with the lowest error was selected for further training. Thus 1,224,100 models were trained and tested at each step of model development.

Next, each of the remaining inputs were then sequentially added to the previous best input, creating n-1 models each containing two inputs. Training was repeated and performance evaluated. The model which showed the best capabilities to model the data was then selected and the process repeated, creating n-2 models each containing three inputs. This process was repeated until no significant improvement was gained from the addition of

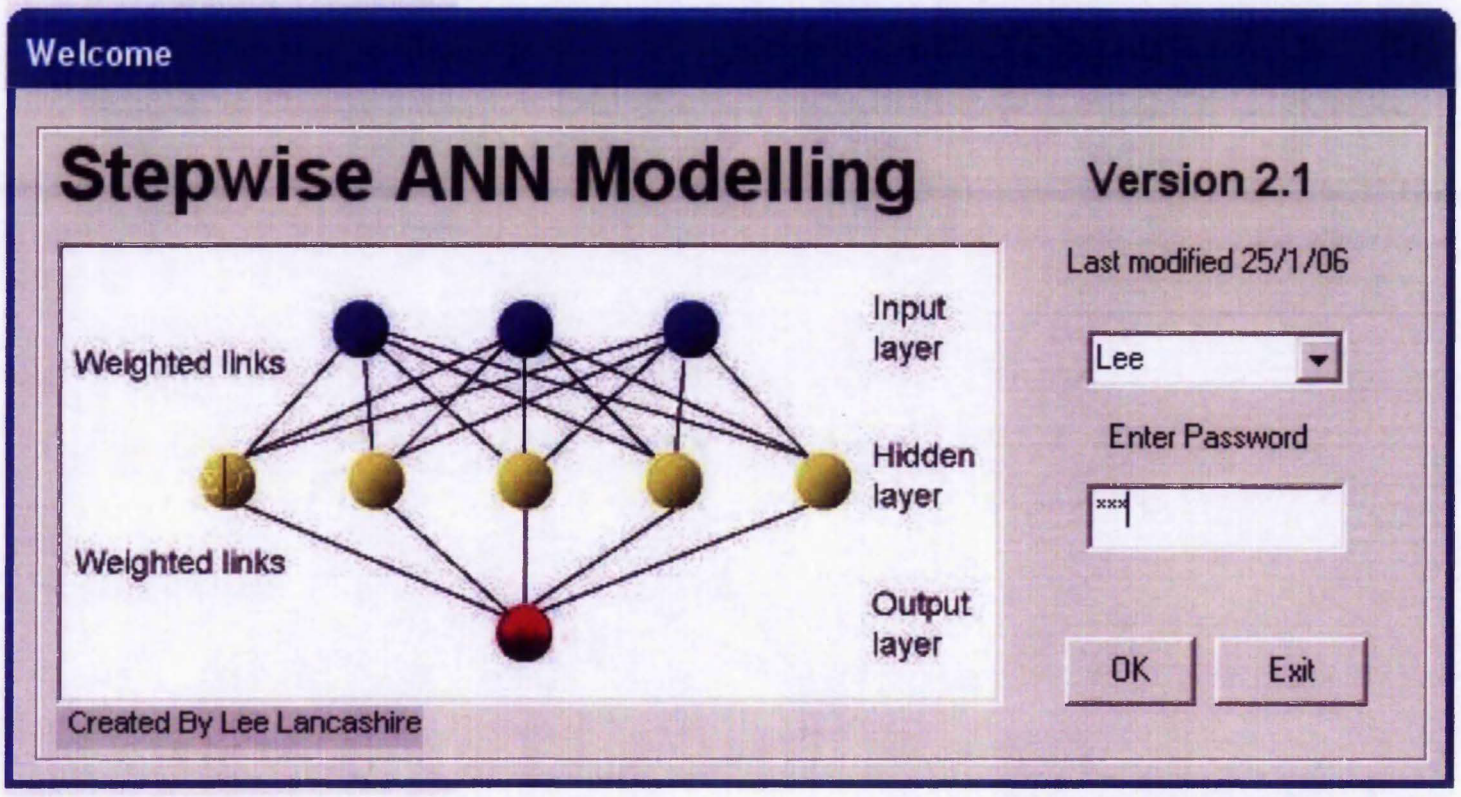
further inputs resulting in a final model containing the gene expression signature which most accurately modelled the data. The biomarkers identified at each step are ordinal to one other within the data space so that the continual addition of inputs explains variation within the data that was not explained by the previous step.

This process requires the training and testing of potentially millions of models. To facilitate this, software to automate the procedure was been created using Microsoft Visual Basic. Here, the inputs are added automatically, selecting the best contender biomarkers at each step. Figure 25a-g shows the software design detailing the various options available for ANN design and analysis. The full source code can be found in the Appendix, and the entire process for running the algorithm can be summarised in Figure 26 and below:

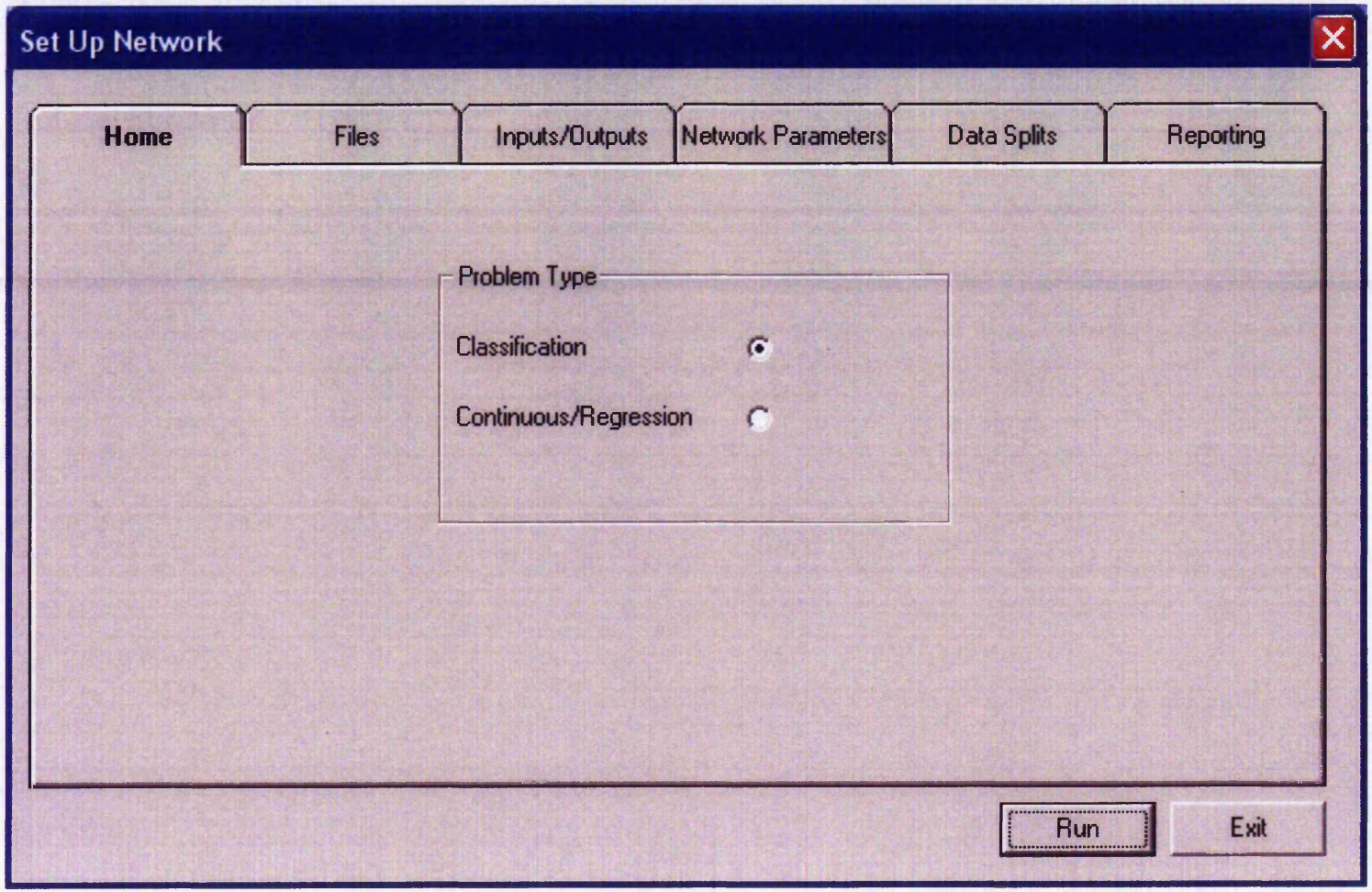
1. Identify input and output variables
2. Start with input 1 as the first input to the model, $input_1$
3. Train the ANN using RSCV
4. Record network performance for $input_1$
5. Repeat steps 3 and 4 using all inputs; $input_2 \ldots input_3 \ldots input_4 \ldots input_n$ as sole inputs in the ANN model
6. Rank inputs in ascending order based on the error on the test data split to determine the best performing input at this step, $input_i$
7. Repeat from step 2, using each input sequentially with $input_i$ in an ANN model
8. Determine the best performing input combination for this step
9. Repeat this whole process from step 2, continually adding inputs until no improvement is gained from the addition of further inputs

Figure 25a-g. Stepwise ANN modelling software. Each diagram represents a different option screen available within the software f
model building and analysis.

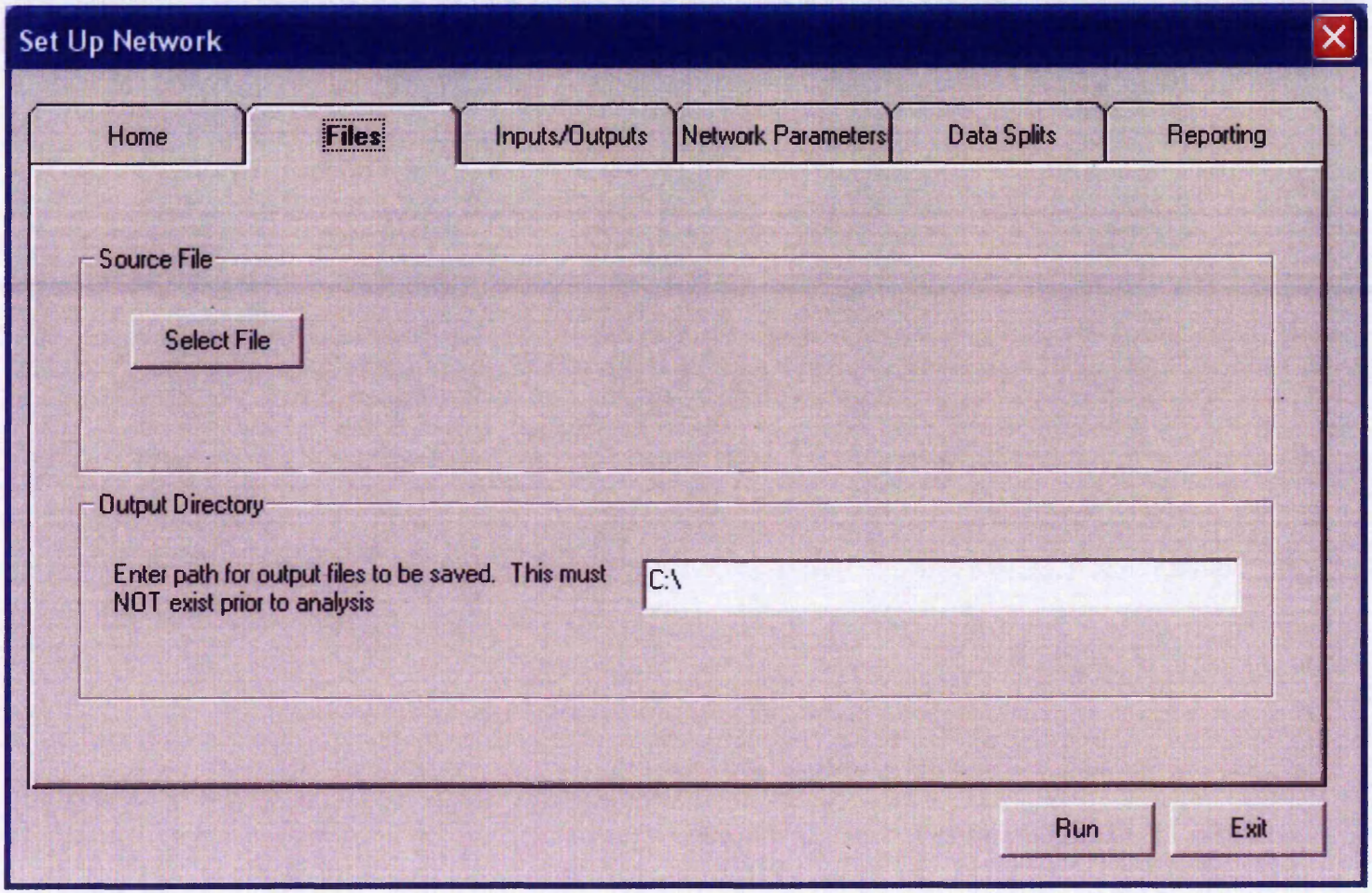
(a)



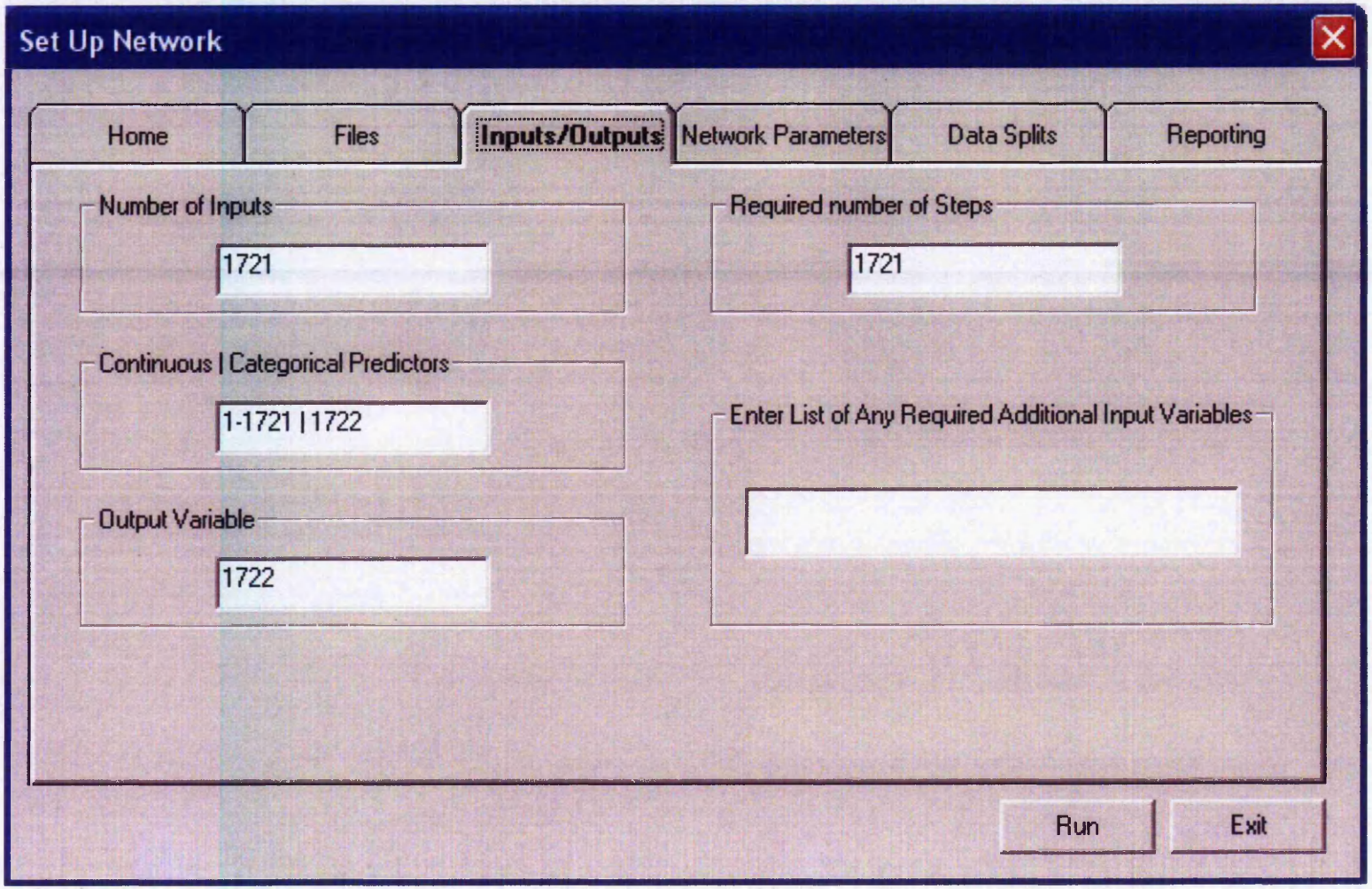
(b)


Set Up Network
Home
Files
Inputs/Outputs
Network Parameters
Data Splits
Reporting
Problem Type
Classification
Continuous/Regression
Run
Exit

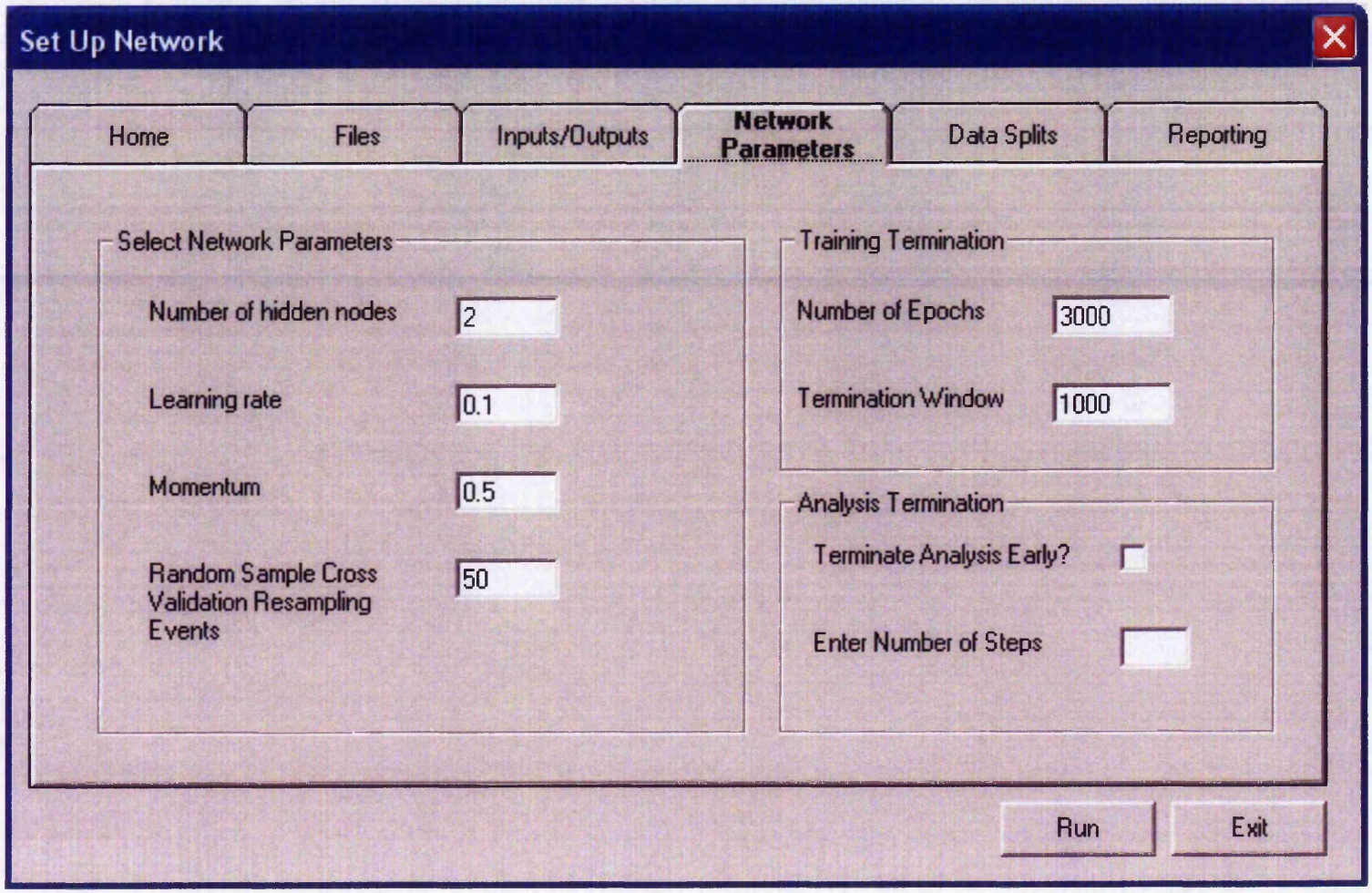
(c)


Set Up Network
Home
Files
Inputs/Outputs
Network Parameters
Data Splits
Reporting
Source File
Select File
Output Directory
Enter path for output files to be saved. This must NOT exist prior to analysis
C:\
Run
Exit

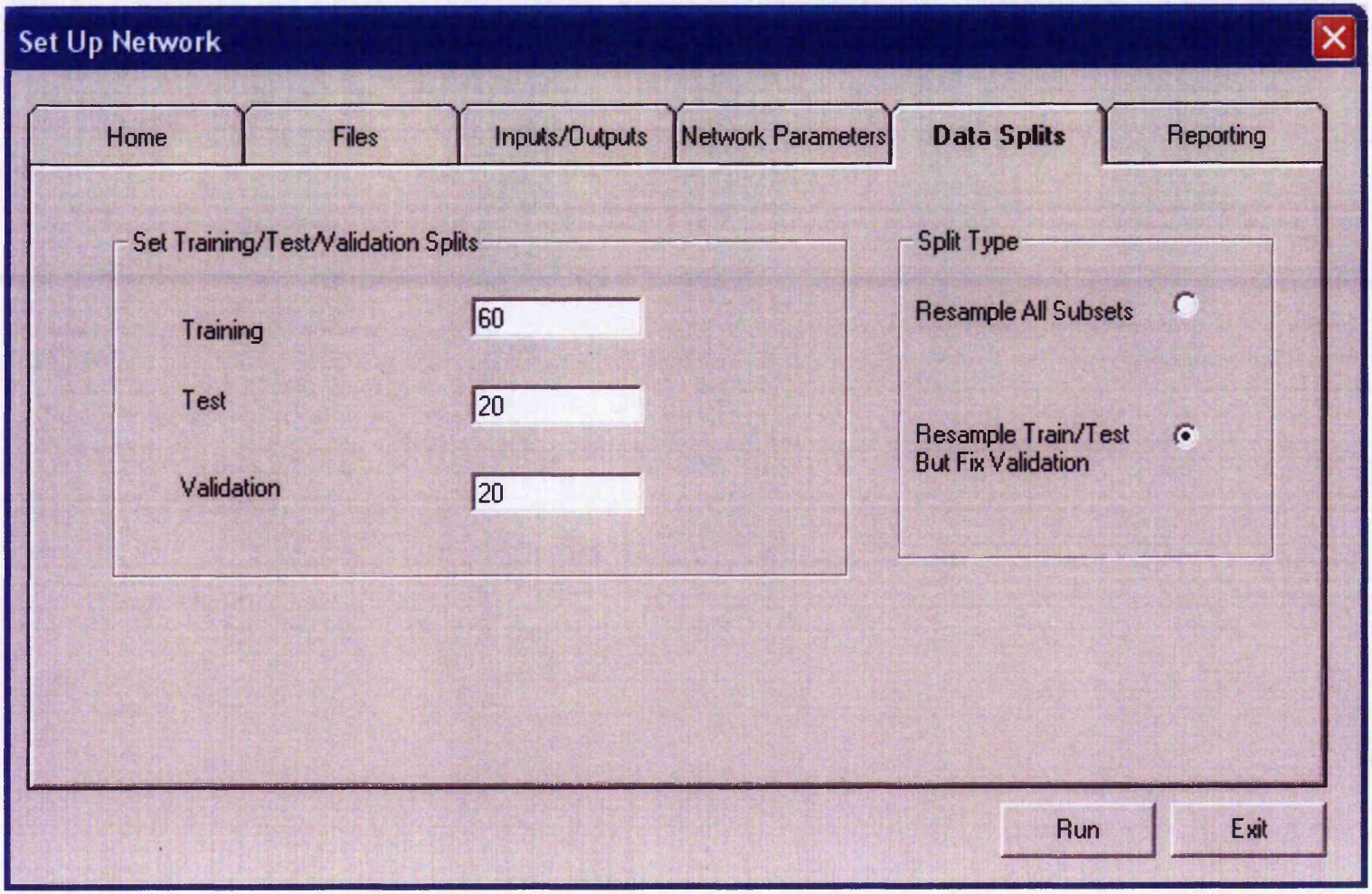
(d)


Set Up Network
Home
Files
Inputs/Outputs
Network Parameters
Data Splits
Reporting
Number of Inputs
1721
Required number of Steps
1721
Continuous | Categorical Predictors
1-1721 | 1722
Enter List of Any Required Additional Input Variables
Output Variable
1722
Run
Exit

(e)


Set Up Network
Home
Files
Inputs/Outputs
Network Parameters
Data Splits
Reporting
Select Network Parameters
Number of hidden nodes
2
Learning rate
0.1
Momentum
0.5
Random Sample Cross Validation Resampling Events
50
Training Termination
Number of Epochs
3000
Termination Window
1000
Analysis Termination
Terminate Analysis Early?
Enter Number of Steps
Run
Exit

(f)


Set Up Network
Home
Files
Inputs/Outputs
Network Parameters
Data Splits
Reporting
Set Training/Test/Validation Splits
Training
60
Test
20
Validation
20
Split Type
Resample All Subsets
Resample Train/Test But Fix Validation
Run
Exit

(g)

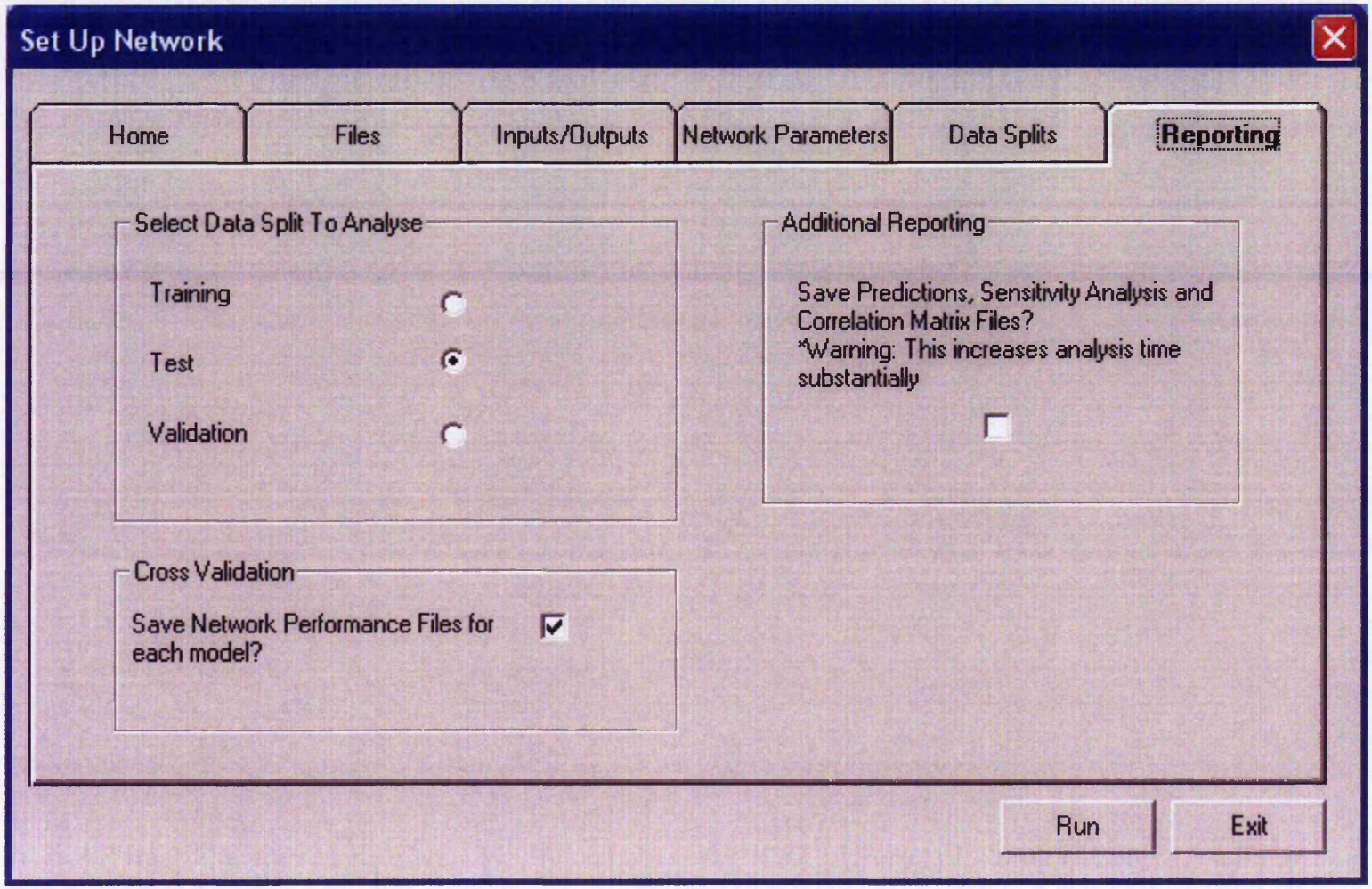
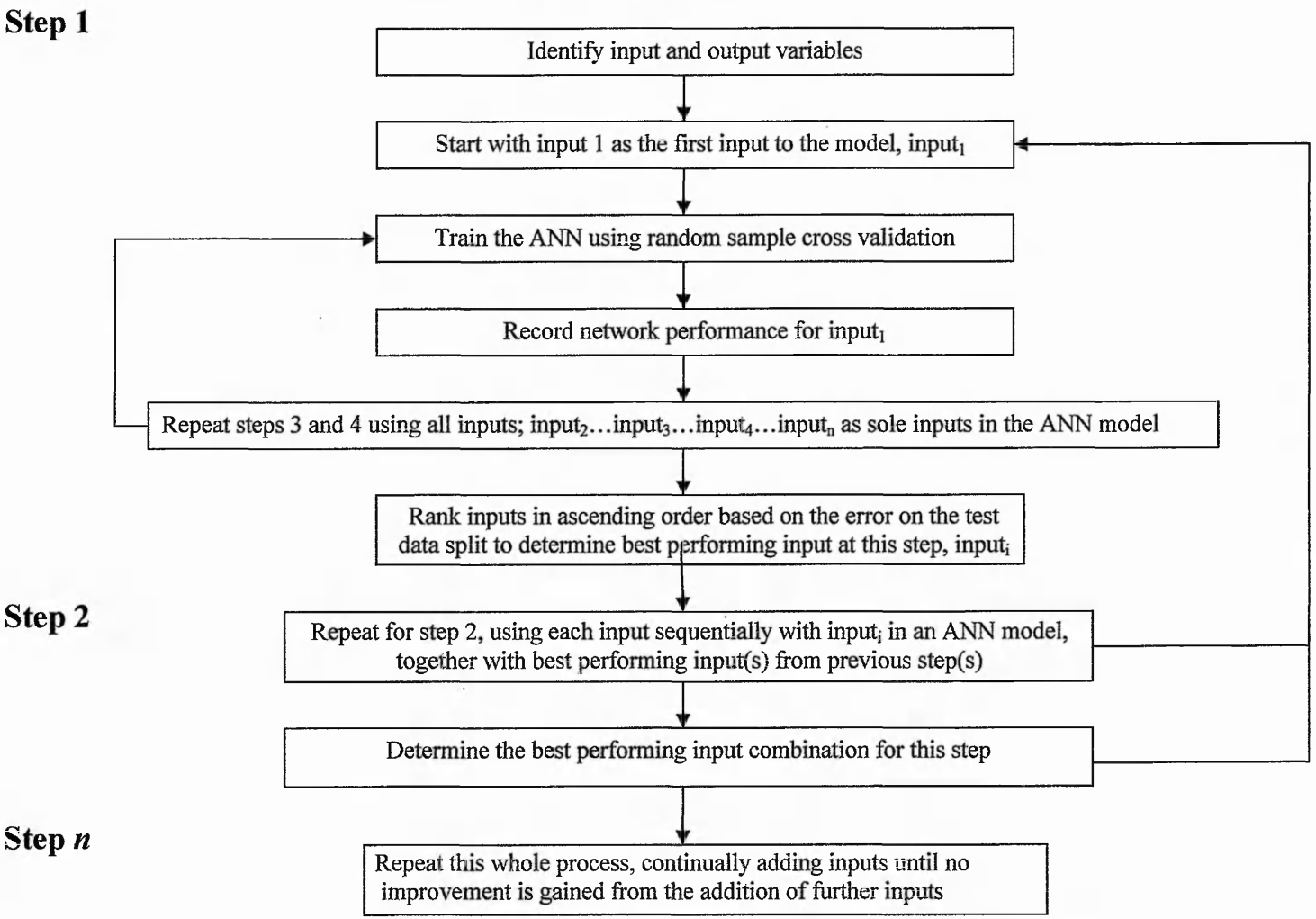

Set Up Network
Home
Files
Inputs/Outputs
Network Parameters
Data Splits
Reporting
Select Data Split To Analyse
Training
Test
Validation
Additional Reporting
Save Predictions, Sensitivity Analysis and Correlation Matrix Files?
*Warning: This increases analysis time substantially
Cross Validation
Save Network Performance Files for each model?
Run
Exit

Figure 26. Stepwise algorithm summary.



3.3. Results

3.3.1 Analysis of melanoma dataset

3.3.1.1. Analysis of control and Stage IV disease samples: Protein and peptide data

Because there are no confirmatory blood markers for metastatic melanoma, in conjunction with colleagues at Nottingham Trent University we sought to develop a validated, robust and reproducible MALDI MS methodology using the same stepwise ANN approach to profile serum protein and tryptically digested peptides. This was applied to data derived from MALDI MS analysis representing (i) protein and (ii) digested peptide data from the control and diseased samples. The MALDI MS data in its raw format was binned over a 1 Da. range, so that each input used in subsequent modelling represented the intensity of an ion spanning a single mass unit range. Various analyses were carried out on these datasets in order to identify biomarker ions indicative of the classes shown in Table 2.

Table 2. Summary of analyses conducted.

Analysis	Class 1	Class 2
Protein ion analysis 1	Healthy Control	Stage IV melanoma
Tryptic peptide ion analysis 1	Healthy Control	Stage IV melanoma

Biomarker patterns containing 9 ions from the protein data and 6 ions from the digested peptides were identified, which when used in combination correctly discriminated between control and Stage IV samples for protein and peptide data to median accuracies of 92.3 % (inter-quartile range 89.4 – 94.8 %) and 100 % (inter-quartile range 96.7 – 100 %) respectively. Table 3a-b shows the performance for the models at each step of the analysis for the protein and peptide data. This shows that with the continual addition of key ions, an overall improvement was seen for both the error associated with the predictive capabilities of the model for blind data, and also the median accuracies for samples correctly classified. Nine biomarker ions was determined to be the most effective subset of biomarker ions producing the best model performance for the protein data as no significant improvement was seen in predictive performance with the addition of further ions. No further steps were conducted beyond step 6 for the peptide data because after this step no significant improvement in performance could be achieved. Therefore these models were considered to contain a subset of ions representing either the proteins or digested peptides which most accurately modelled the data. Figure 27 shows the error and performance

progression for the peptide data when using the stepwise approach for biomarker identification.

Table 3a. Summary of Stage IV vs control protein ions identified at each step of the analysis. Ions belonging to the optimal biomarker subset are highlighted in red.

Step	Ion Added	Median Accuracy (%)	Inter-Quartile Range
1	12000	64.1	58.7-69.2
2	14847	73.2	69.8-75.8
3	1649	80.4	77.4-83.3
4	15477	80	77.9-84
5	13255	82.7	79.1-85.2
6	3031	83.8	79.8-86.1
7	4791	87	83.9-90.4
8	9913	86.6	83.2-89.8
9	4835	92.3	89.4-94.8
10	15269	90.4	87.2-92.6
11	2730	90.3	87.1-92.2
12	9919	90.4	87.3-92.5
13	9971	91.9	88.3-94
14	11735	90.4	87.1-92.5

Table 3b. Summary of Stage IV vs control digested peptide ions identified at each step of the analysis.

Step	Ion Added	Median Accuracy (%)	Inter-Quartile Range (%)
1	1753	77.8	74.4-83.2
2	1161	93.3	90.2-96.4
3	1505	93.7	92.4-96.7
4	854	96.7	95.8-100
5	1444	100	96.5-100
6	1093	100	96.7-100

3.3.1.2. Analysis of digested peptide data: Diseased Stages I, II, III and control samples

Next, because the analysis of the peptide data provides the potential for subsequent protein identification, it was decided that these peptide MALDI-TOF MS profiles would be analysed in the search for differential biomarker ions which would be representative of firstly disease stage (by analysing the individual stages against control populations) and secondly disease progression (by generating predictive models classifying between adjacent disease stages). The analyses conducted in this part of the study are summarised in Table 4.

Initially, in order to identify ions which were representative of disease stage, the stepwise approach was applied to identify subsets of biomarker ions which could predict between disease stage and control samples. This would therefore provide valuable information concerning which peptide ions were showing differential intensities that were specific to the disease stage of interest. Table 5 shows the biomarker subsets identified in each model, and their median performance when predicting validation subsets of data over 50 RSCV resampling events. Figure 28 shows the stepwise analysis summary across all of the models for each step of analysis. As expected, the models predicted Stage I v control with the least accuracy (80 %), suggesting that because early stage disease is a non-penetrating skin surface legion, changes occurring in the serum at the protein level are less significant than at advanced stages of disease. Nonetheless, to be able to predict Stage I melanoma to accuracies of 80 % using serum would be viewed as clinically significant. It was interesting to note that of the biomarker ions identified by this approach, in several instances the same ions were occurring across different models. Ions 1299 and 3430 (3432) were found to differentiate between both Stage I and Stage II disease vs control samples. Ions 1251 and 1283 (1285) were found to differentiate between Stage II and Stage III disease vs control, whilst ion 1753 (1754) was identified in both the Stage III and Stage IV diseased vs controlled models. Considering that 3500 individual ions are trained and tested at each step of analysis over 50 RSCV resampling events, it seems unlikely that their consistent identification as the most important ions at a given step would be a consequence of chance, providing confidence that these ions are representing proteins which are showing a true change in intensity in patients with disease at differing stages.

Table 4. Summary of analyses conducted.

Analysis	Class 1	Class 2
Tryptic peptide ion analysis 2	Healthy Control	Stage I melanoma
Tryptic peptide ion analysis 3	Healthy Control	Stage II melanoma
Tryptic peptide ion analysis 4	Healthy Control	Stage III melanoma
Tryptic peptide ion analysis 5	Stage I melanoma	Stage II melanoma
Tryptic peptide ion analysis 6	Stage II melanoma	Stage III melanoma
Tryptic peptide ion analysis 7	Stage III melanoma	Stage IV melanoma

Figure 27. Stepwise summary of ions added at each step of the analysis of digested peptide data: Stage IV melanoma v Control. Blue line represents mean squared error value at each step with 95 % confidence intervals being shown as error bars. Red line represents median model accuracy at each step of analysis with inter-quartile ranges being shown as error bars.

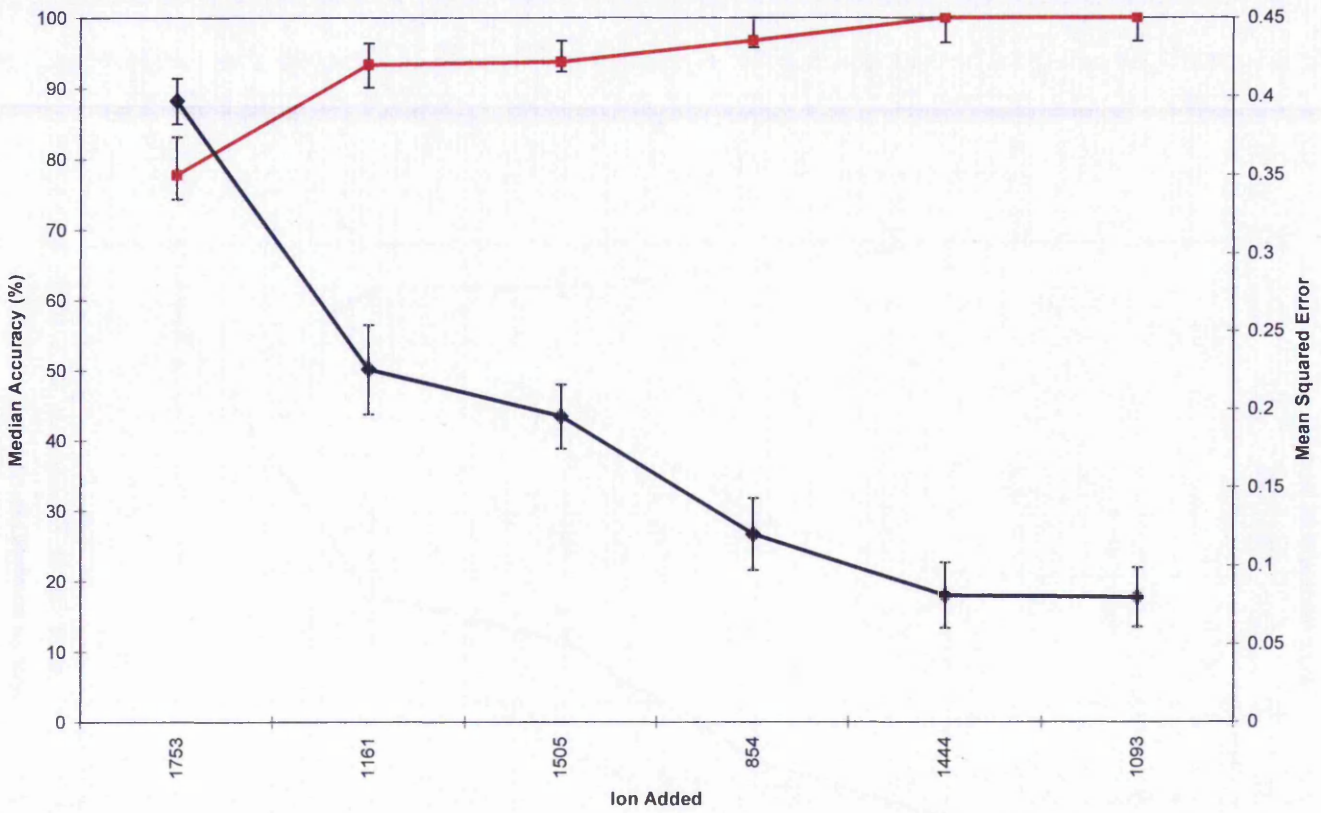


Table 5. Summary of overall results from digested peptide analysis: Stages I,II,III, and IV vs Control. Peptide ions highlighted in red represent ions corresponding to multiple groups.

Dataset Modelled	Ions identified	Median Performance for Validation data (%)
Stage I v Control	864, 933, 980, 1299, 2309, 2886, 2966, 3220, 3430, 3489	80
Stage II v Control	1251, 1283, 1299, 1968, 2244, 2411, 3432, 3443	96.5
Stage III v Control	1251, 1285, 1312, 1371, 1754, 2624, 2715, 2999, 3161, 3326	91.7
Stage IV v Control	854, 1093, 1161, 1444, 1505, 1753	100

3.3.1.3. Analysis of adjacent diseased groups

Once biomarker ions representative of individual disease stage had been determined, it was decided important to analyse adjacent group stages of disease, which would potentially identify biomarker ions which would represent those responding differently as disease progressed, and would be predictive and indicative of disease stage. Table 6 shows the biomarker subsets identified in each model, and their median performance when predicting validation subsets of data over 50 RSCV resampling events. It was interesting to find that subsets of ions could be identified which were able to predict between stages to extremely high accuracies; 98 % for Stage I v Stage II and 100 % for Stage II v Stage III and Stage III v Stage IV. Furthermore, only two peptide biomarker ions were required in order to perfectly discriminate between Stage II and Stage III, with one of these ions, 903, also being important in the classification of Stage III v Stage IV, suggesting that this ion is potentially of importance in disease progression to advanced stages, and appears to be downregulated as melanoma stage advances from Stage II to IV, which could only be confirmed by further studies.

Figure 28. Overall summaries of stepwise model performance of diseased groups v control samples.

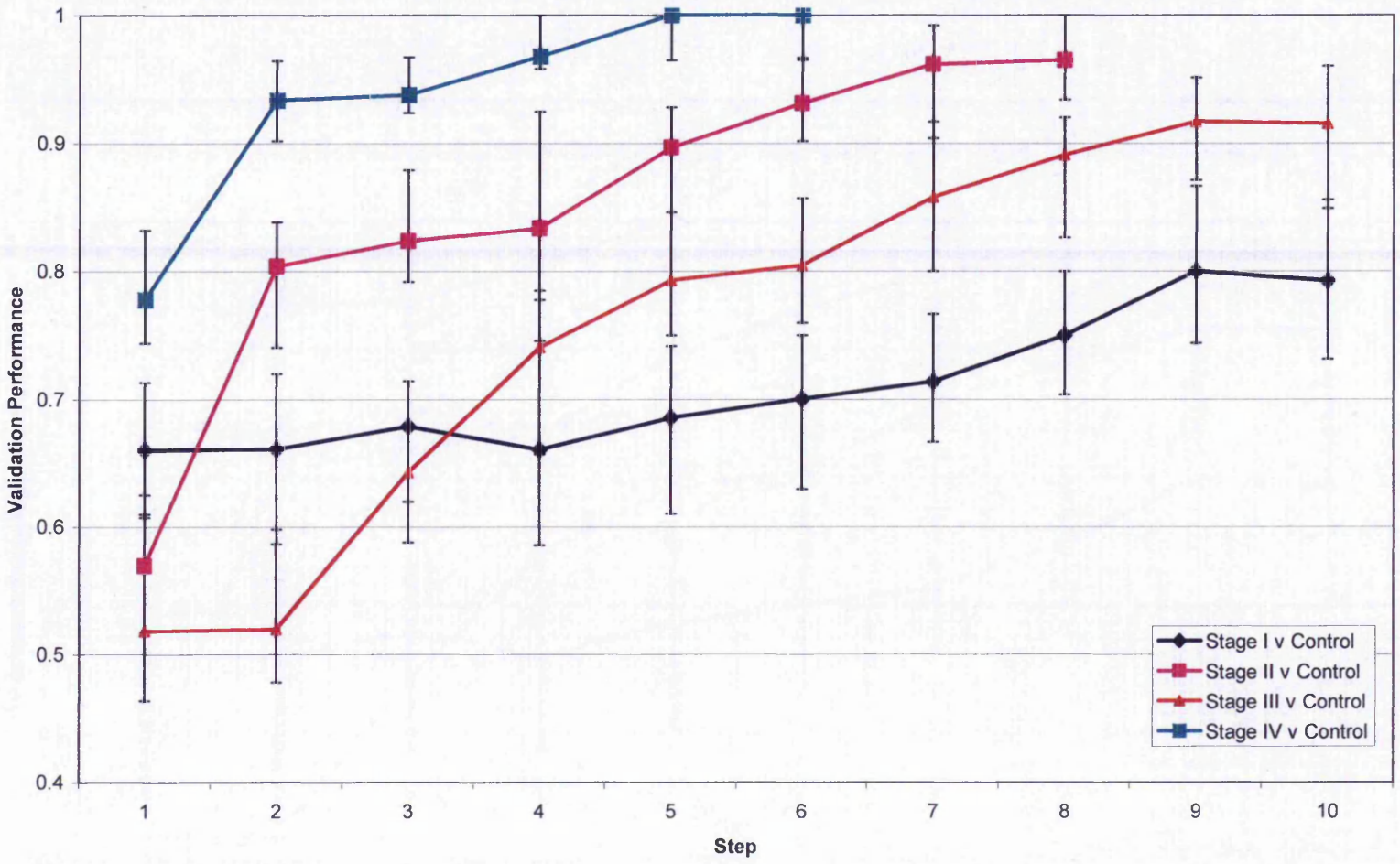


Table 6. Summary of overall results from digested peptide analysis: Adjacent stage analysis. Peptide ions highlighted in red represent ions corresponding to multiple groups.

Dataset Modelled	Ions identified	Median Performance for Validation data (%)
Stage I v Stage II	1251, 1731, 1825, 1978, 2053	98
Stage II v Stage III	861, 903	100
Stage III v Stage IV	877, 903, 1625, 2064, 2754	100

The overall summaries for the stepwise analysis conducted here can be seen in Figure 29. For visualisation of the feature space that these samples are occupying, and to understand the decision surface that these models are generating, PCA was conducted using the subset of ions identified by the ANN stepwise approach. Figure 30a-c shows the PCA for the Stage I v Stage II, Stage II v Stage III and Stage III v Stage IV models respectively. It is evident that when using the biomarker ions identified by ANNs the samples can be separated into distinct clusters using PCA, with the clearest separation being with the Stage II v Stage III model. It is interesting to draw attention to the samples highlighted by arrows and circles in the Stage I v Stage II model (Figure 30a). The first of these samples was identified as a Stage I sample, but according to its profile PCA has placed it more indicative of Stage II.

Interestingly, the ANN model also predicted this sample as a Stage II sample, suggesting it has strong features corresponding more to a Stage II sample than a Stage I sample which it was categorised as by the clinicians. Similarly, the region of samples highlighted on Figure 30b which appear to be lying on the border of the decision surface were also predicted closely to the 0.5 decision threshold by the ANNs, again suggesting that these samples are showing characteristics of both classes according to their proteomic profiles.
The relative closeness in feature space of the Stage III and Stage IV samples according to Figure 30c suggests that the proteomic profiles of these samples are similar, and cannot be as clearly separated using the PCA as they are when using the ANN modelling, therefore requiring a nonlinear decision surface to correctly classify this cohort of samples which are at a more advanced diseased stage. Furthermore, the mean group intensities of these ions has been analysed, with the summary being shown in Figure 31. This shows how the biomarker ions identified as most important in the discrimination of sample groups has

changed during the different stages of disease. It is clear from this that not all of these biomarker ions are being up regulated as disease progresses. All five of the ions identified in the Stage I v Stage II analysis show statistically significant ($p = <0.05$) increases in intensity. In the Stage II v Stage III model, both biomarker ions appear to be down regulated when disease is more advanced, with ion 861 significantly so. In the Stage III v Stage IV model, all ions (except for ion 2754) showed a significant increase or decrease in intensity as disease progressed, with ion 1625 showing a huge increase in intensity as disease progressed to Stage IV. A scatterplot was produced of the two ions (*m/z* 861 and 903) identified in the Stage II vs Stage III model, this can be seen in Figure 32. A clear separation of Stage II and Stage III samples is evident, with the Stage III samples clearly showing lesser levels of ion 861. This enables one to derive a hypothetical decision boundary between the two classes.

3.3.1.4. Model validation

To study the question about stability of this procedure over multiple experiments (Baggerly, et al., 2004), and to assess batch to batch reproducibility of the mass spectrometry analysis, both the proteins and peptides from the Stage IV and control samples were run by the group on two separate occasions and the results of the second experiment were used to validate the stepwise methodology. This dataset was obtained by a different operator and on a different date. The second sample set was then passed through the developed ANN models to blindly classify them as a second order of blind data for class assignment. For the protein data, the model correctly classified 85 % of these blind samples correctly. This consisted of 43/49 of the control samples and 40/49 of the cancerous samples being correctly identified, with sensitivity and specificity values of 82 and 88 % respectively, with an AUC value of 0.9 when evaluated with a ROC curve.

Figure 29. Overall summaries of stepwise model performance of diseased groups v control samples.

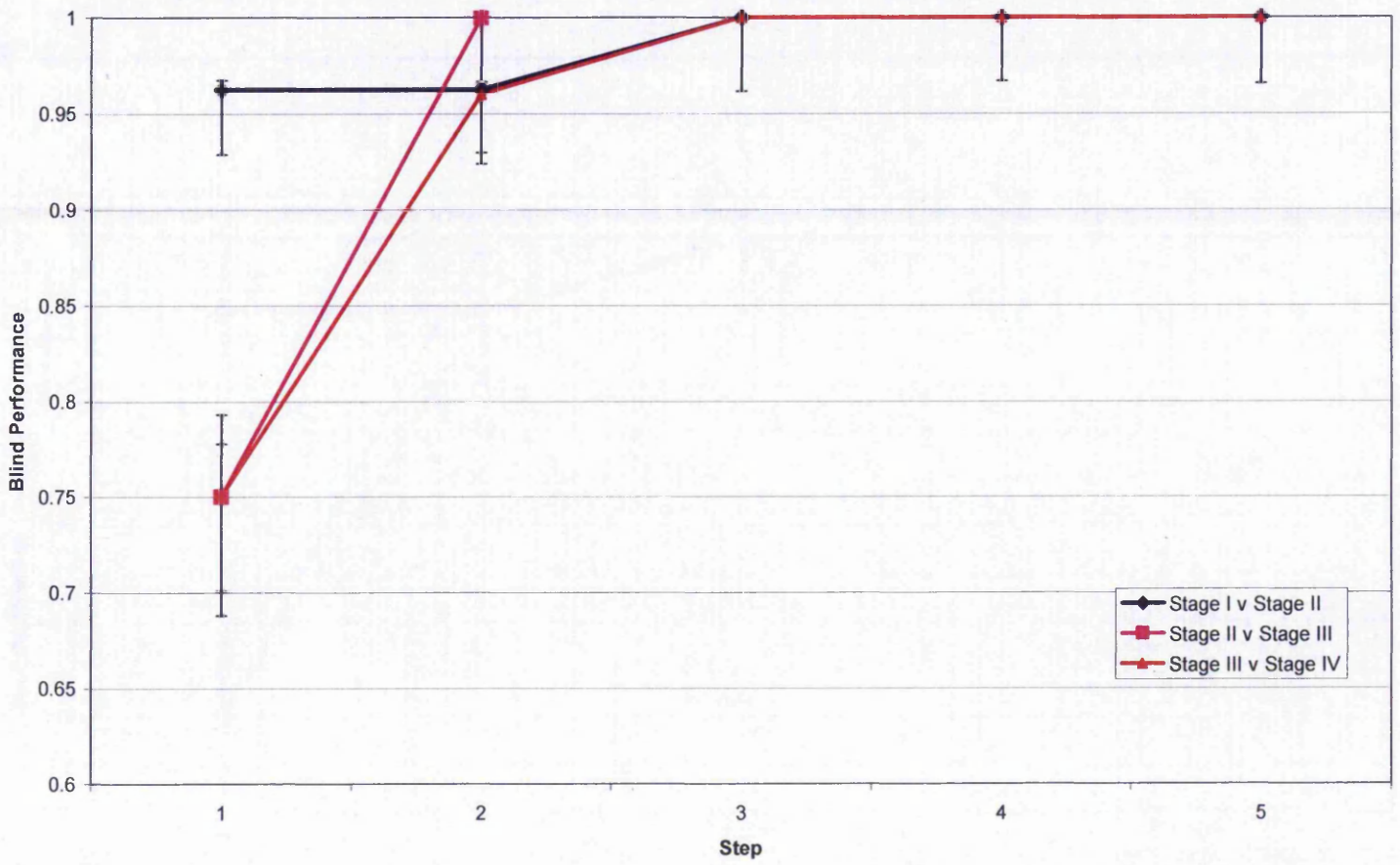
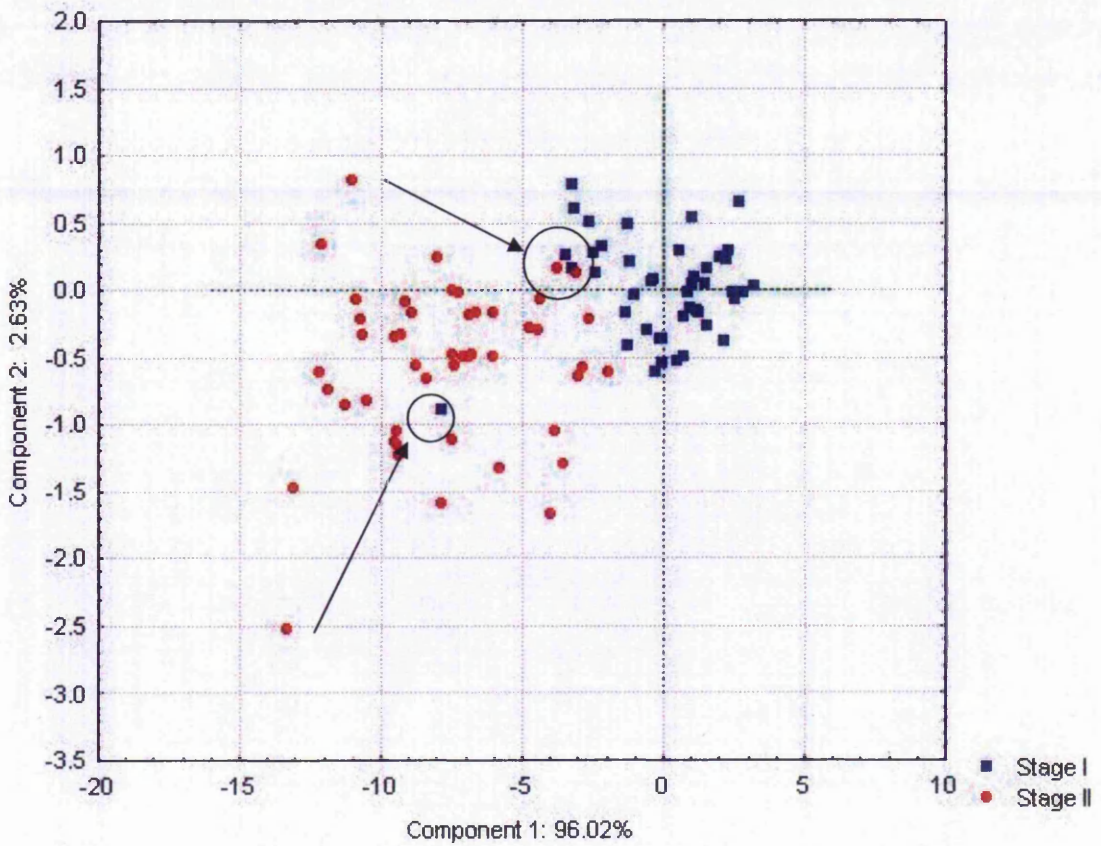
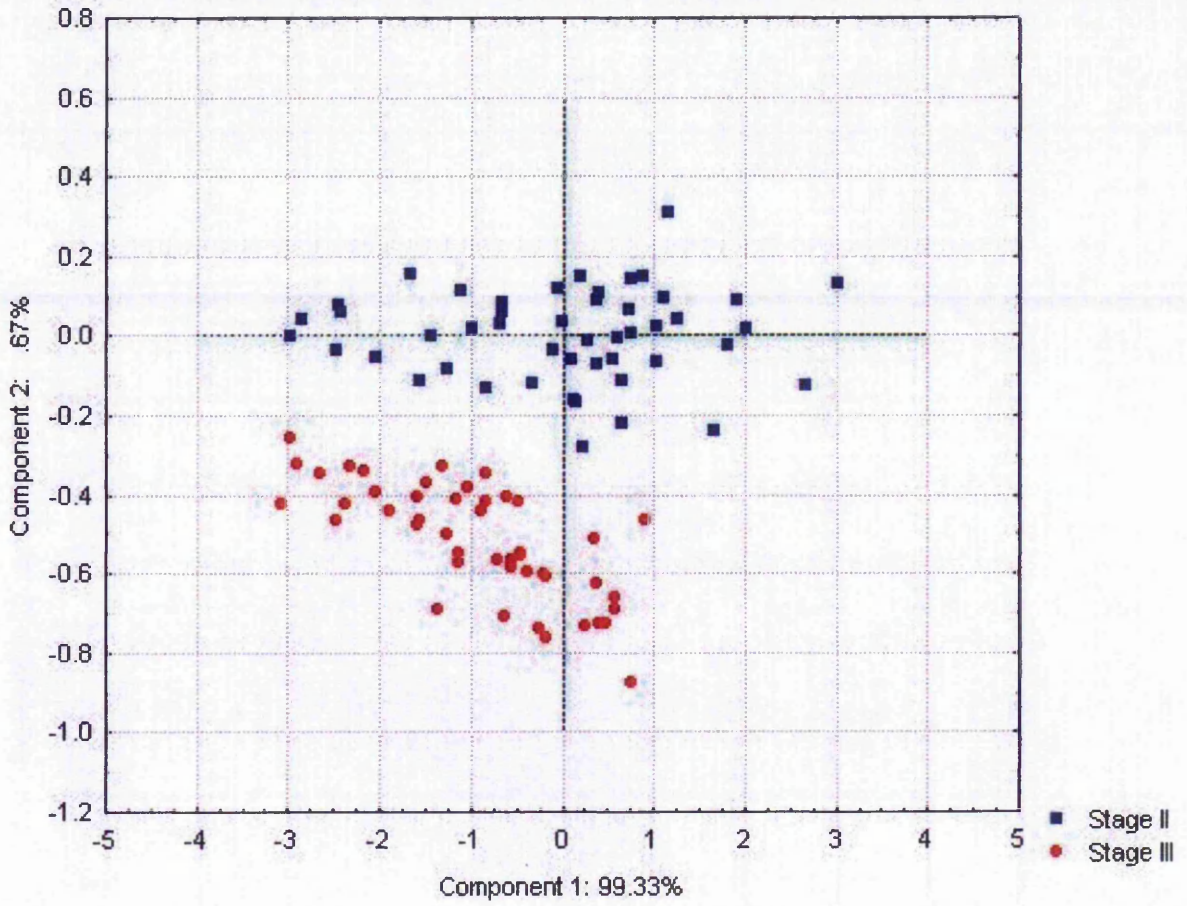


Figure 30a-c. Principal Components Analysis using the biomarker ions identified by ANN stepwise approaches. Samples groups are differentiated by colour and style.

(a)



(b)


0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
Component 2: .67%
-5
-4
-3
-2
-1
0
1
2
3
4
5
Component 1: 99.33%
Stage II
Stage III

(c)

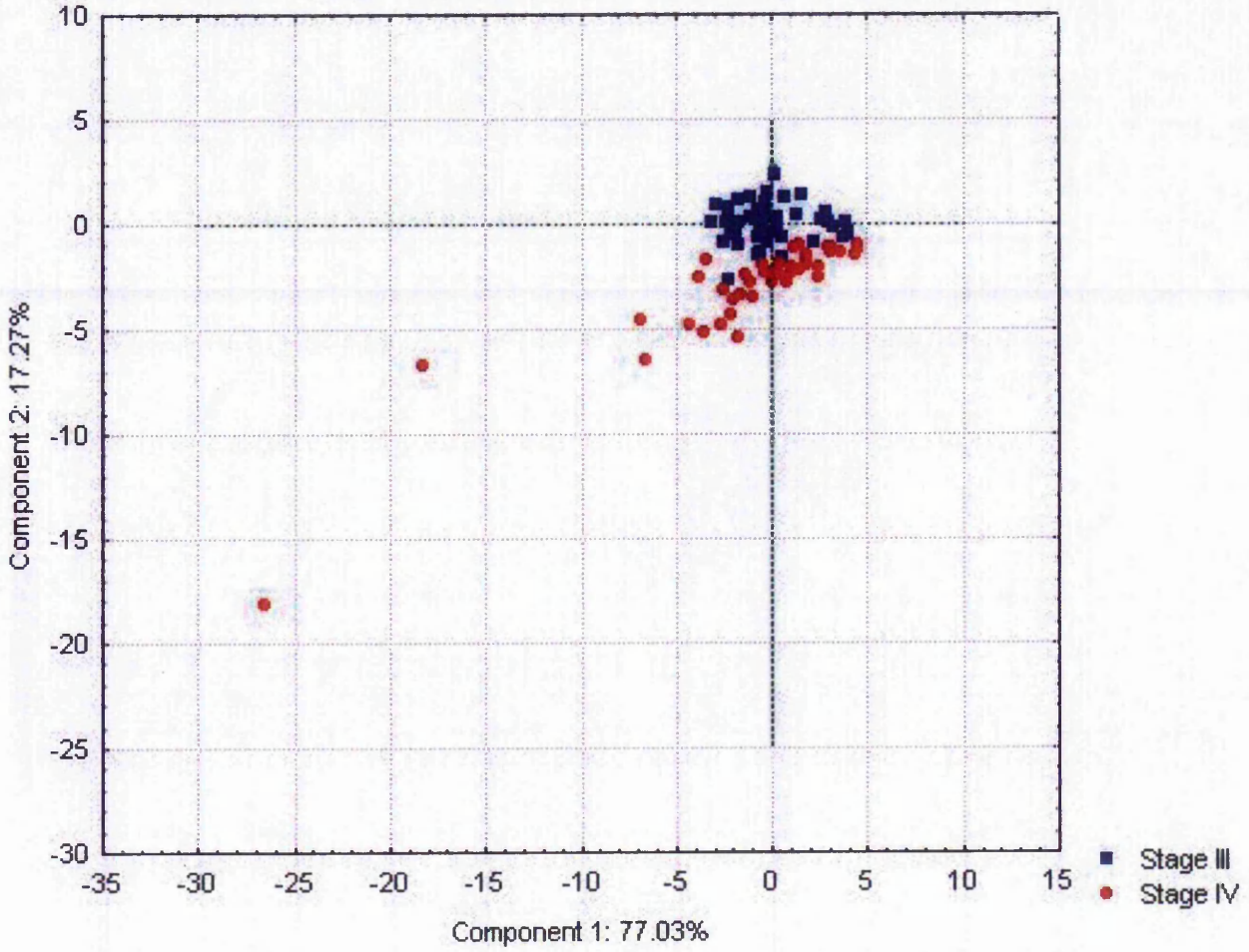

10
5
0
-5
-10
-15
-20
-25
-30
-35
-30
-25
-20
-15
-10
-5
0
5
10
15
Component 2: 17.27%
Component 1: 77.03%
Stage III
Stage IV

Figure 31. Mean group intensities of peptide biomarker ions identified by ANNs. All of the key biomarkers across the different stages are shown.

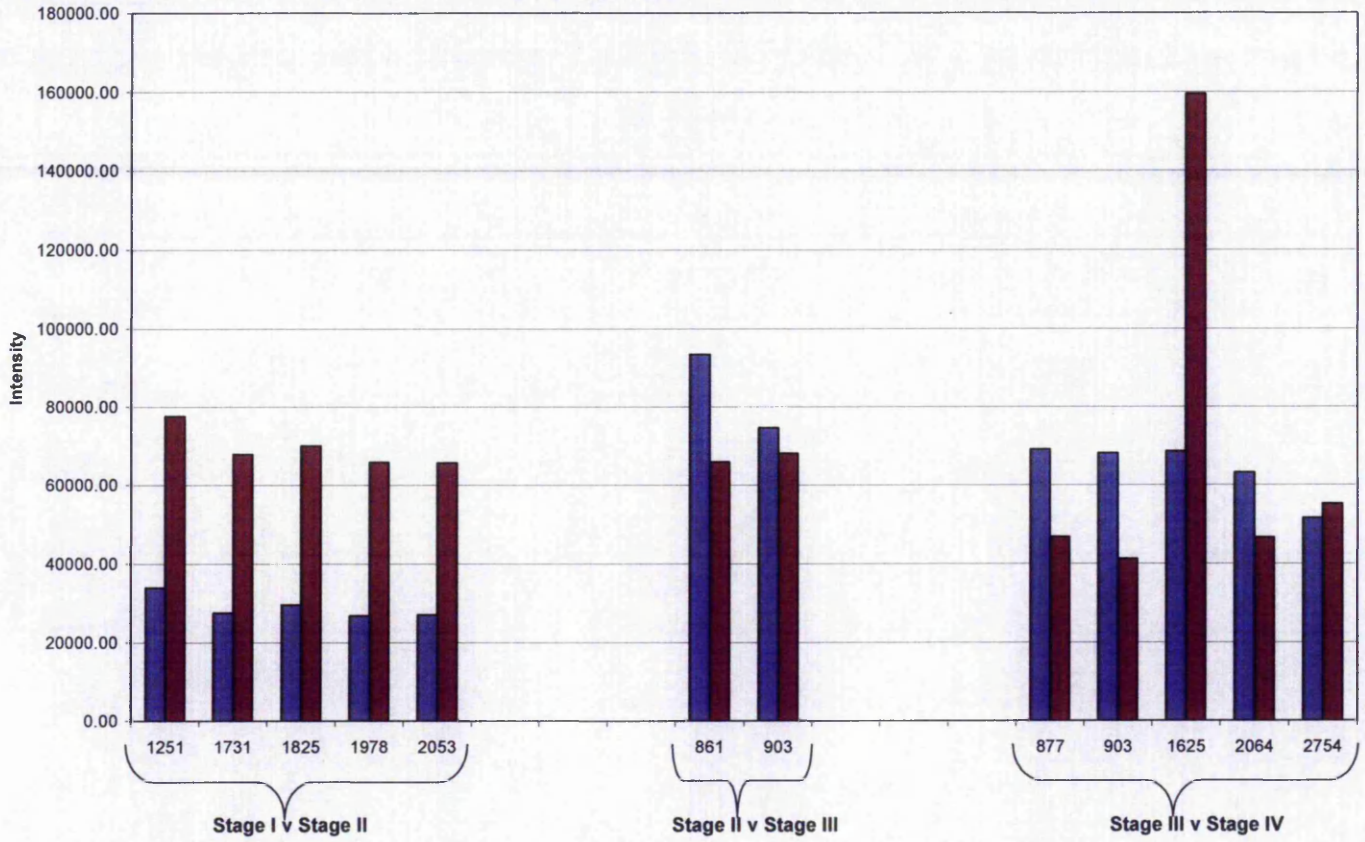
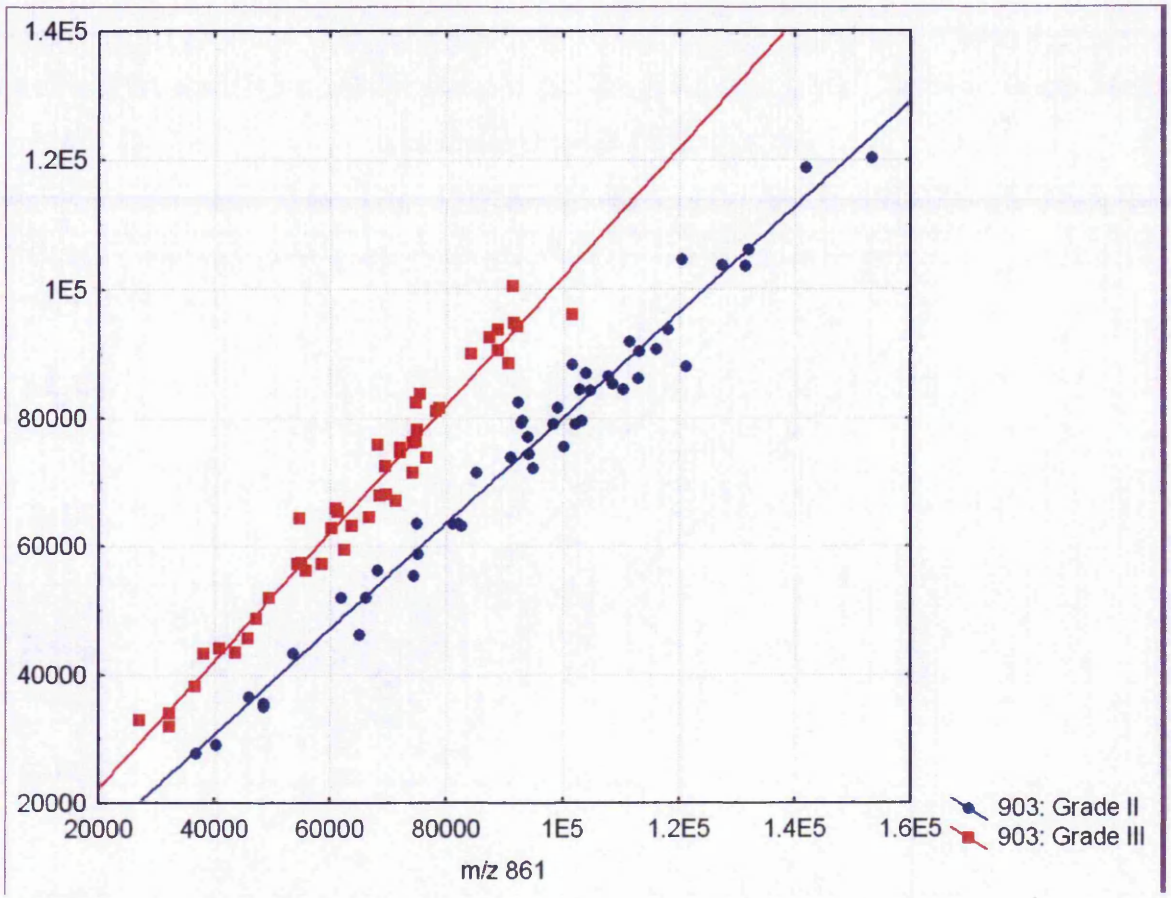


Figure 32. Scatterplot of ion 861 against ion 903 for Stage II and Stage III melanoma. Red squares indicate Stage III samples, whilst blue circles show Stage II samples.



For peptides, the model correctly classified 43/47 samples originating from control patients, and 43/43 samples from cancerous patients. This gave an overall model accuracy of 95.6 %, with sensitivity and specificity values of 100 and 91.5 % respectively, and an AUC value of 0.98. This suggests that the peptide data was more reproducible than the protein data for this second batch of mass spectrometry analysis. The predictive peptide ions were subsequently sequenced and identified by colleagues using a variety of mass spectrometric techniques leading to the identification of two proteins; Alpha 1-acid glycoprotein (AGP) precursor ½ and complement C3 component.

3.3.2. Analysis of van't Veer et al. breast cancer dataset

The aim of the analysis here was to utilise this novel stepwise ANN modelling approach in order to identify a gene expression signature which would accurately predict whether a patient would develop distant metastases within a five year time period, and thus identifying potential markers and giving an insight into disease aetiology. Following the rule of parsimony which suggests that the simplest model fitting the data should be used, an initial analysis was carried out using logistic regression (Subasi and Ercelebi, 2005). This method led to poor predictive performances with a median accuracy of just 53 % (inter-quartile range 47-61 %), suggesting that this dataset is not linearly separable.

The application of the stepwise approach resulted in the identification of a gene expression signature consisting of twenty genes which predicted patient prognosis to a median accuracy of 100 % (inter-quartile range 100-100%, mean squared error of 0.085), where samples were treated as blind data over 50 models with RSCV. The overall screening process assessed over ten million individual models. When evaluated with a ROC curve the model had an AUC value of 0.971 with sensitivity and specificity values of 98 % and 94 % respectively. Figure 33 shows the performance for the models at each step of the analysis. It is evident that the continual addition of key genes leads to an overall improvement in the predictive capabilities of the model. The model showed a decrease in performance at steps 10 and 11 which may be due to a possible interaction between the genes present at these steps with one or more of the other genes in the model. After this point the model improved further still until step twenty, so this was considered to contain the genes which most accurately modelled the data. Further steps were not conducted because no significant improvement in performance could be achieved. A summary of the

performances of the models at each step, together with the identity of the genes (where known) can be seen in Table 7.

To further validate the model, an additional set of 19 samples were selected, as in the original manuscript (van 't Veer, et al., 2002). This set consisted of 7 patients who remained metastasis free, and 12 who developed metastases within five years. The 20 gene expression signature identified here correctly diagnosed all 19 samples, further emphasising the models predictive power.

Table 7. Summary of twenty genes used in the gene expression signature at each step of model development.

Step	Gene Name	Gene Description	Median Accuracy (%)	Inter Quartile Range (%)	Mean Squared Error
1	CA9	Carbonic anhydrase IX	70	66.7-77	0.438
2		EST	80.5	77.7-87.7	0.383
3		EST	83	76.1-85.9	0.377
4	FLJ13409	EST	87	79.6-88.7	0.351
5	LCHN	LCHN protein	80	73.9-84.7	0.397
6	TMEFF2	Transmembrane protein with EGF-like and two follistatin-like domains 2	94.7	89.4-95.3	0.233
7	HEC	Highly expressed in cancer, rich in leucine heptad repeats	94.8	89.3-96.7	0.217
8	HSPC333	*Homo sapiens* HSPC337 mRNA, partial cds	96	95-100	0.171
9		EST	98.1	94.6-100	0.154
10		EST	95	90.9-95.9	0.23
11	HUGT1	UDP-glucose:glycoprotein glucosyltransferase 1	78.2	71.3-83.5	0.393
12	LOC56899	putative 47 kDa.	85.1	80-91.8	0.322

		protein			
13	DJ462O23.2	Hypothetical protein dJ462O23.2	96.1	94.3-100	0.16
14	HSU93243	Ubc6p homolog	96.1	95.2-100	0.155
15	NRG2	Neuregulin 2	95.8	94-100	0.174
16		EST	95.9	90.5-100	0.17
17		EST	100	95.4-100	0.168
18		ESTs	96.1	92.5-100	0.176
19	NPHP1	Nephronophthisis 1 (juvenile)	95.8	92-100	0.165
20	QDPR	Quinoid dihydropteridine reductase	100	100-100	0.085

Median accuracy, lower and upper inter-quartile ranges, gene names (where known) and descriptions are shown.

3.3.3. Analysis of West et al. breast cancer dataset

The aims here were to (i) identify a gene expression signature that would accurately predict ER status, and (ii) determine whether it was possible to generate a robust model containing genes which would discriminate between patients based upon LN status. As before, an initial analysis was carried out using logistic regression which again led to poor predictive performances with a median accuracy of 78 % (inter-quartile range 67-88 %) for the ER data, and just 56 % (inter-quartile range 44-67 %) for the LN dataset, which is comparable to the predictions one would gain from using a random classifier.

Here, using the stepwise methodology, two gene expression signatures were identified. The first discriminated 100 % of the cases correctly with regards to whether they were positive or negative for ER, and the second predicted whether metastasis of the tumour to the axillary lymph node had occurred, again to an accuracy of 100 %. Again, the accuracies reported are from separate validation data splits, with samples treated as blind data over 50 models with RSCV. The overall screening process assessed over five million individual models. When evaluated with a ROC curve the model had an area under the curve value of 1.0 with sensitivity and specificity values of 100 % and 100 % respectively for both ER and LN status.

Figure 33. Model performance with each input addition over the course of the analysis. Red line represents median model accuracy with lower and upper inter-quartile ranges shown as error bars. Blue line shows the mean squared error for the predictions at each step with error bars indicating 95 % confidence intervals.

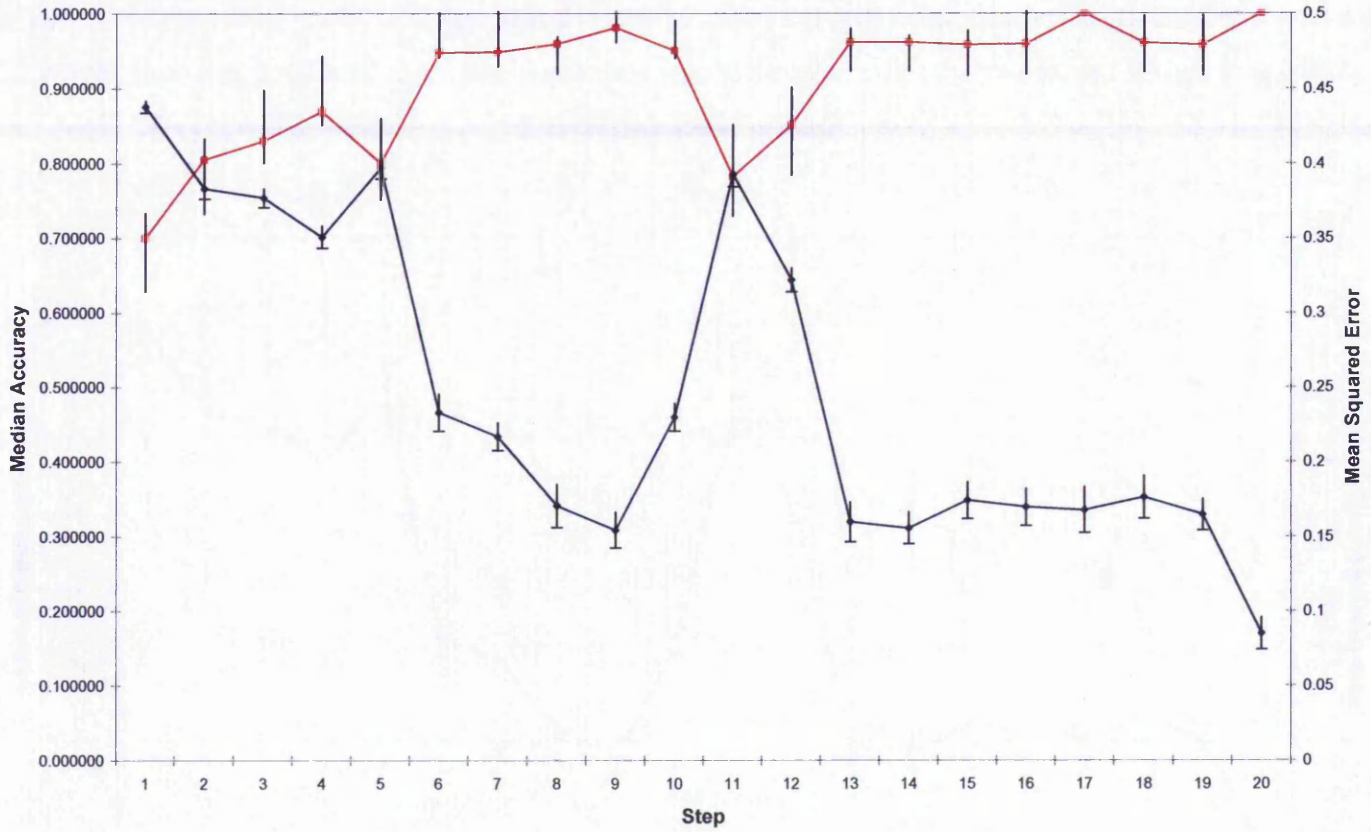
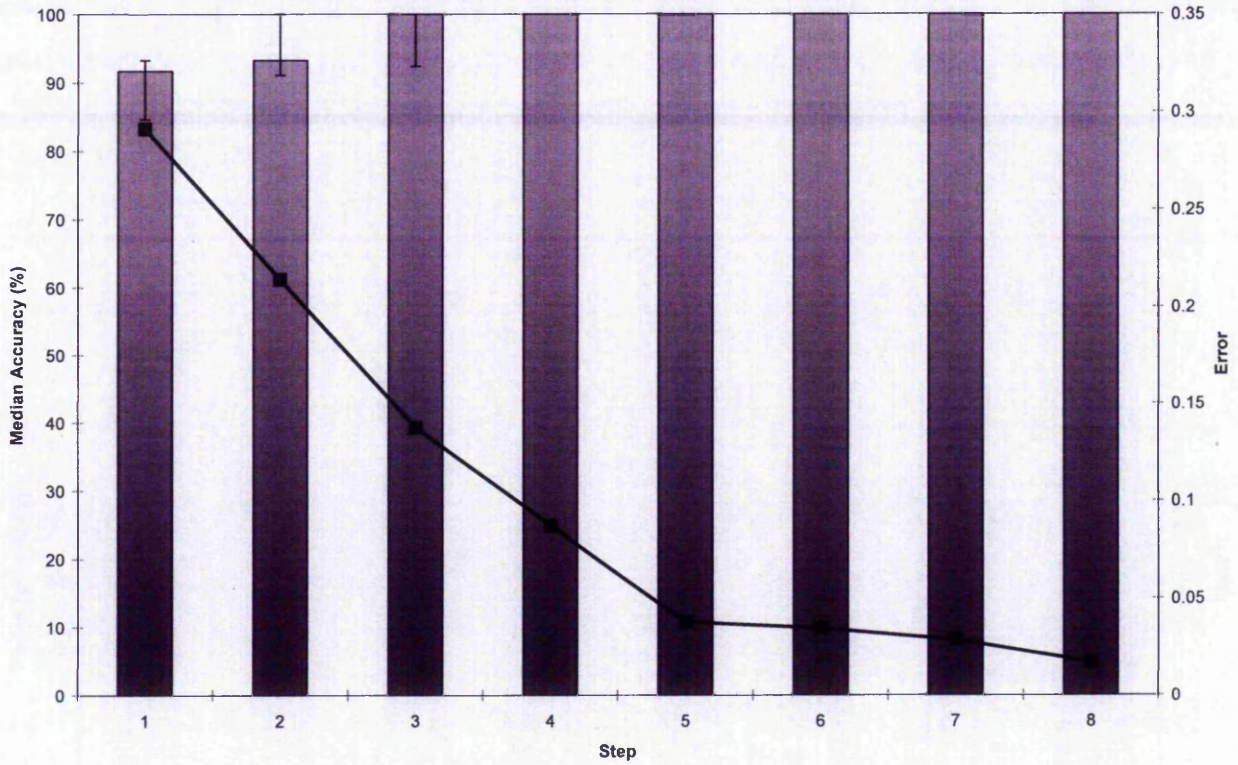


Figure 34a-b shows the performance for the models at each step of the analysis. It is evident that the continual addition of key genes leads to an overall improvement in the error associated with the predictive capabilities of the model for blind data. After steps 8 and 7 for the ER and LN data respectively, no further steps were conducted because no significant improvement in performance could be achieved, therefore these models were considered to contain the genes which most accurately modelled the data. A summary of the performances of the models at each step, together with the identity of these are given in Table 8a-b.

The models developed using the gene subsets identified by the approach described were applied to an additional 88 samples from Huang and colleagues (Huang, et al., 2003). This was generated on a different microarray chip than the original, with the genes overlapping across both chips. There were many differences in this data to the original, for instance it was generated with an Asian based cohort of patients, as opposed to the original data where women of US origin were used. Further, the women taking part in this second set were generally younger with smaller tumours at surgery. This sample set was therefore thought to provide a stern challenge for the stepwise methodology, and were subjected to classification based upon ER and LN status as with the first dataset. 88.6 % of the samples could be classified correctly based on ER status, with a sensitivity and specificity of 90.4 and 80 % respectively. 83 % of samples were correctly classified based upon their LN status, with a sensitivity of 86.7 % and specificity of 80 %. The AUC values by result of ROC curve analysis were 0.874 and 0.812 for the ER and LN gene subset models respectively. It was expected that the predictive accuracies would be reduced when the models were applied to this additional dataset, but the accuracies reported here remain extremely encouraging because of the larger sample size, the differences in sample characteristics and microarray analysis described above. The ability to predict ER status at a higher rate than that of LN status suggests that there is a greater level of variation in the gene expression profiles with respect to LN status compared to that of ER.

Figure 34a-b. Model performance with each input addition over the course of the analysis for (a) ER status and (b) LN status. (■) represent median model accuracy with lower and upper inter-quartile ranges shown as error bars. (▲) shows the mean squared error for the predictions at each step with error bars indicating 95 % confidence intervals.

(a)



(b)

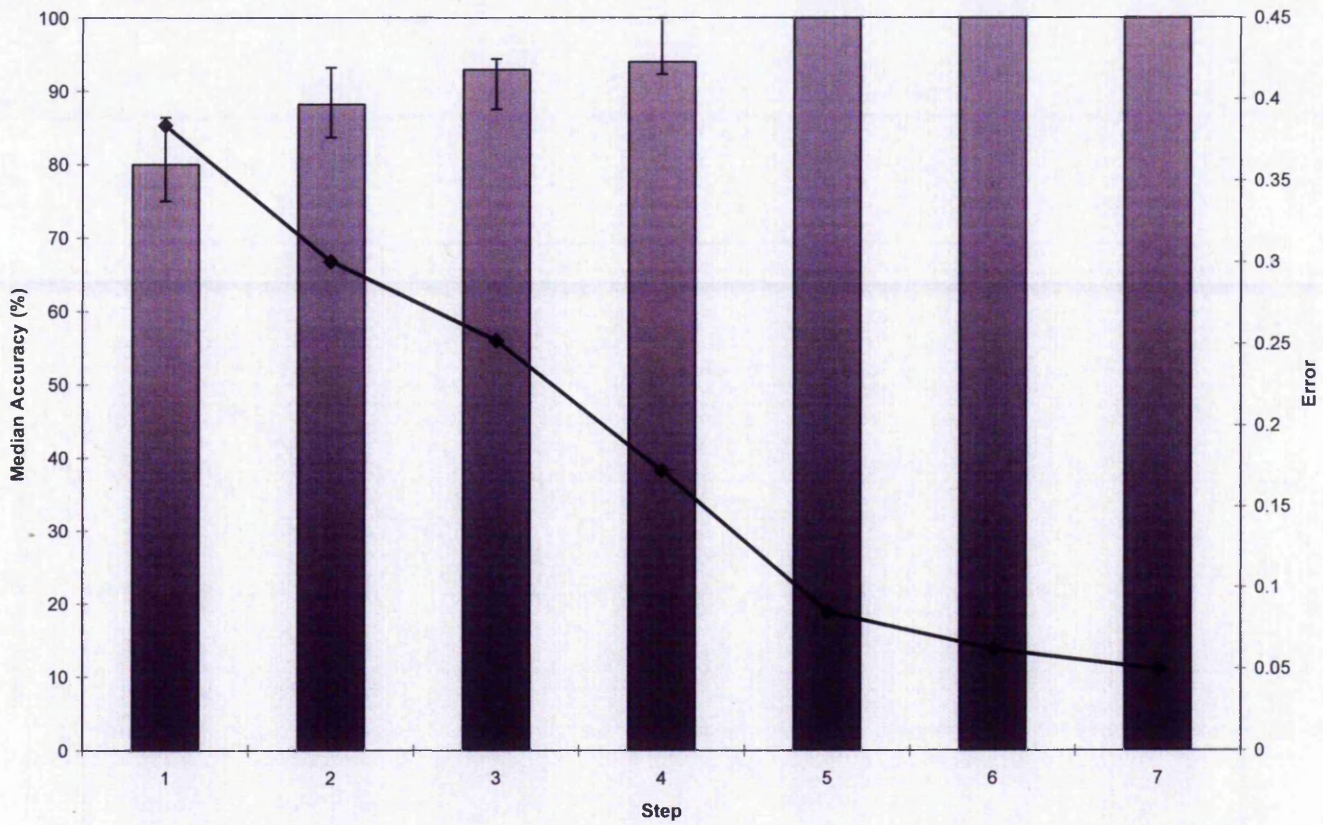

Median Accuracy (%)
Error
Step
100
90
80
70
60
50
40
30
20
10
0
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1
2
3
4
5
6
7

Table 8a-b. Summary genes used in the gene expression signature at each step of model development for (a) ER status and (b) LN status.

(a)

Step	Gene Accession Number	Gene Description	Median Accuracy (%)	Inter Quartile Range (%)	Mean Squared Error
1	X58072-at	Human hGATA3 mRNA	91.7	84.6-93.3	0.291
2	Z29083-at	H.sapiens 5T4 gene for 5T4 Oncofetal antigen	93.3	91.1-100	0.214
3	M81758-at	SkM1 mRNA	100	92.4-100	0.138
4	M60748-at	Human histone H1 (H1F4) gene	100	100-100	0.087
5	M74093-at	Human cyclin mRNA	100	100-100	0.038
6	U22029-f-at	Human cytochrome P450 mRNA	100	100-100	0.034
7	U96131-at	Homo sapiens HPV16 E1	100	100-100	0.028
8	M96982-at	Homo sapiens U2 snRNP auxiliary factor small subunit	100	100-100	0.017

Median accuracy, lower and upper inter-quartile ranges, gene accession numbers, gene descriptions are shown.

(b)

Step	Gene Accession Number	Gene Description	Median Accuracy (%)	Inter Quartile Range (%)	Mean Squared Error
1	AFFX-CreX-3-st	Bacteriophage P1 cre recombinase	80	75-86.4	0.384
2	M83221-at	Homo sapiens I-Rel mRNA	88.2	83.7-93.2	0.301
3	S79862-s-at	PSMD5	92.9	87.5-94.4	0.252
4	U39817-at	Human Bloom syndrome protein (BLM) mRNA	94	92.3-100	0.172
5	U63139-at	Human Rad50 mRNA	100	100-100	0.085
6	M83652-s-at	Homo sapiens complement component properdin mRNA	100	100-100	0.062
7	U30894-at	Human N-sulphoglucosamine sulphohydrolase (SGSH) mRNA	100	100-100	0.05

Median accuracy, lower and upper inter-quartile ranges, gene accession numbers, gene descriptions are shown.

3.4. Summary and discussion

In this chapter novel stepwise ANN approaches to data mining of biomarker ions representative of disease status have been applied to different datasets. This ANN based stepwise approach to data mining offers the potential for identification of a defined subset of biomarkers with prognostic and diagnostic potential. These biomarkers are ordinal to each other within the data space and further markers may be identified by examination of the performance of models for biomarkers at each step of the development process. In order to assess the potential of this methodology in biomarker discovery, three datasets were analysed. These were all from different platforms which generate large amounts of data, namely mass spectrometry and gene expression microarray data.

3.4.1. Mass spectrometry data

Firstly a dataset generated by colleagues here at Nottingham Trent University was analysed. Currently there are no clinically validated markers for metastatic melanoma and researchers based at Nottingham Trent University have been developing MS based proteomic protocols for protein and tryptically digested peptides in order to generate proteomic profiles of human serum samples from patients with melanoma at various stages of disease. Using the stepwise ANN approaches developed in the course of this thesis, 9 protein ions were identified that distinguished Stage IV melanoma patients from healthy controls with an accuracy of 92 %. Using the same approach to analyse the proteomic profiles of digested peptides 6 ions were identified which predicted validation subsets of samples to an accuracy of 100 %. The groups of ions identified here distinguish Stage IV metastatic melanoma from healthy controls with high sensitivity and specificity. This is extremely promising when considering that conventional S-100 ELISA can result in a reported 20% 'false negative' rate in patients with detectable metastases by routine clinical and radiographic studies. Furthermore, it was shown that these ANN based analysis procedures could identify subsets of biomarkers which could accurately classify blind groups of samples to high accuracies for control samples v Stage I (80 %), Stage II (96.5 %), and Stage III (91.7 %), as well as discovering groups of biomarker ions which could identify small groups of biomarker ions with a potential role in disease progression. The ions identified here predicted with extremely high accuracies for Stage I v Stage II (98 %), Stage II v Stage III (100 %), and Stage III v Stage IV (100 %). In addition to this, the same ions were appearing as important predictors across multiple problems in both the

control v diseased, and the inter-stage analyses, further highlighting the confidence that the changes in intensities of these ions is not a random artifact caused by the high dimensionality of the system, but are true biomarkers which are potentially playing a critical role in disease status and are of high diagnostic value.

In previous studies using MALDI or SELDI-based technologies, little attention has been focused upon the mass accuracies and intensities of the spectral peaks and has raised questions as to whether MALDI-MS data of proteins can achieve standards of reproducibility that are expected of clinical tests (Diamandis and van der Merwe, 2005; Mian, et al., 2005). Pre-analytical conditions of blood collections do influence protein/peptide pattern outcomes and it has been argued that many of the reported discriminatory protein profiles have been based largely on experimental artifacts rather than on true biological differences (Sorace and Zhan, 2003). Because of these concerns, the models developed for the Stage IV disease and control samples were validated on a further cohort of samples, which were run on the MALDI mass spectrometer by a different operator on a different date. The models were found to be highly reproducible and robust.

Potential serum protein melanoma biomarker ions by MS using SELDI chips have been reported recently (Mian, et al., 2005), where a mass region around 11,700 Da provided a highly statistically significantly difference in intensity between Stage I and Stage IV melanoma samples. This also addressed some of the inherent problems of resolution, reproducibility and validation of SELDI as a technique for establishing protein fingerprints for use as a clinical aid in diagnosis and prognosis. The MALDI MS method used to generate the data analysed here provides a more rapid data analysis with higher resolution. These data were subsequently subjected to stepwise ANN analysis and nine ions were identified that discriminated between melanoma Stage IV and healthy control sera. This analysis by ANNs of serum proteins resulted in a median accuracy of 92% (inter-quartile range 89.4 – 94.8%) in discriminating between sera from Stage IV melanoma and control patients. The top ion at *m/z* 12000 was able to discriminate between classes with a median predictive accuracy of 64 % (inter-quartile range 58.7-69.2 %). This ion is similar in mass to the biomarker ion of *m/z* 11700 reported using the SELDI technology, also for Stage IV metastatic cancer (Mian, et al., 2005). The difference may be attributed to the fact that this ion was found to be significant when used in discriminating between Stage I melanoma versus Stage IV patients whereas here the ion reported at *m/z* 12000 was identified when classifying between IV melanoma and unaffected healthy control individuals. Further, in

the manuscript by Mian and colleagues (Mian, et al., 2005) predictive performance was based primarily on spectra obtained from Ciphergen SELDI chip platform which are associated with inherent low resolution read-outs using low resolution MS equipment ,whereas here protein biomarker detection was carried out using a higher resolution MALDI-MS analyser, so the *m/z* value of 11700 may have some variation associated with it. Although both studies used ANNs the approaches applied were different; here novel stepwise analysis approaches were used which allow for the identification of individual mass ions with high predictive performance, whereas the SELDI analysis (Mian, et al., 2005) used larger mass ranges to identify regions of the profile which were important in discriminating between groups. Therefore it is important to consider different data mining techniques may elicit different markers with differing importance.

A recent report (Baggerly et al., 2000) has documented the inherent problems that can be associated with the mining of MALDI-TOF MS data for biologic signatures. These will have to be taken into account if a systematically reproducible approach is to be found using biomarker expression patterns as predictive indicators. Several reports on detection of potential melanoma serum biomarkers have used SELDI or MALDI-MS, mostly without bioinformatic analysis (Ferrari, et al., 2000; Mian, et al., 2005; Ragazzi, et al., 2003; Seraglia, et al., 2005). Two studies reported the importance of protein ions between *m/z* 10000 – 30000 but, in particular ions between *m/z* 2500-3000 showed significant variations, increasing from Stage I to Stage II melanoma but absent in controls (Ferrari, et al., 2000; Ragazzi, et al., 2003). In agreement with this, this study showed that ions could be identified in this low molecular weight peptide region which could discriminate between Stage I and Stage II melanoma to high accuracies, and were different from those which classified Stage I and Stage II from control patients. Due to small sample numbers (n = 17) used by Ferrari et al., (2000) no statistical analysis was performed. Recently (Seraglia, et al., 2005) reported the analysis of low molecular weight, *m/z* 800-5000, native peptides in serum and plasma in patients with Stage III melanoma versus healthy volunteers. A small number of ions were found to be present only in the melanoma samples, however, corresponding peptide sequences were not identified and the study concluded no data mining was carried out due to the small sample numbers investigated. Conversely in this thesis enough samples were made available to allow for robust ANN analysis leading to eventual protein identification. There was no overlap with the *m/z* values reported in other studies and those presented here; differences in sample preparations, MS and bioinformatic analysis plays a critical role in data evaluation.

Therefore it was not surprising that the studies detailed failed to identify common biomarkers.

Sequence analysis of the six predictive peptides identified two peptide ions belonging to AGP which when used together in a predictive model could account for 95 % (47/50) of the metastatic melanoma patients. Additionally, another of the peptide ions was identified and confirmed to be associated with complement C3 component. Both proteins have been previously associated with metastatic disease in other types of cancers (Djukanovic, et al., 2000). The tryptic ion at *m/z* 1753 was found to be derived from AGP precursor 1/2. Interestingly, the same AGP tryptic peptide that was identified at *m/z* 1753 was identified by Koomen et al. (Koomen, et al., 2005) in plasma from pancreatic cancer patients using MS in conjunction with MASCOT database searching. Other studies have also shown that increased levels of AGP are found in cancer (Bleasby, et al., 1985; Duche, et al., 2000; Kremmer, et al., 2004). MALDI-TOF MS has shown the up-regulation of AGP in breast cancer patients by analysis of proteins from nipple aspirate fluid (Alexander, et al., 2004). Furthermore, both Duche et al. (Duche, et al., 2000), and Bleasby et al. (Bleasby, et al., 1985) demonstrated that AGP is up-regulated in plasma samples from breast, ovarian and lung cancer patients. The significance of the apparent up-regulation of AGP in melanoma Stage IV serum samples warrants further investigation. AGP, a highly heterogeneous glycoprotein, is an acute-phase protein produced mainly in the liver. However, its physiological significance is not fully understood. The tryptic peptide identified as important by this stepwise approach at *m/z* 1093 was subsequently identified Complement C3 protein precursor 1, an abundant serum protein produced within the kidney. This may be an important mediator of local inflammatory and immunological injury (Djukanovic, et al., 2000).

3.4.2. Gene expression data

To assess whether this approach could also be carried over to the analysis of gene expression data, two publicly available datasets were analysed in an identical stepwise fashion as with the melanoma MALDI-TOF MS dataset. Both of these datasets are associated with breast cancer.

The first was a dataset published by van't Veer and co-workers (van 't Veer, et al., 2002) and the aims here were to identify subsets of genes which could accurately discriminate

between patients who developed distant metastases within five years and those who did not. The initial analysis by van't Veer and colleagues (van 't Veer, et al., 2002) used a form of unsupervised clustering and supervised classification whereby genes were selected by the correlation coefficient of expression with disease outcome. This approach led to the identification of a 70 gene classifier which predicted correctly disease outcome to an accuracy of 83 %. The ANN stepwise approach applied here resulted in the identification of twenty genes which accurately predicted patient prognosis to a median accuracy of 100 % for blind data over a number of RSCV resampling events. Furthermore, an additional cohort of 19 samples were used as a second order validation to the model, resulting in all 19 of these being correctly classified using the gene expression signature identified by the stepwise approach. Some of the genes which constitute this expression signature have previously been associated with cancer outcome. For example the first gene identified by this approach was CA9, and was capable of predicting 70 % of the samples correctly by itself. Carbonic Anhydrase IX (CA IX) has been suggested to be functionally involved in pathogenesis due to its increased expression and abnormal localisation in colorectal tumours (Saarnio, et al., 1998). CA IX has also been suggested for use as a diagnostic biomarker due to its expression being related to cervical cell carcinomas (Liao, et al., 1994). Chia et al. (Chia, et al., 2001) concluded in their study that CA IX expression was associated with a higher relapse rate and worse overall survival in breast cancer patients. Another gene involved in this expression signature is known as TMEFF2, and encodes for a transmembrane protein. Gery et al. (Gery, et al., 2002) showed that TMEFF2 could suppress growth of prostate cancer cells, and later showed (Gery and Koeffler, 2003) that TMEFF2 suppression may contribute to the oncogenic properties of c-Myc, a proto-oncogene involved in cell proliferation, differentiation, apoptosis and also chromosomal abnormalities with roles in tumourigenesis. A third gene identified as important in our genetic profile is NRG2, otherwise known as Neuregulin-2. Meiners and colleagues (Meiners, et al., 1998) reported in their investigations that expression of neuregulin promoted metastases. Similarly, Tsai and co-workers (Tsai, et al., 2003) remarked how breast cancer tumourigenicity and metastasis can be suppressed by blocking neuregulin expression, leading to a reduction in tumour formation and tumour size *in vivo*. Seven of the twenty genes identified as important represent EST's and the associated gene is therefore of unknown function. However, given their predictive capability with regards to survival, further analysis is justified.

The final dataset was published by West et al. (West, et al., 2001) and the ANN stepwise approach was applied to this dataset in order to identify groups of genes would accurately predict the ER status and LN status of the patient. The initial analysis by West and colleagues used regression models in order to calculate classification probabilities for the various outcomes. In their study, when analysing ER status, a 100 gene classifier was identified which predicted 34 of the 38 samples used in the training set accurately and with confidence, and which performed well during cross-validation. Using the same approach, the authors identified a 100 gene classifier which could classify a training set of samples according to lymph node status for the samples used in the training set. However, this approach was less successful in predicting LN status during cross-validation, where all of the LN+ cases had estimated probabilities of approximately 0.5, indicating these predictions contained a great deal of uncertainty, possible due to high levels in variation in the expression profiles of these samples. Here, using the stepwise methodology, two gene expression signatures were identified. The first discriminated 100 % of the cases correctly with regards to whether they were positive or negative for ER, and the second predicted whether the tumour had spread to the axillary lymph node, again to an accuracy of 100 %. The accuracies reported here are from multiple separate validation data splits, with samples treated as blind data over 50 models with RSCV.

If technologies such as these are to be developed to support clinical decision making in the medical arena, and to improve the care and management of patients on an individual basis (a term coined "personalised medicine"), then it is fundamental that they are robust enough to be able to cope with the inherent heterogeneity, complexity and variation which is notorious within diseases such as cancer. To assess whether the gene expression signatures identified here could be applied to a different cohort of patients the models developed using our approach were applied to a follow up dataset generated by the same group as the original (Huang, et al., 2003). This dataset contained many differences to the original dataset used in developing our model. It was generated using an Asian based cohort of patients, whereas the original was derived from women based in the US. In addition to the racial differences, the US women were generally older and had larger tumours at surgery. Moreover this second dataset was obtained using a U95av2 Affymetrix microarray representing 12625 genes, whilst the initial dataset used to develop the models and identify important genes was obtained using an earlier HU6800 array representing 7129 genes, which overlap onto the newer microarray. Because of all of the above discrepancies, and the fact that such sample to sample inconsistencies would likely be commonplace in a

clinical setting, this dataset was thought to provide a robust benchmark for the testing of the methods and subsequent gene signatures described in this chapter. With the models applied to this separate dataset, 88.6 % of the samples were classified correctly based on ER status, whilst 83 % of samples were correctly classified with regards to LN status. Further encouragement was gained from the observation that seven out of the eight genes used in our ER classifier, and three from the seven used in our LN classifier were also present in the patterns of metagenes described in Huang et al. (Huang, et al., 2003), further highlighting their potential biological relevance.

As with the previous analysis, some of the genes identified as important here have also been previously reported to be associated with cancer. For example the first gene identified by the stepwise approach in relation to ER status was GATA3, and was capable of predicting almost 90 % of the samples correctly by itself. GATA3 is a transcription factor which has been shown to be highly expressed in human breast tumours (van 't Veer, et al., 2002; West, et al., 2001), and although the role of this gene remains largely unknown, its expression is strongly and positively correlated to that of the oestrogen receptor alpha gene (Lacroix and Leclercq, 2004), which has also been demonstrated by others analysing similar gene expression data (Sotiriou, et al., 2003). Another gene in the ER expression signature identified here is that which encodes for Oncofetal Antigen, 5T4, a glycoprotein expressed during foetal development and re-expressed as a surface antigen by tumour cells. It has been shown to be detectable in all types of tumours, and undetectable in normal cells (Rohrer, et al., 1999). 5T4 expression is correlated with poor prognosis and metastatic spread, and is potentially linked to tumour cell invasion processes (Carsberg, et al., 1995). Starzynska and colleagues (Starzynska, et al., 1992) reported a highly significant association between 5T4 expression and metastatic spread in colorectal and gastric carcinoma, concluding that "the appearance of 5T4 molecules in cancer cells reflects a change which may contribute to the development of metastatic potential."

Here it has been shown that gene expression profiles can be used as a basis for determining the most significant genes capable of discriminating patients of different status in breast cancer. In agreement with van't Veer et al. (van 't Veer, et al., 2002; West, et al., 2001) we have shown that whilst single genes are capable of discriminating between different disease states, multiple genes in combination enhance the predictive power of these models. In addition to this, the results provide further evidence that ER+ and ER- tumours display gene expression patterns which are significantly different, and can even be

discriminated between without the ER gene itself. This suggests that these phenotypes are not only explained by the ER gene, but a combination of other genes not necessarily primarily involved in the response of ER, but which may be interacting with, and modulating ER expression in some unknown fashion.

Unlike some analysis methods, this method takes each and every variable into account for analysis, and does not use various cut-off values to determine significant differences between them. This overcomes data analysis limitations outlined by Ioannidis (Ioannidis, 2005), and at the same time overcomes many associated problems with high dimensional systems. These models can then form a foundation for future studies using these biomarker subsets to develop simpler prognostic tests, or as candidate therapeutic targets for the development of novel therapies. The proof of concept of this methodology is evident, and was further validated on separate cohorts of patients using the biomarker signatures identified. Given the relevance of the biomarkers identified by this method across different datasets and platforms, and the applicability of these to wider populations, it is felt that this approach is a valid way of identifying subsets of biomarkers associated with disease characteristics. Confidence in the identified markers is increased further still in that many of these have known associations with cancer and some were also included in the sets of metagenes identified by Huang et al. (Huang, et al., 2003).

CHAPTER 4. EXTENDING AND VALIDATING THE STEPWISE APPROACH

4.1. Identification of multiple biomarker subsets

4.1.1. Introduction

The stepwise methodology described in the previous chapter facilitates the identification of subsets of biomarkers which can accurately model and predict sample class for a given complex dataset. In order to facilitate a more rapid biomarker subset analysis, the stepwise approach adds only the best performing biomarker at each step of the analysis. Although this appears to be an extremely robust method of biomarker identification, the question remains as to whether there are additional subsets of biomarkers existing within the dataset, which are also capable of predicting class to high accuracies. If this is true, then this would lead to a further understanding of the system being modelled. Further, if multiple biomarkers were to appear in more than one marker subset, then this would further validate their identification, and enhance their potential use in disease diagnosis and prognosis, warranting further investigation.

To achieve these aims, the same West dataset was used as previous (West, et al., 2001). As can be seen from Table 9a-b, in addition to the number one ranked biomarker at step one (which was subsequently used as the basis for the gene biomarker signature described in the previous chapter), there are several other potential candidate biomarkers which by themselves are able to classify a significant proportion of the sample population into their respective classes. Therefore an individual stepwise analysis was conducted on each of the remaining top ten genes identified in step one of the analysis, for both ER and LN status to determine if multiple biomarker subsets existed within the dataset.

This chapter concludes by using different methods in order to validate the stepwise approaches developed in this research. This is done in order to provide confidence that the biomarker subsets identified using these methodologies are not random artifacts due to the complexity of the dataset. This ensures that the biomarkers are of biological relevance in the disease of interest.

4.1.2. Results

Figure 35a-b shows the network performance at each step of analysis for all of these genes for (a) ER and (b) LN status. It is evident that all of these subsets have the ability to predict for blind subsets of samples to extremely high accuracies of greater than 90 %, with no significant differences between individual models. This suggests that there may be multiple genes acting in response to disease status, subsequently altering various pathways and altering the expression levels of many other genes. It is worthwhile to note that some of these genes were identified in many of the models (Table 10), for example an EST appeared in seven out of ten models, further highlighting its potential importance in LN status. This shows that there is not necessarily just one set of biomarkers which are correlates of a particular disease status of interest, but there may be many, and when one particular subset of biomarkers are affected in such a way that is indicative of disease status, then this may consequently have a cascade effect on many other biomarkers, altering their expression in a similar fashion.

Table 9a-b. Summary of step 1 analysis for (a) ER and (b) LN status. Table shows the gene identification and respective predictive performances of the top 10 ranked genes identified at step 1 of the analysis.

(a)

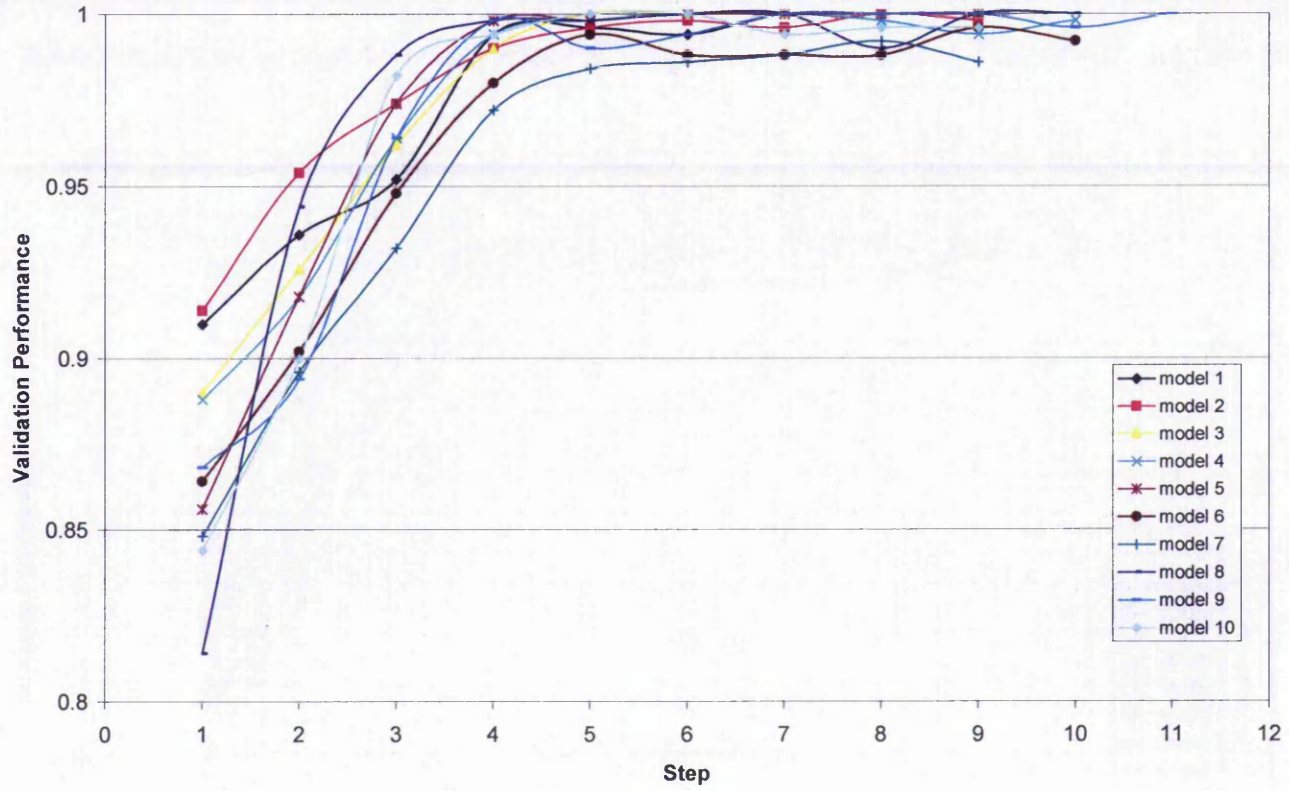
Rank	Gene ID	Blind Performance (%)
1	GATA3	89.8
2	ESR1	87.6
3	SLC39A6	85.5
4	EST	85.3
5	HSD17B4	83.3
6	EST	84.2
7	AR	83.0
8	LAD1	84.0
9	SCNN1A	84.2
10	MAPT	80.2

(b)

Rank	Gene ID	Blind Performance (%)
1	EST	80.4
2	GYPA/B	70.9
3	BLM	71.2
4	ACVR1B	70.4
5	EST	64.3
6	WNT5A	66.7
7	RELB	61.3
8	GK	64.1
9	PDE4B	64.3
10	TLE1	64.7

Figure 35a-b. Summary of stepwise analysis for top ten genes identified at step 1 for (a) ER and (b) LN status.

(a)



(b)

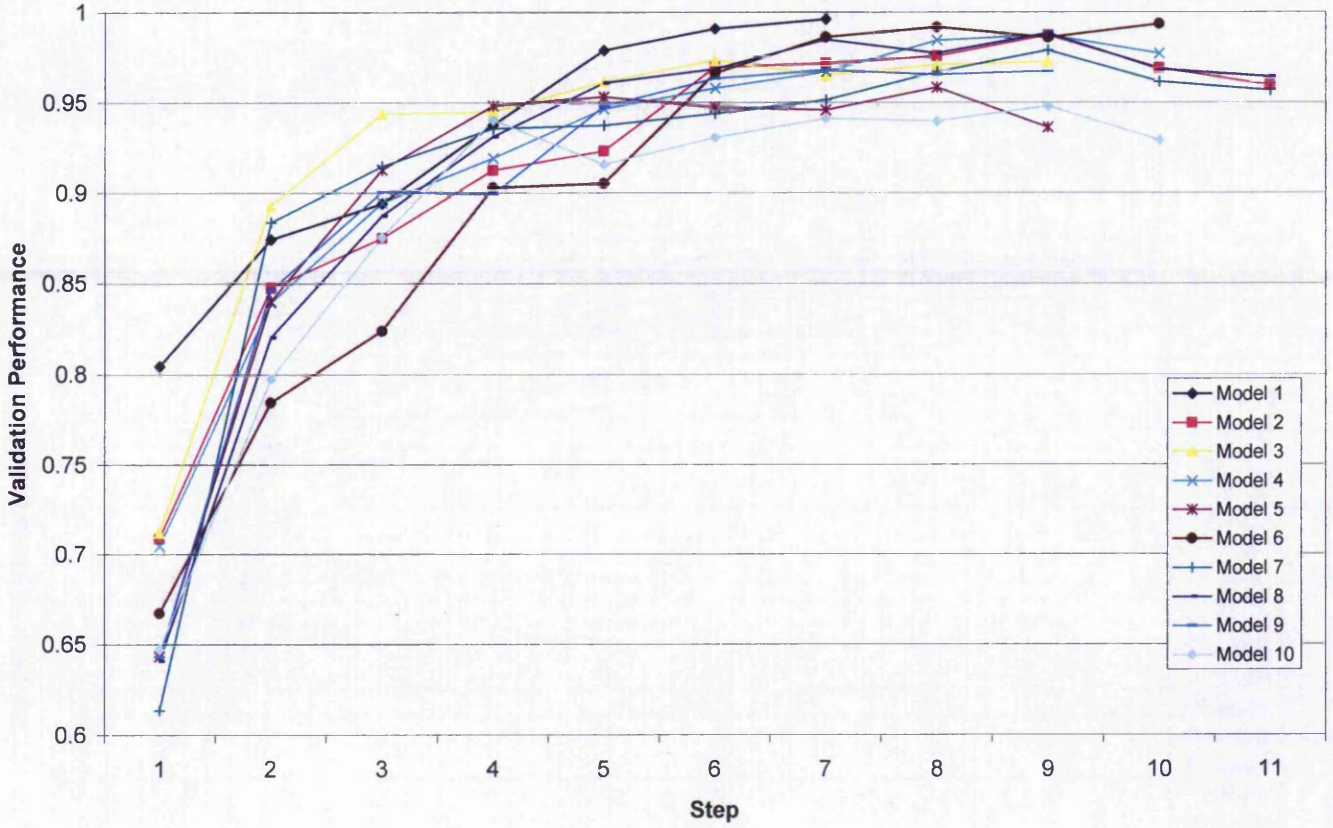

Validation Performance
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
1
2
3
4
5
6
7
8
9
10
11
Step
Model 1
Model 2
Model 3
Model 4
Model 5
Model 6
Model 7
Model 8
Model 9
Model 10

Table 10. Summary of genes identified in multiple stepwise modelling which occur in more than one model in (a) ER and (b) LN status.

(a)

Gene ID	Actual Gene Name	Number of Occurrences (In 10 Models)
CYP2B6	Cytochrome p450 polypeptide 6	3
CTSC	Cathepsin c	3
GATA3	Gata binding protein 3	2
EST	EST	2
CYP2A7	Cytochrome p450 polypeptide 7	2
LRRC17	Leucine rich repeat	2
NFKBIE	Nuclear factor of kappa	2
COX6C	Cytochrome c oxidase	2
HLF	Hepatic leukemia factor	2
IGLC	Immunoglobulin lambda	2
ZBTB16	Zinc finger	2
RTN1	Reticulon 1	2

(b)

Gene ID	Actual Gene Name	Number of Occurrences (In 10 Models)
EST	EST	7
BLM	Bloom syndrome	6
ACVR1B	Activin a receptor	4
GYPA/GYPB	Glycophorin a/b	3
AXIN1	Axin 1	3
RELB	V-rel reticuloendotheliosis viral oncogene homolog b	2
PSMD5	Proteasome (prosome, macropain)	2
SGSH	N-sulfoglucosamine sulfohydrolase (sulfamidase)	2
CTSH	Cathepsin h	2
NUP88	Nucleoporin 88 kDa.	2
ENG	Endoglin	2
SYBL1	Synaptobrevin-like 1	2

4.2. Stepwise analysis validation

To provide further evidence and confidence that the biomarker subsets identified in all of the above analyses by the stepwise approach were not random as a consequence of the high dimensionality of the datasets, two validation exercises were conducted. Firstly, ten inputs were randomly selected from the datasets and trained over 50 RSCV events in an ANN model identically as for the stepwise method. This process was repeated 1,000 times, and the summary results are presented in Table 11.

It is clear from Table 11 that the variation amongst models generated with these random input subsets is small, suggesting that a randomly generated model is able to predict sample class to accuracies in the region of 64 % for blind data. These models will very rarely predict significantly higher than this value, which is highlighted in Figure 36 which details the distribution of the model performance across the various models. The data follows a normal distribution, and therefore it is unlikely that a random model would generate a subset of inputs capable of very high classification accuracies, indicating that the stepwise ANN approach to modelling described here is selecting inputs which are discriminating between the groups of interest in a biologically relevant manner.

Table 11. Summary results of random input selection.

Summary Statistic	Validation data accuracy	Validation data error
Average	64 %	0.495
Standard Deviation	0.024	0.014
Standard Error	2.45×10^{-5}	1.41×10^{-5}
95 % confidence interval	4.89×10^{-5}	2.82×10^{-5}
Median	64 %	0.495
Inter Quartile Range	62-66 %	0.485-0.504

Figure 36. Normal distribution of randomly generated models.

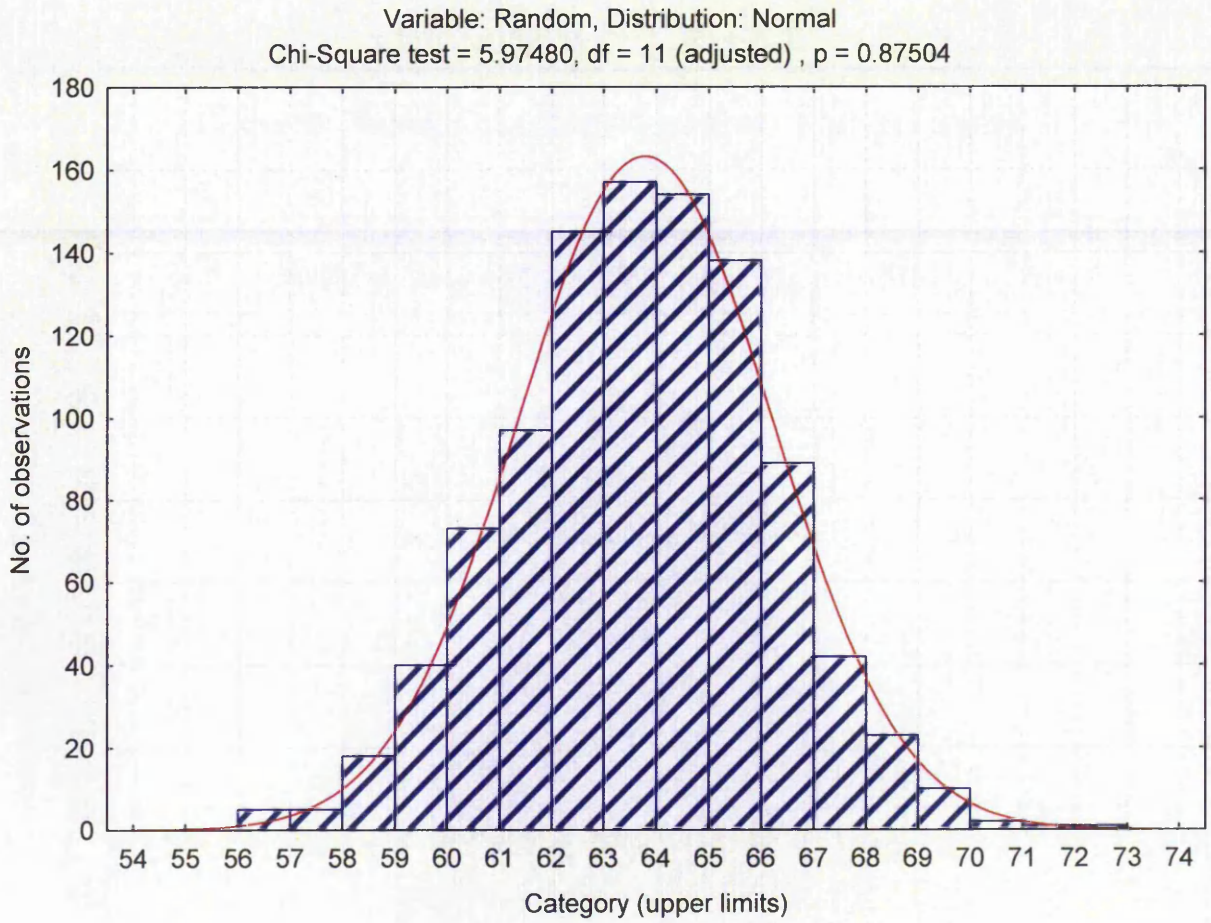


Figure 37a-c highlights the significance between the performance of the randomly generated models and those developed with the stepwise approach for the van't Veer and West gene expression datasets (van 't Veer, et al., 2002; West, et al., 2001).

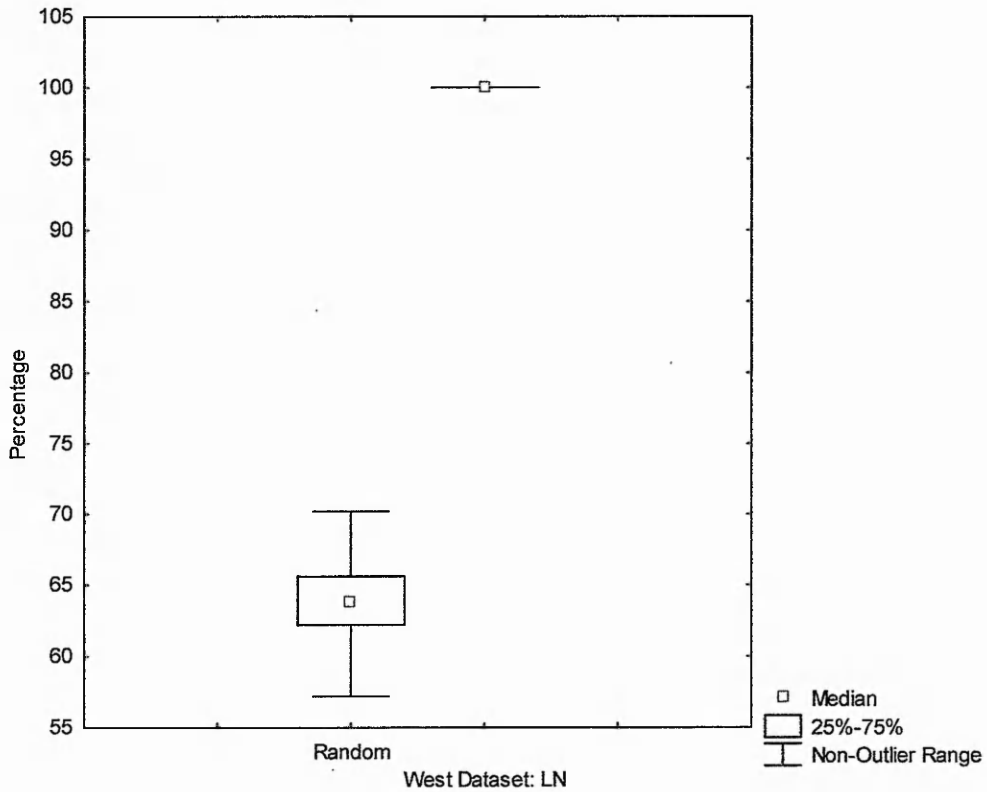
These results show that a random classifier would indeed, as expected, lead to classification accuracies close to random, and therefore it can be said that the stepwise approach truly identifies subsets of inputs which predict well on unseen data.

Now it was necessary to investigate whether this stepwise approach would identify the same inputs if the analysis was run several different occasions, starting over each time with the same dataset. If not, it would be necessary to determine how many RSCV resampling events would be necessary to generate models which are both generalised to new unseen data, and that are reproducible so that confidence can be gained in the inputs identified as important. To achieve this, the stepwise analysis was run and trained on the van't Veer dataset with samples randomly split by RSCV into training, test, and validation subsets 10, 20, 50 and 100 times and subsequently trained. This whole process was repeated five times to calculate how consistent the ranking of the individual inputs was with regards to model performance. This consistency was calculated for the top fifty most important inputs, and was the ratio of *actual* ranking (based upon the average error of the model), to the *average* ranking over the multiple runs. These are summarised in Table 12.

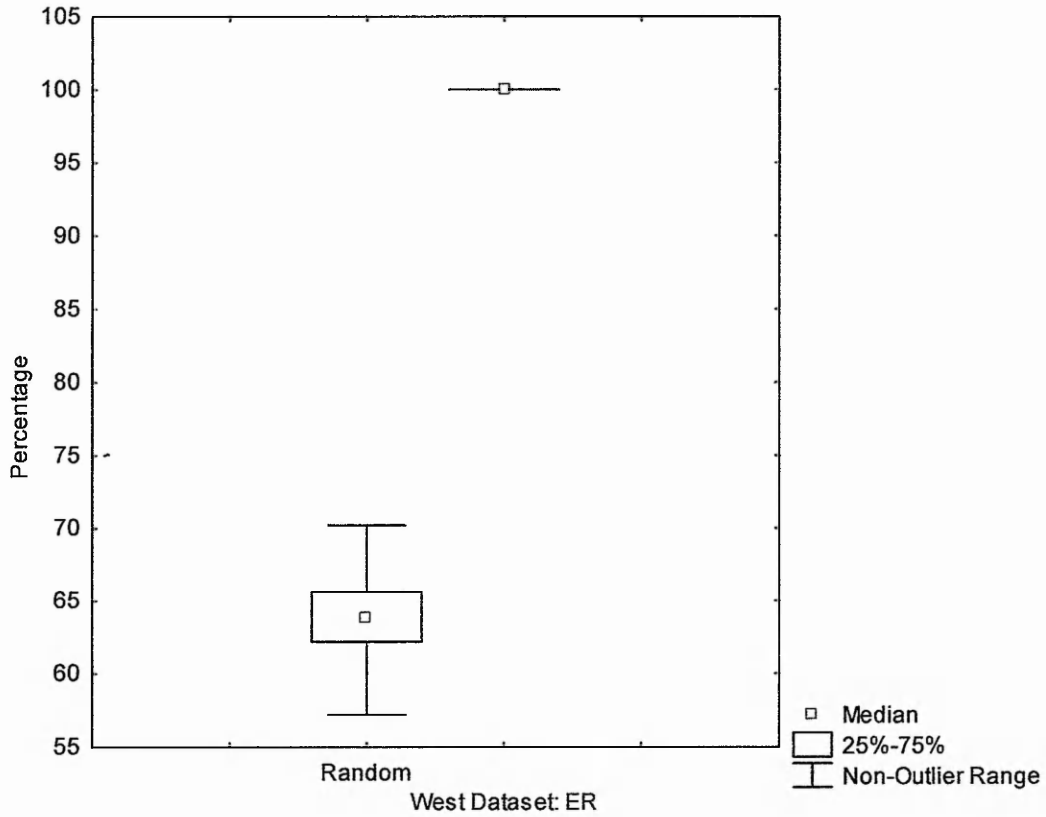
There was a significant increase in consistency amongst the performance of inputs when increasing from 10 to 20 ($p=0.000$), and 20 to 50 RSCV datasplits ($p=0.000$), but not from 50 to 100 ($p=0.2213$). Interestingly, for all analyses, the same two inputs were ranked as first and second every time, with the majority of the variation in rankings appearing towards the bottom of the top 50 list, which accounts for the 14 and 12 % variability in the 50 and 100 RSCV event models respectively. This showed step 1 to be extremely consistent in identifying the same inputs as important across multiple analyses.

Figure 37 a-c. Comparison of the performance of random models to those generated with the stepwise approach.

(a)



(b)


105
100
95
90
85
80
75
70
65
60
55
Percentage
Random
West Dataset: ER
Median
25%-75%
Non-Outlier Range

(c)

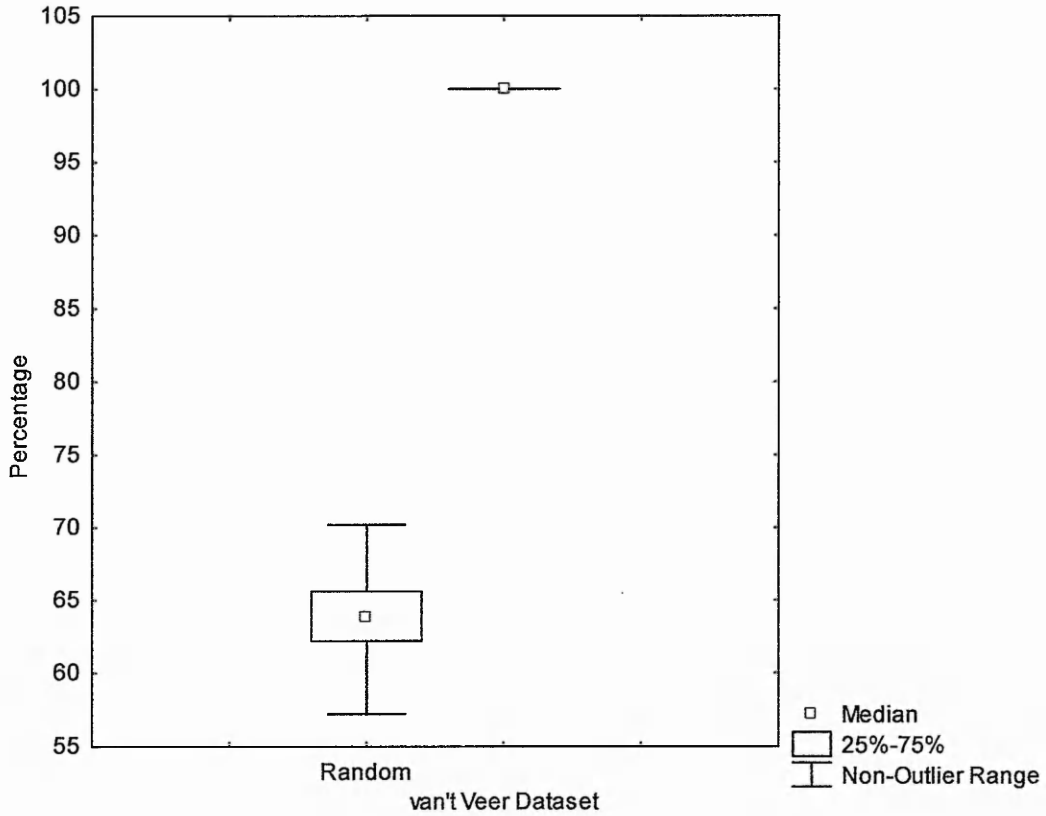

105
100
95
90
85
80
75
70
65
60
55
Percentage
Random
van't Veer Dataset
Median
25%-75%
Non-Outlier Range

Table 12. Summary of the consistency of inputs identified as important using varying random sample cross validation data splits in step 1 of the analysis.

Number of RSCV datasplits	Mean Group Consistency	95 % confidence intervals
10	0.547	0.009
20	0.708	0.009
50	0.859	0.010
100	0.880	0.013

The same procedure was then carried out for step 2, with the input identified as the most important across all the models in step 1 used to form the basis of this second step. Table 13 shows the average consistency ratios for step 2. It is clear from this that consistency across multiple repeats of the analysis showed a dramatic decline, with only the 100 RSCV model retaining its consistency in input identification, and the improvement in consistent input performance was statistically significant ($p=0.000$) at each increment. Again, the models using 50 and 100 RSCV resampling events both identified the same input ranked as number one and number two. It therefore appears that a *minimum* of 50 RSCV datasplits is required to ensure that the same inputs are consistently identified as important multiple times in 80-90 % of analyses.

Table 13. Summary of the consistency of inputs identified as important using varying random sample cross validation data splits in step 2 of the analysis.

Number of RSCV datasplits	Mean Group Consistency	95 % confidence intervals
10	0.140	0.004
20	0.487	0.011
50	0.657	0.009
100	0.811	0.009

4.3. Summary and discussion

It is important to recognise that in biological problems it is unlikely that there lies only one solution to a particular problem, and there are potentially many biomarker subsets which have the ability to predict class to high accuracies. This has been shown in this chapter where multiple genes were taken as a starting point for subsequent stepwise analysis. All of these gene subsets were capable of predicting disease status to high accuracies. Interestingly, many of the genes identified appeared in several models, further validating and highlighting the potential importance of their role in disease. Again, various studies have shown some of these genes to be associated with cancer. For example cytochrome p450 (CYP2B6) was shown to be an ER responsive gene in human breast cancer by Bieche and colleagues. (Bieche, et al., 2004). The BLM (Bloom syndrome) gene was found to occur in six of the ten gene signatures identified as important in the classification of LN status. Mutations in the BLM gene result in a genomic instability that causes Bloom's syndrome, a rare autosomal recessive genetic disorder characterised by sun sensitivity, stunted growth infertility and immunodeficiency (Kaneko, et al., 1996). Bloom's syndrome patients have been documented to be highly predisposed to cancers (Calin, et al., 1998; German, 1997; Kaneko, et al., 1996; Watt and Hickson, 1996), and its clear high importance in the gene subsets identified here suggest the expression of this gene is a high correlate of disease. RELB was another gene associated with LN status across multiple gene subsets. This has previously been shown to be implicated in carcinogenesis (Demicco, et al., 2005). Also extremely interesting were cathepsins C and H, identified to be important across multiple models discriminating in both ER and LN status respectively. Cathepsins have been observed in several types of human cancers, including breast, and have been particularly associated with aggressive cancer cells. Cancers appear to recruit various cathepsins to progress malignant disease and as such these have been shown to be useful in the prognosis of disease and cancer progression (Nomura and Katunuma, 2005).

To further show that the analysis approach proposed here is not identifying random input subsets as a consequence of the high dimensionality and complexity of the datasets, subsets of inputs were selected at random one thousand times and trained using RSCV. This showed that a random subset of input variables would predict class to approximately 60 %. Furthermore, to measure consistency, it was shown that if a minimum of 50 RSCV resampling events is conducted during model development, the same inputs are identified

as the most important variables in class prediction over multiple repeats of the experiment. In summary, this validation exercise has shown that the inputs being identified are not random or artifacts and confidence in their importance in differentiating between the classes of interest can be high.

One drawback of traditional linear based stepwise models is that they often cannot generalise well to problems and therefore may only be applicable to the dataset they are originally applied to. Simulation experiments have shown that stepwise logistic regression has limited power in selecting important variables in small data sets, and therefore risks overfitting (Steyerberg, et al., 1999). Additionally the automatic selection procedure is non- subjective and ignores logical constraints. It was shown here that the neural network stepwise approach described does not suffer from these limitations because the models have been shown to be applicable to separate datasets used for validation. As such the stepwise approach has generated models that are capable of generalisation to new data so is less likely to suffer from overfitting, which has not been observed in the analyses conducted here.

To conclude, the last two chapters have shown that using these novel ANN methodologies, a powerful tool has been developed for the identification of biomarker subsets that predict disease status in a variety of analyses. The potential of this approach is apparent by the high predictive accuracies as a result of using the biomarker subsets identified here. These biomarker subsets were then shown to be capable of high classification accuracies when used to predict for additional validation datasets, and were even capable of being applied to predict the ER and LN status of a dataset very different in origin from the one used in the identification of the important gene subsets. This in combination with the various validation exercises that have been conducted suggests that these biomarkers have biological relevance and their selection is not arbitrary or an artifact of the high dimensionality of the system as they were shown to be robust to cope with sampling variability and reproducible across different sample studies.

CHAPTER 5. MODEL INTERROGATION

5.1. Introduction

Once a parsimonious set of biomarkers have been identified and models have been developed from these relating to clinical groups of interest, it is possible to interrogate these further. Here, in addition to having a prognostic or diagnostic model capable of the accurate prediction and classification of samples on the basis of their disease class, one can begin to elucidate biological meaning from the data, providing further insight into the disease of interest. This chapter discusses methods aimed at interrogating these key sets of biomarkers. This incorporates methods which show how the class of an individual sample case can be related to the expression or intensity of an individual biomarker, accomplished by monitoring the response of these markers in relation to the network classification. It is shown how these responses may be positive, negative, discriminatory and co-factorial. Secondly it is shown how a population of case samples can be investigated to identify how characteristic a sample is of the class they were assigned to. This is achieved by monitoring and interpreting the network predictions in a continuous manner, enabling sub-populations of cases within a disease group to be identified. Additionally, new individuals may be positioned within a defined population allowing for risk analysis with reference to a specific disease or condition. Finally a proposed method with a view to understanding relationships and interactions between biomarkers to derive gene regulatory networks is introduced. This is of particular interest from a clinical perspective because the ability to define new pathways would possibly lead to the identification of novel therapeutic targets. This chapter illustrates these methods using the van't Veer and West gene expression breast cancer datasets (described in previous chapters).

5.2. Determining the response effect of biomarkers

It is possible to monitor the ANN output in order to determine the effect the individual biomarkers have on the classes of interest. This enables one to understand how these biomarkers are governing sample classification and begin to derive rules for molecular classification by assessing the strength of response and whether it is discriminatory or cofactorial. This is achieved by using the trained ANN model, and adjusting an input variable of interest in order to monitor what affect this adjustment has on the output variable. The output with respect to the changing input value can then be plotted to produce a response graph. This can be extremely useful in proteomic and genomic studies, where it becomes possible for one to gain an insight into how disease picture changes with respect to the expression or relative abundance of genes and proteins of interest.

5.2.1. Methods

The biomarker subsets identified using the stepwise approach to biomarker identification for the van't Veer (refer to chapter 3.3.2, and Table 7) and West (refer to chapter 3.3.3, and Table 8a-b) gene expression datasets were used to generate response graphs.

As previously mentioned, each model was trained and tested over 50 RSCV resampling events. To investigate the response of these biomarkers, the value of each input variable was sequentially adjusted whilst all other input variables remained fixed over these 50 RSCV events. This adjustment was a value slowly incremented through the minimum and maximum range of that variable across all sample cases. The output change with respect to this changing input value was then recorded for all of the 50 sub-models. These were then averaged and the subsequent response graphs were plotted with the changing input variable plotted on the x-axis of the graph.

5.2.2. Results

5.2.2.1. Response graph analysis of the gene expression signature identified in the van't Veer data analysis

The response graphs of the twenty genes identified with the stepwise approach when applied to the van't Veer dataset were generated. Results showed that seventeen of the

twenty response curves had strong discriminatory responses between output classes. The direction of this response enables one to determine whether these genes were being induced or repressed with regards to the development of distant metastases, and therefore calculate how the probability of metastases changes with increased expression of a particular gene. Table 14 illustrates the level of expression of the individual genes or EST's that correlate with the probability of metastases.

Many genes, for example LCHN show an increased likelihood of metastases, whilst others, for example HSU93243 identified at step 14, showed weak responses that associated with distant tumour metastasis. Eight of the genes showed that with increased expression, the probability of developing distant metastases increased, so these may have a possible role in disease pathogenesis. Nine other genes showed that with increased expression, the likelihood of metastases development decreased, suggesting that these genes may play a role in tumour suppression. The final three genes, showed a weak response, *i.e.* the probability of metastases developing was approximately 0.5 irrespective of expression. This suggests that these gene may play a secondary role, interacting with other gene responses in an additive fashion modulating the overall expression pattern. Example curves for genes CA9, TMEFF2 and NRG2 are shown in Figure 38a-c.

Since some of the genes identified with the stepwise ANN analysis had previously been associated with cancer, they were examined further to establish if this correlated with the response curves. This would provide valuable insight into the molecular nature of the development of distant metastases within a five year time frame. Firstly, the gene identified at step one of the analysis was CA9, which as described in the previous chapter, has been shown to be involved in tumour pathogenesis and labelled as a potential candidate for use as a diagnostic marker. The response curve from our study shows that as CA9 expression increases, the probability of developing distant metastases also increases (Figure 38a), which is comparable to what others have found in their laboratory studies (Saarnio, et al., 1998) (Liao, et al., 1994) (Chia, et al., 2001). Another gene identified as an important predictor in this gene expression signature is known as TMEFF2, encoding for a transmembrane protein. It is evident from the response curve in Figure 38b that increased expression of this gene leads to a higher probability of a patient remaining metastases free in the five year time frame under investigation here. This suggests that TMEFF2 expression may inhibit tumour growth, which was also shown in other published studies (Gery and Koeffler, 2003; Gery, et al., 2002). NRG, or Neuregulin-2, has also been

shown to be associated with cancer, and a promoter of metastases (Meiners, et al., 1998; Tsai, et al., 2003). Figure 38c shows that with increased expression of this gene, the probability of metastases development also increased, which again appears to be in agreement to what was previously shown by the other groups.

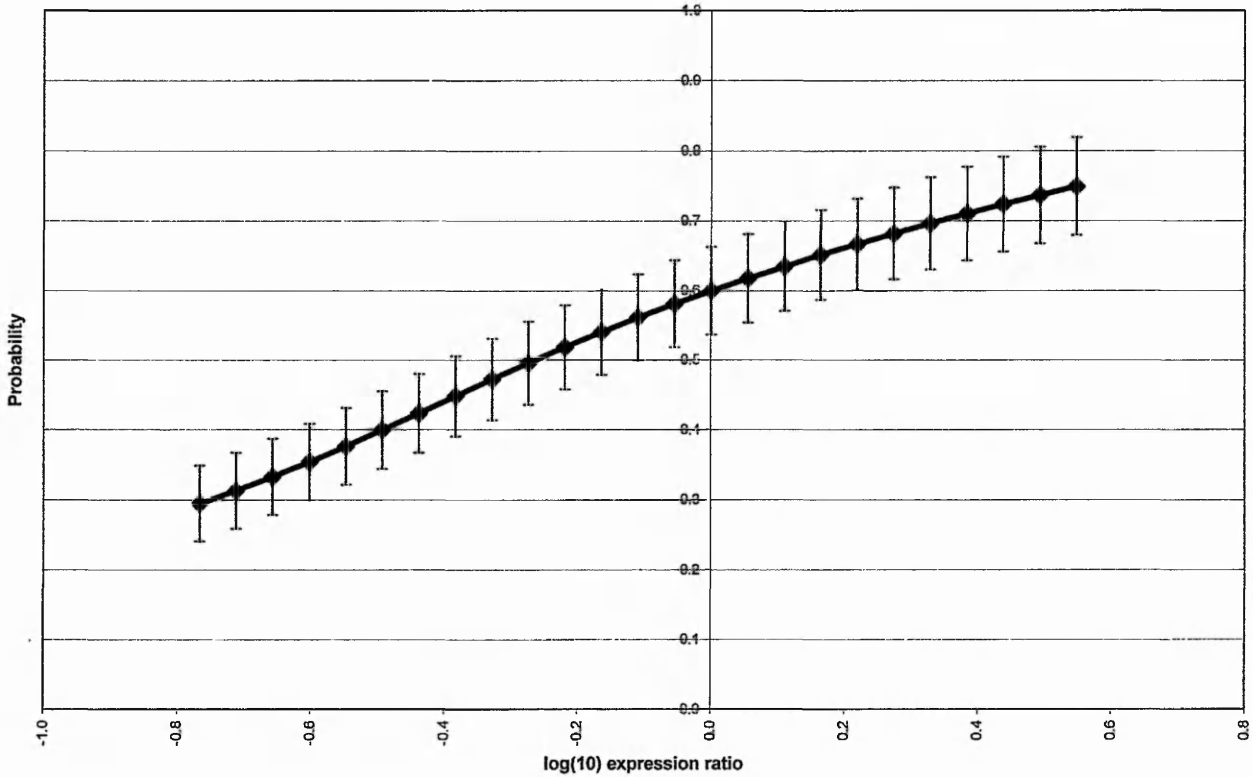
Table 14. Response curve summary for genes identified as the most important discriminators from the van't Veer dataset.

Gene Description	Response
Carbonic anhydrase IX	*
EST's	†
ESTs, Weakly similar to RL17_HUMAN 60S RIBOSOMAL PROTEIN L17 [H.sapiens]	†
ESTs, Weakly similar to the KIAA0191 gene is expressed ubiquitously [H.sapiens]	*
LCHN protein	*
Transmembrane protein with EGF-like and two follistatin-like domains 2	†
Highly expressed in cancer, rich in leucine heptad repeats	*
Homo sapiens HSPC337 mRNA, partial cds	†
EST's	‡
Homo sapiens cDNA: FLJ22044 fis, clone HEP09141	*
UDP-glucose:glycoprotein glucosyltransferase 1	*
putative 47 kDa. protein	†
Hypothetical protein dJ462O23.2	†
Ubc6p homolog	‡
Neuregulin 2	*
EST's	‡
EST's	*
EST's	†
Nephronophthisis 1 (juvenile)	†

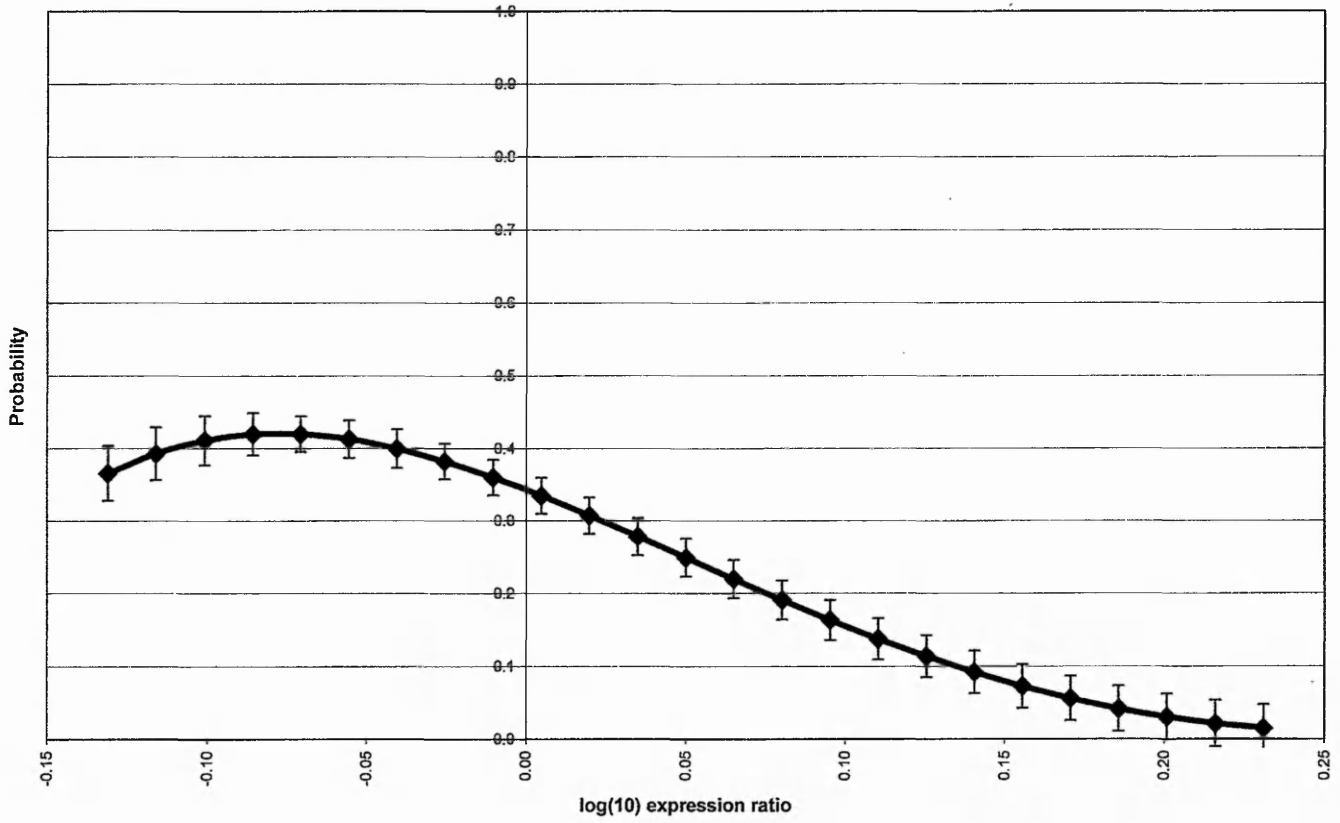
The description of the response is summarised with regards to the probability of whether a patient will or will not develop distant metastases within a five year period.
* Increased gene expression leads to increased probability of developing distant metastases
† Probability of no metastases increases with expression of this gene, possible tumour suppressor
‡ Weak response, possibly interacting to modify response of other genes

Figure 38a-c. Response curves for (a) CA9, (b) TMEFF2 and (c) NRG2. Figures show the log (10) gene expression ratio of each gene plotted against the probability of developing distant metastases within a five year time period. Error bars indicate 95 % confidence intervals.

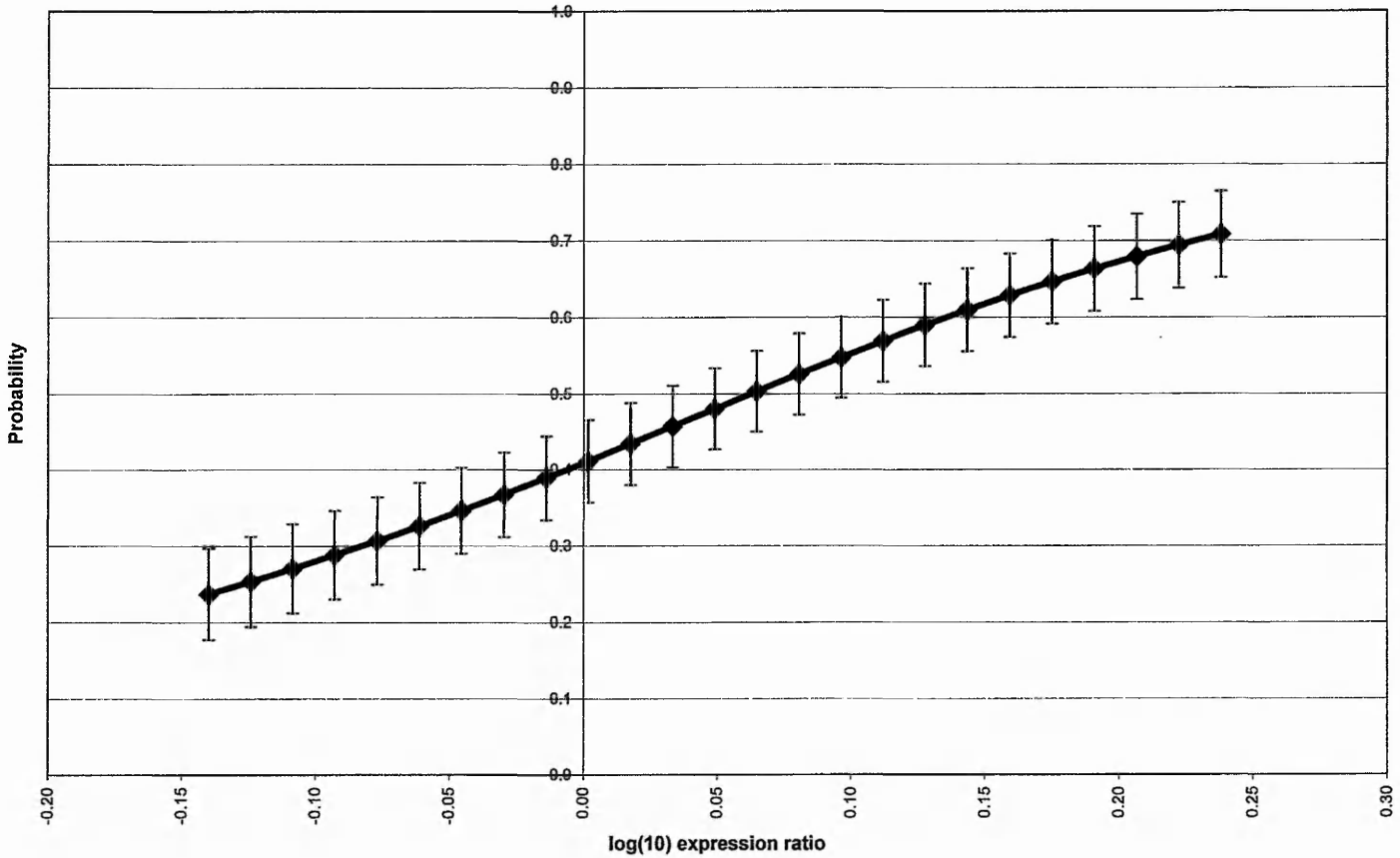
(a)



(b)


Probability
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.15
-0.10
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
log(10) expression ratio

(c)


Probability
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.20
-0.15
-0.10
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
log(10) expression ratio

5.2.2.2. Response graph analysis of the gene expression signature identified from the West data analysis

The response curves of the genes identified in the initial West analysis were then evaluated as with the previous dataset. As before, the majority of these genes had strong discriminatory responses indicating whether they appeared to be induced or repressed with regards to ER or LN status. Table 15a-b describes how the level of expression of the individual genes correlates with these different groups of patients for ER and LN status respectively. Some genes, for example RELB showed that with increased expression the probability of the patient being LN+ increases, suggesting that this gene may be involved in tumour metastasis and pathogenesis. The same pattern was seen for GATA3 mRNA as with an increased expression of this gene, the profile of the patient was more representative of an ER positive case. Other genes, for example BLM showed that with increased expression, the likelihood of the tumour having spread to the lymph node decreased, appearing to have a possible role in tumour suppression. Genes such as the Human cytochrome P450 gene showed weak responses indicating more intermediary roles in classification. Examples response curves for several genes are shown in Figure 39a-d.

Some of these genes had also been previously reported to be associated with cancer. Therefore it was worthwhile to review the response curves for these genes to determine whether the direction of this response appeared to concur with what was reported in the literature. As mentioned, the first gene identified by our model in the classification of ER status was GATA3, which has been shown on numerous occasions to be highly expressed in breast cancer tumours (van 't Veer, et al., 2002; West, et al., 2001). The response curve from our study for this gene (Figure 39b) agrees with these previous findings in that as GATA3 expression increases, so does the likelihood of the sample originating from an ER+ patient, further validating the correlation between these two genes. Another gene, 5T4, which encodes for Oncofetal Antigen has shown to be present in tumours but not in normal cells and as such has been associated with poor prognosis (Rohrer, et al., 1999). This was examined for its response with regards to ER status. Results showed that with increased expression of this gene, the probability of a patient being ER+ decreased. Therefore a sample that is over-expressing this gene was more characteristic of an ER- phenotype, and poorer prognosis. The ANN response for the 5T4 gene (Figure 39c) therefore correlates well with the literature since ER- breast cancer patients generally have a poorer prognosis than their ER+ counterparts.

Table 15a-b. Response curve summary for genes identified as the most important discriminators for (a) ER and (b) LN status from the West dataset.

(a)

Gene Description	Response
Human hGATA3 mRNA	*
H.sapiens 5T4 gene for 5T4 Oncofetal antigen	†
SkM1 mRNA	†
Human histone H1 (H1F4) gene	‡
Human cyclin mRNA	†
Human cytochrome P450 mRNA	‡
Homo sapiens HPV16 E1	†
Homo sapiens U2 snRNP auxiliary factor small subunit	‡

The description of the response is summarised with regards to the probability of whether a patient is or is not ER positive.
* Increased gene expression leads to increased probability of ER+ status.
† Increased gene expression leads to increased probability of ER- status.
‡ Weak response, possibly interacting to modify response of other genes.

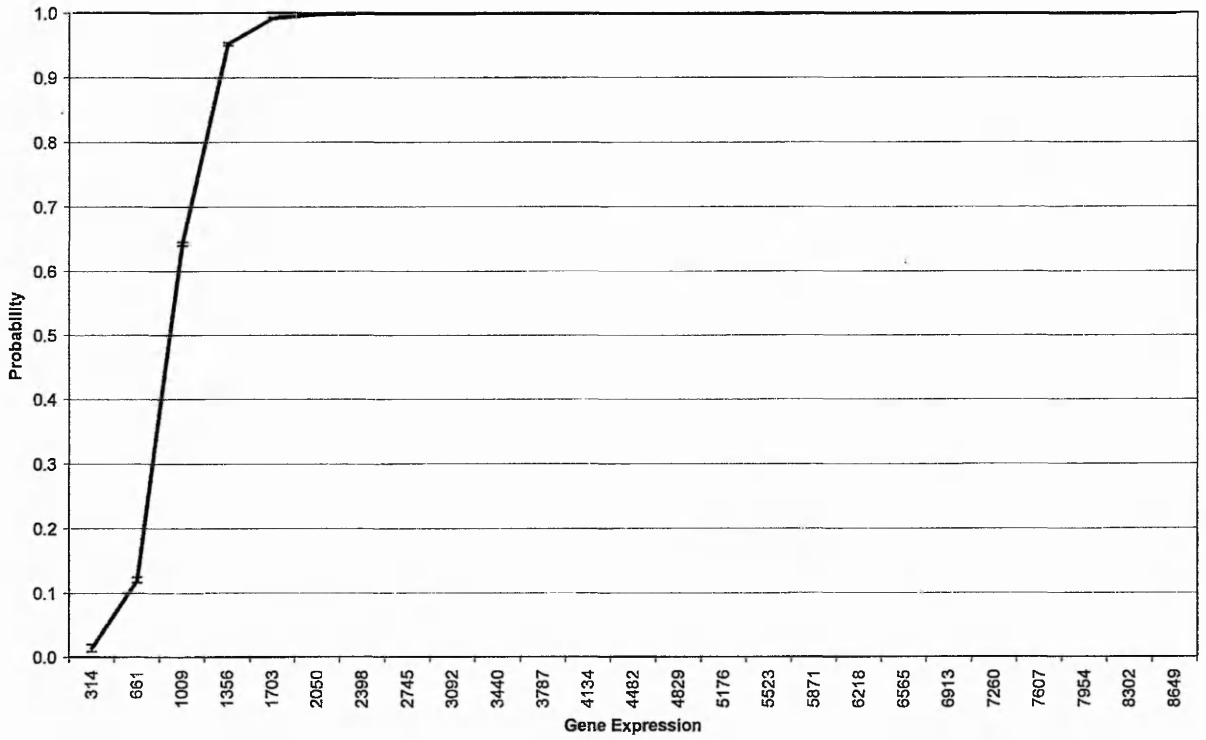
(b)

Gene Description	Response
Bacteriophage P1 cre recombinase	†
Homo sapiens I-Rel mRNA (RELB)	*
PSMD5	‡
BLM	†
Human Rad50 mRNA	†
Homo sapiens complement component properdin mRNA	†
Human N-sulphoglucosamine sulphohydrolase (SGSH) mRNA	†

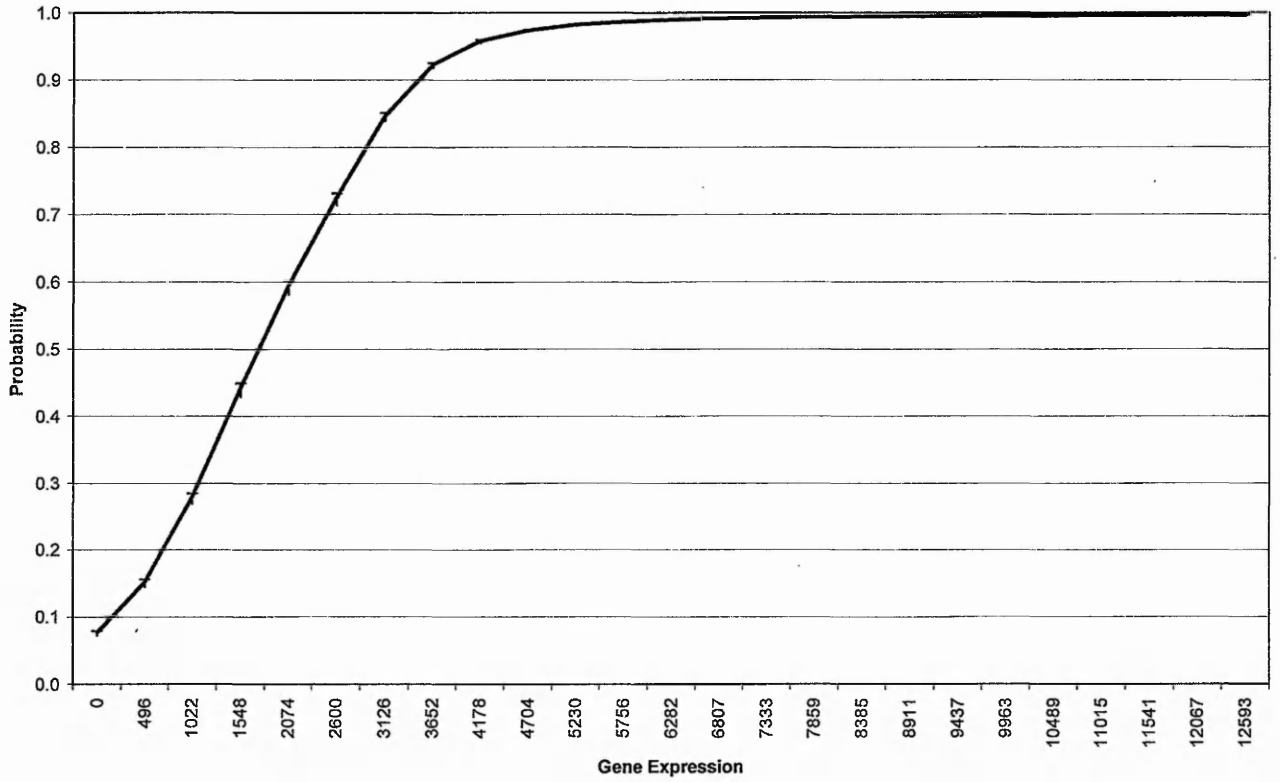
The description of the response is summarised with regards to the probability of whether a patient is or is not LN positive.
* Increased gene expression leads to increased probability of LN+ status.
† Increased gene expression leads to increased probability of LN- status.
‡ Weak response, possibly interacting to modify response of other genes.

Figure 39a-d Response curves for (a) I-Rel mRNA, (b) GATA3 mRNA, (c), 5T4 and (d) Human cytochrome P450 gene. Figures show the gene expression ratio of each gene plotted against the probability of the patient being either (a) LN positive, (b) and (c) ER positive and (d) no change regardless of expression. Error bars indicate 95 % confidence intervals.

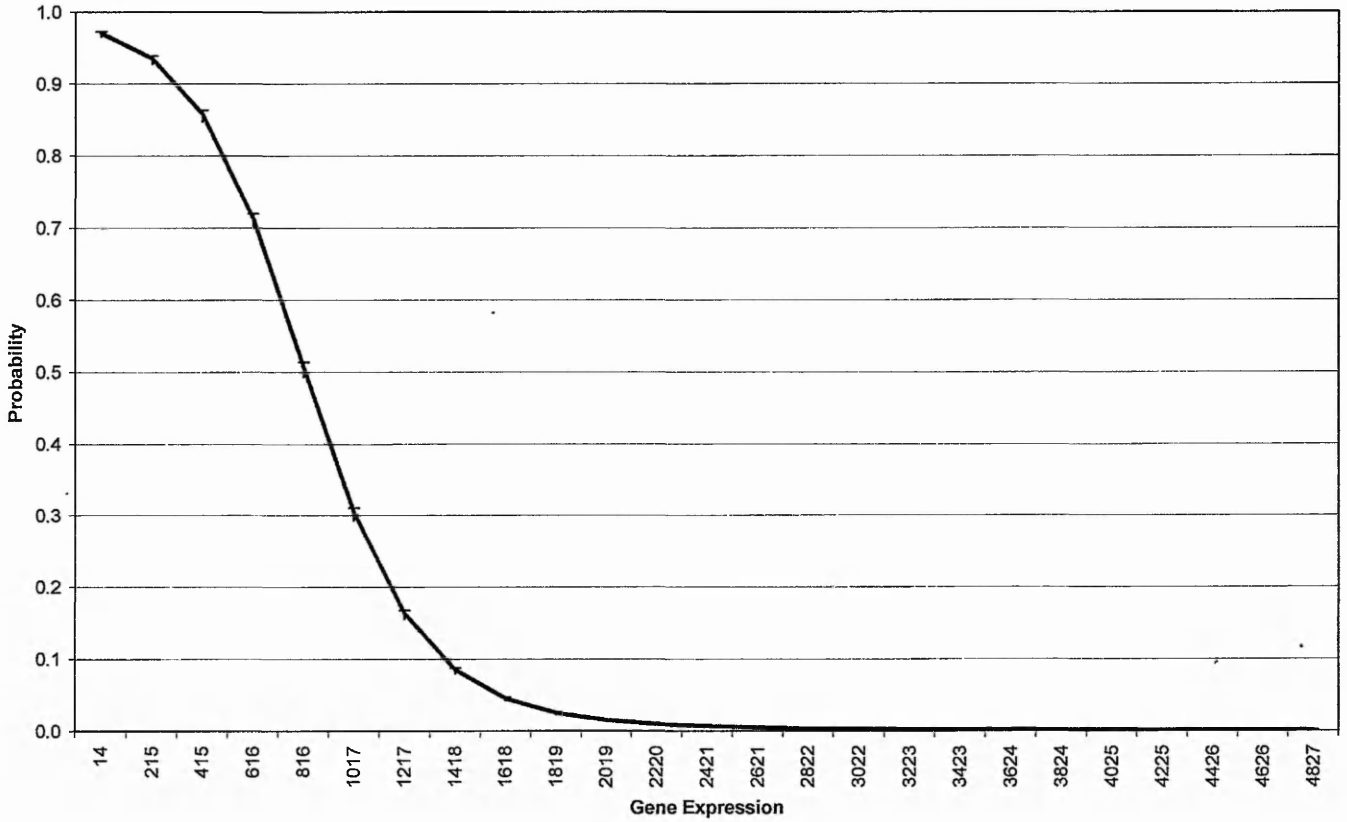
(a)



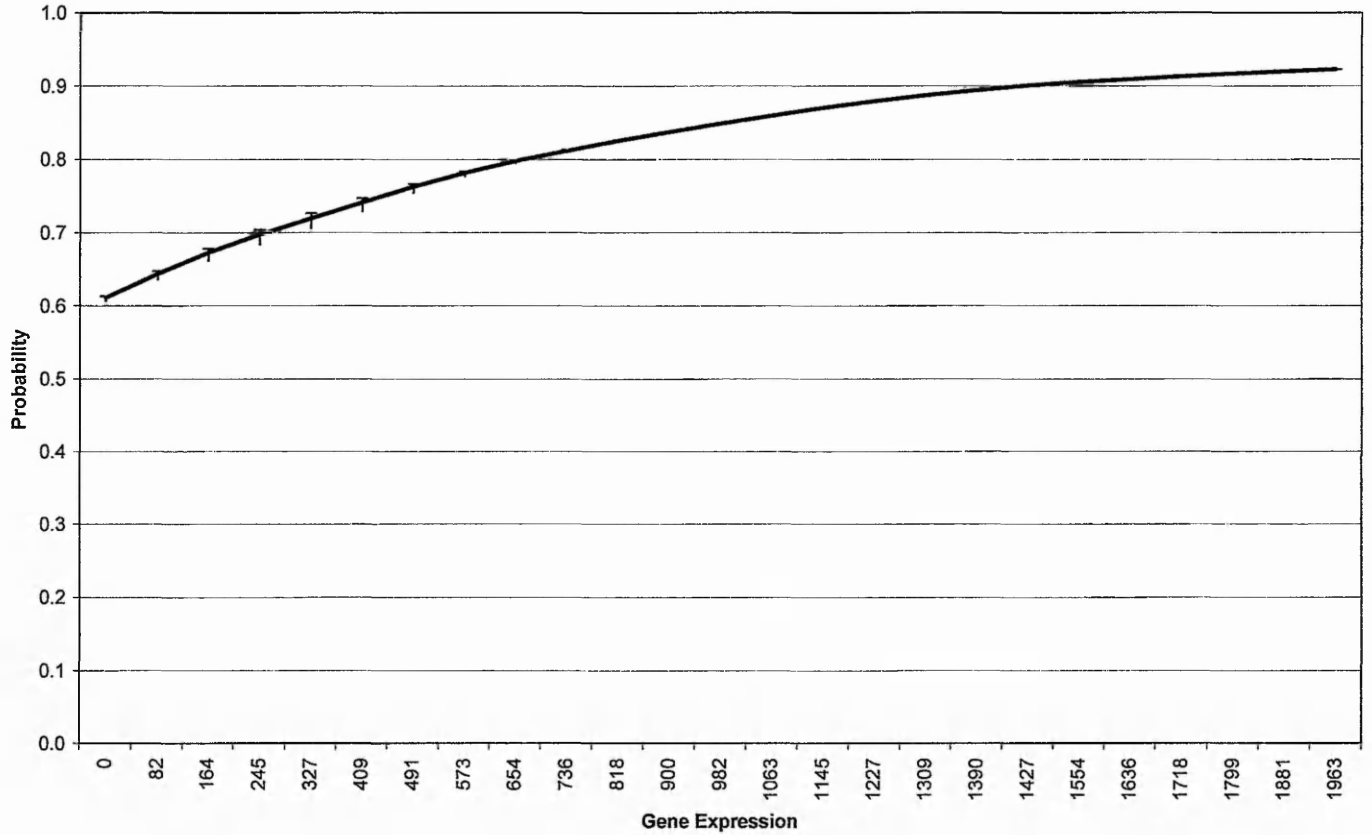
(b)


Probability
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
496
1022
1548
2074
2600
3126
3652
4178
4704
5230
5756
6282
6807
7333
7859
8385
8911
9437
9963
10489
11015
11541
12067
12593
Gene Expression

(c)


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Probability
14
215
415
616
816
1017
1217
1418
1618
1819
2019
2220
2421
2621
2822
3022
3223
3423
3624
3824
4025
4225
4426
4626
4827
Gene Expression

(d)


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Probability
0
82
164
245
327
409
491
573
654
736
818
900
982
1063
1145
1227
1309
1390
1427
1554
1636
1718
1799
1881
1963
Gene Expression

5.3. Population analysis

Breast cancer is a morphologically and clinically heterogeneous disease (Garcia-Closas, et al., 2006; Jeffrey, et al., 2005; Murphy, et al., 2005), and therefore includes several different subgroups (Martin, 2006). With this in mind, it was of interest to determine the structure of the population under analysis in order to identify these sub-groups, or sub-populations of samples associated with disease. This can be accomplished by monitoring the network predictions in a continuous manner, which offers the potential for diseases to be monitored in a patient specific manner with the ability for new individuals to be positioned within this population structure. Furthermore this allows for risk analysis to be carried out by determining the location of a new individual within the population.

5.3.1. Methods

The actual ANN model predictions were assessed for each individual sample over 50 sub-models generated from random resampling. These values were taken from the predictions where the sample was treated as validation data and the average prediction from this series of sub-models was calculated. These were then ranked in ascending order, and the position of every sample case within the disease population was determined. This enabled varying subgroups of patients to be identified.

5.3.2. Results

The population analysis for the West data (LN status) and the van't Veer data can be seen in Figures 40 and 41 respectively. These population structures can be viewed as to contain several sub-populations. There are those which are classified as clear negatives (with a predicted value between 1.0 and 1.5). Next there are those showing strong characteristics of, and are therefore showing profiles very similar to a patient case belonging to the positive group (with a predicted value of 1.5 to 2.0). Then there are those who are close to the classification threshold value of 1.5. These cases are potentially at risk of metastases, and have been identified as a further sub-population of patients warranting further analysis. It is evident that the population in Figure 40 detailing LN status was more clearly defined than that in Figure 41 regarding distant metastases. This is to be expected, as it seems reasonable to suggest that distant metastases is a more difficult phenomenon to predict than LN metastasis, as with distant metastases, the disease may have spread anywhere in the

body, and not locally defined to the lymph node. Figure 41 shows a clear sample progression between the predicted values of 1 and 2, suggesting that a continuum exists between patients that do and do not show signs of disease spread. If new cases were to be placed on this continuum, it would be feasible to imply that the further up this population profile they are situated, the more at risk they are of developing distant metastases if they have not already done so.

Figure 40. Population analysis for LN status. Figure shows known LN- patients in blue, and known LN+ patients in red. Y-axis shows the probability of the disease having spread to the lymph node. Error bars indicate 95 % confidence intervals.

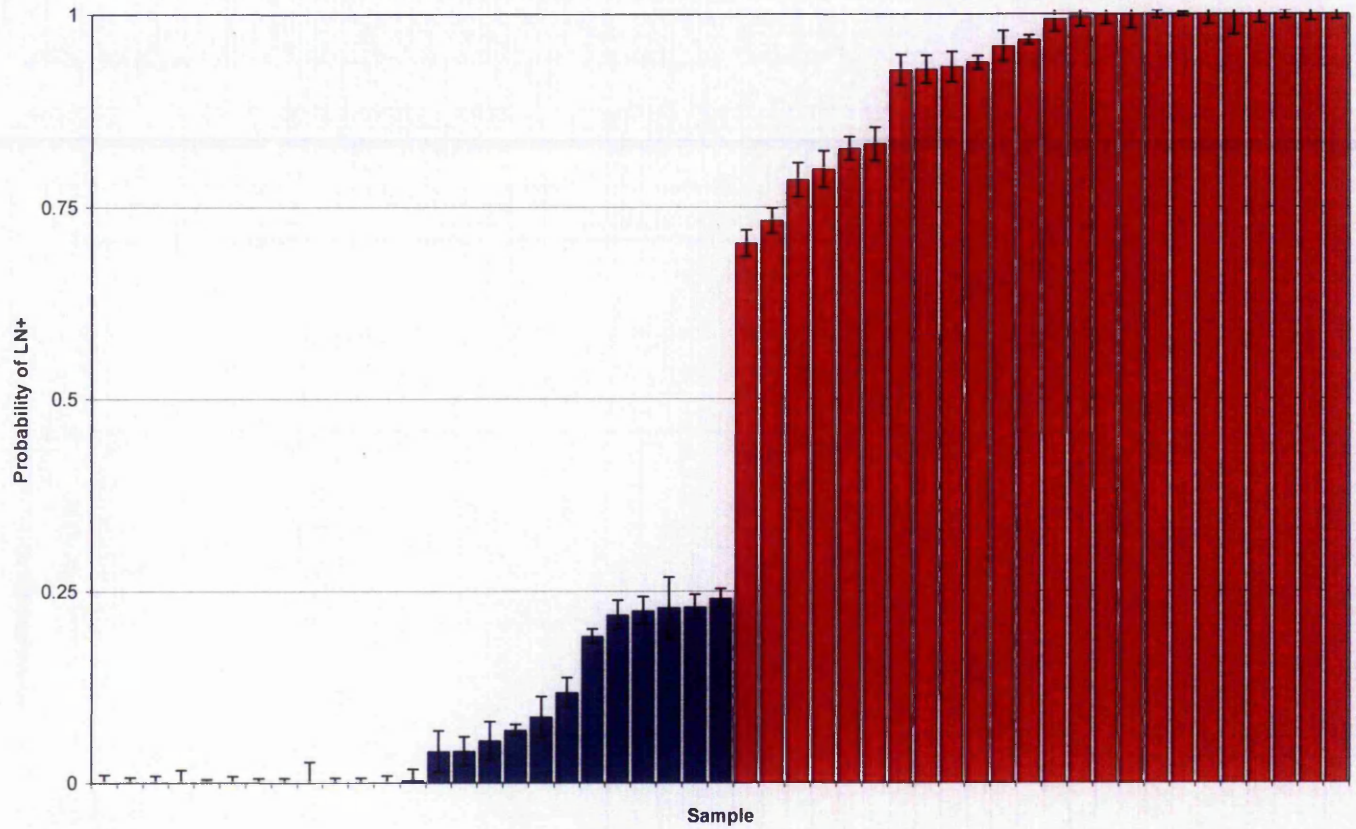
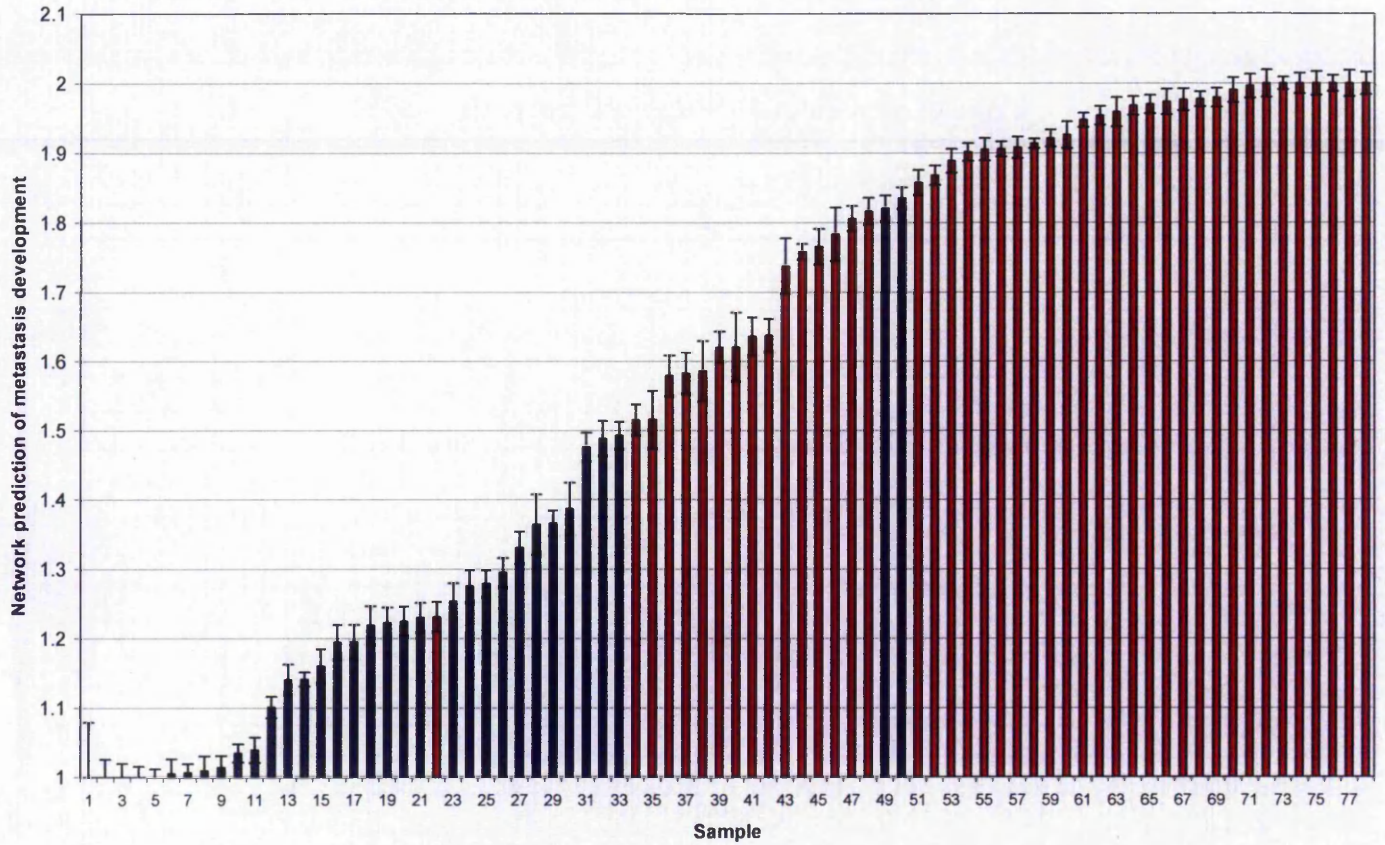


Figure 41. Population analysis for disease metastasis status. Figure shows patients who developed distant metastases in red, and those who did not in blue. Error bars indicate 95 % confidence intervals.



5.4. Discovering gene interaction networks

5.4.1. Introduction

In addition to biomarker identification, there is now a growing interest in a more systems biology approach to bioinformatics, where the focus is on how molecules and pathways interrelate (Guffanti, 2002), so that all of the components in a biological system can be modelled and examined. However, the inference of gene regulatory networks from experimental data still remains one of the main unsolved problems in the post-genomic era. A gene regulatory network is a model representing dependencies between genes using a directed graph. Here, each node is a gene (or component) of the system of interest and each edge represents a regulatory impact from one component to another (Spieth, et al., 2005), for example the up or down regulation of the dependent gene.

With the identification of multiple subsets of biomarkers which all appear to be important in the classification of disease status, it was of interest to determine whether ANNs could be used as a method for elucidating potential interactions between these biomarkers and whether certain markers are altering the expression of others in a positive or negative manner. The stepwise approach to biomarker identification has been shown in the previous chapter to be successful in the discovery of multiple subsets of biomarkers from complex datasets that have the ability to classify disease status to high accuracies for blind data. Experiments on the multiple subsets of genes identified using this approach within the West et al. dataset show the potential of this proposed approach for discovering gene network interactions representing ER and LN status in a population of breast cancer patients.

The aim of this approach therefore is to identify the strength of the interactions between genes, and whether these connections are excitatory or inhibitory. A draft gene regulatory network can then be produced showing whether one gene influences either the increased or decreased expression of another gene with regards to disease status within the context of the biomarkers identified. This is achieved by analysing the strength of the interconnecting weights in an ANN model. This ANN model consists of a subset of genes identified using the stepwise approach and looks at how well these genes are able to model the expression values of another gene. That is, each gene is sequentially used as an output in the model whilst the remaining genes in the subset are used as the inputs. When the strength of the

weighting between $gene_i$ and $gene_j$ is outside a normally distributed range, the connection is considered to be a regulatory one.

5.4.2. Methods

The algorithm works in the following manner, and is summarised in Figure 42. For each gene subset identified using the stepwise approach:

1. Start with gene 1 as the output gene, $gene_j$
2. Use the remaining genes as inputs ($gene_1 \ldots gene_2 \ldots gene_3 \ldots gene_n$)
3. Train the ANN until the input genes classify the output of $gene_j$ to a pre-determined stopping criterion.
4. Determine the weights between each input $gene_i$ and output $gene_j$.
5. Those inputs whose weight values are +/- 2 standard deviations outside of the mean weight value for all inputs are considered to regulate $gene_j$ in either a positive or negative manner.
6. Repeat from step 1, with gene 2,3,4…*n* as the output gene, $gene_j$

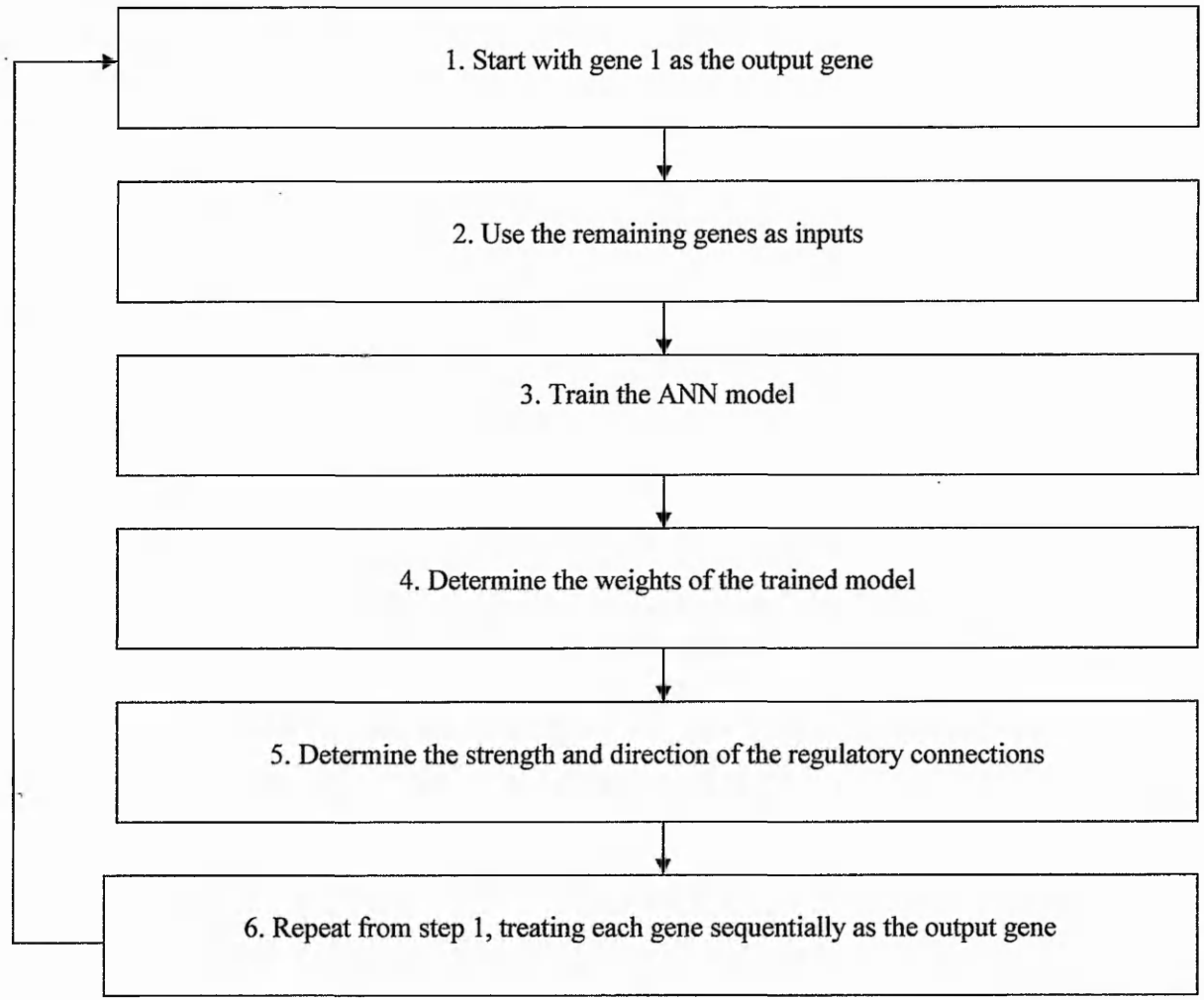
Once a matrix of weights for each input gene to the output gene had been derived, it was then possible to generate an interaction network for the genes in the dataset. The gene subsets may then be amalgamated together at points where the same gene is shared across multiple models. This then provides an overall representation of how important gene subsets constitute their own motifs, and also shows how these motifs possibly link together in relation to the gene of interest.

5.4.3. Results

5.4.3.1. Predicting the expression values of genes

The different subsets of biomarkers identified using the stepwise approach when analysing the West dataset were used to develop the interaction models. This consisted of 10 gene subsets predicting ER status, and 10 predicting LN status. Each gene subset contained between 7 and 11 genes. The ANN was run as described above in the methods, with each gene being sequentially used as the output to the model, with the remaining being treated as inputs. Results from ANN training showed that the expression of all genes could be

Figure 42. Summary of algorithm for the derivation of gene networks



predicted successfully with low errors. Figure 43a-b shows example observed v predicted plots for the first genes identified in predicting ER and LN status.

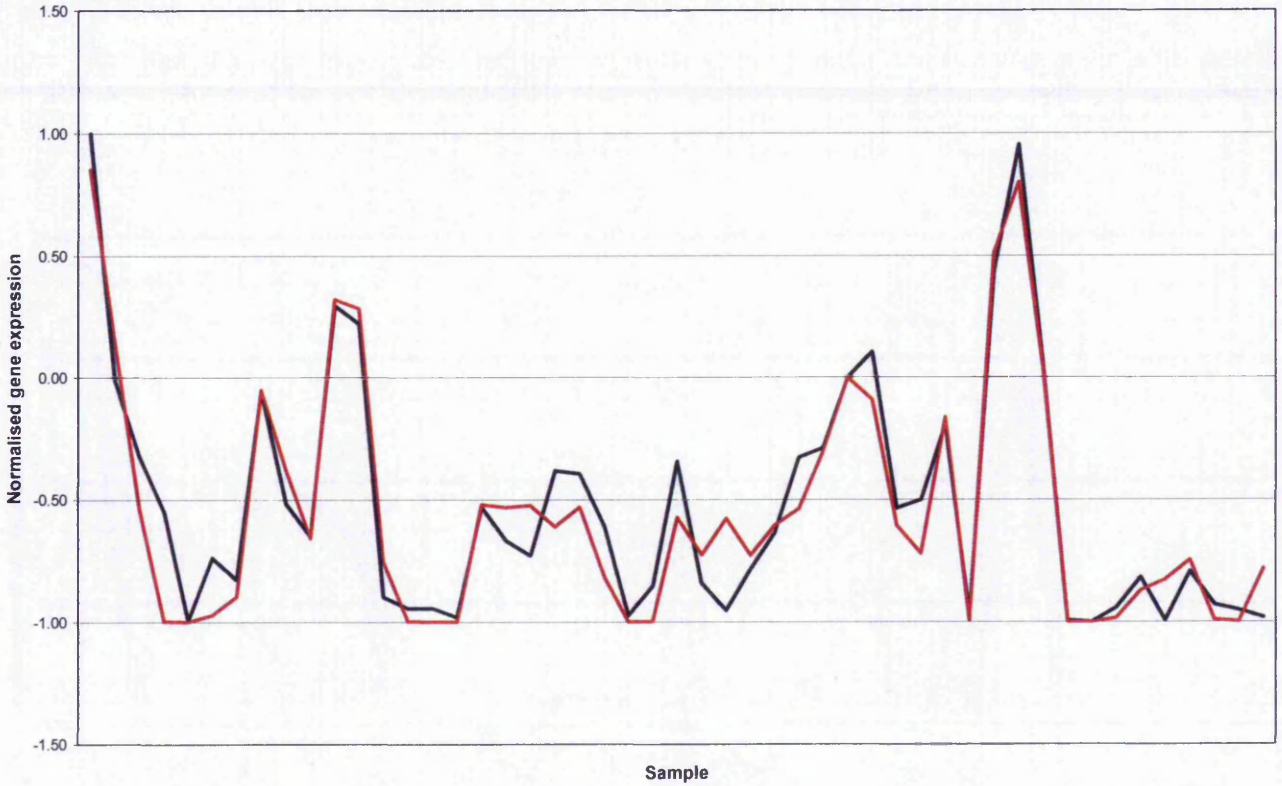
5.4.3.2. Derivation of the interaction network

The weight matrix for the series of models developed in the interaction process was imported into a visualisation program called Cytoscape. This is an open source bioinformatics software platform for visualising molecular interaction networks and integrating these interactions with gene expression profiles (http://www.cytoscape.org). Figure 44a-b shows the interaction network for both ER and LN models. Red connections indicate positive excitatory weightings thus showing genes which appear to cause the increased expression of other genes. Blue connections indicate negative inhibitory weights, showing genes that when expressed in a certain fashion cause the subsequent decrease in expression of other genes. The thickness of this connection indicates the strength of interaction between two genes. It is clear from Figure 44a that the ER interaction network appears less complex than the LN network. There appears to be fewer genes in the ER network linking the different gene subsets together, and as such these individual motifs appear to be more segregated in this system. The LN interaction network appears to be more complex, with several gene subsets overlapping with many genes. Interestingly there appears to be a small cluster of genes in the centre of both networks which are the focal point of the network, having strong interactions with and appearing to regulate a whole host of other genes.

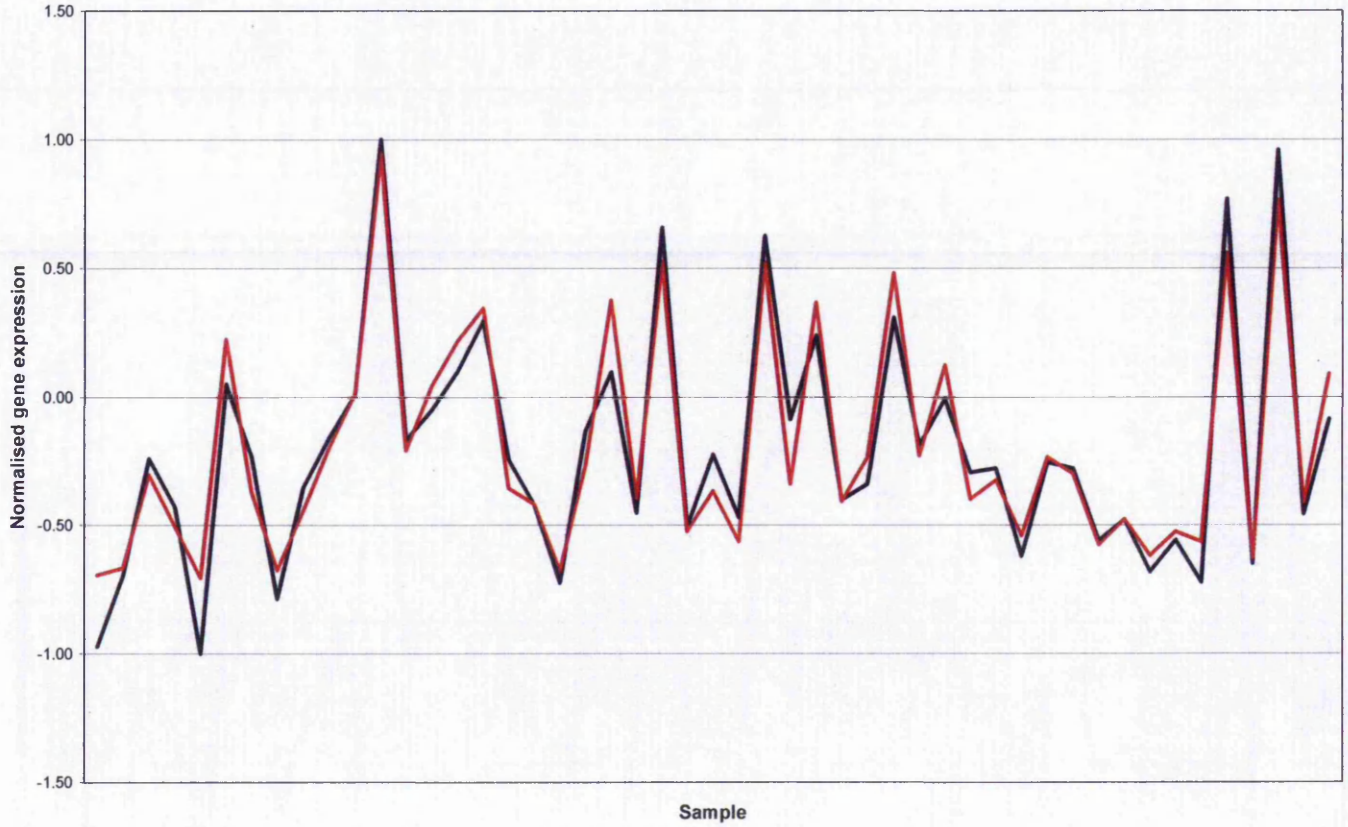
Once derived, this gene interaction network may then be interrogated further. The purpose here is to define potential pathways that were previously unknown. From a clinical perspective this may be fundamental in the identification of novel biomarkers to target in order to prevent the spread of disease. An example of this pathway interrogation can be seen in Figures 45 and 46. Here, one of the biomarker subsets has been examined in more detail and combined with a response curve analysis to determine how the effect of one gene on other genes governs their expression and outcome response. This enables a possible pathway of events contributing to an outcome of interest to be determined. Figure 45 focuses in on a region of interest from the interaction network shown in Figure 44a.

Figure 43a-b. Observed v predicted plot of gene expression values of the first gene identified as important in the classification of (a) ER and (b) LN status. Blue line indicates known observed gene expression value, whilst red line shows the ANN predicted value.

(a)



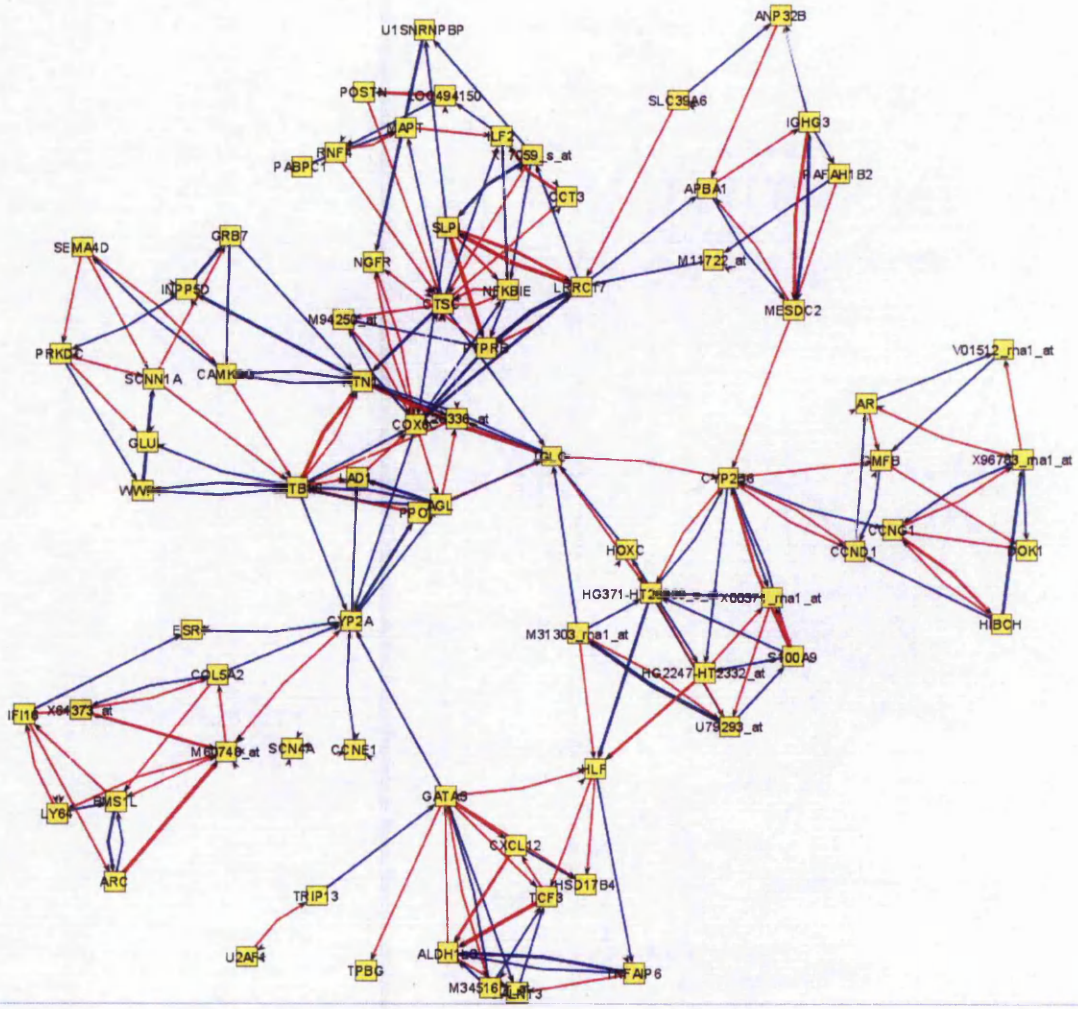
(b)


1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
Normalised gene expression
Sample

The response curves for each of these genes are also shown. This shows how the up or down-regulation of a particular gene is influenced by another, and how this relates to the response of this gene with respect to the class of interest. From this it appears that the up-regulation of HLF causes the subsequent increased expression of GATA3. This has a cascade effect, causing the change in expression of various other genes, for example GALNT3 and CXCL12. The change in expression of these genes then in turn leads to the up or down regulation of others. Figure 46 summarises this theoretical pathway, showing the potential for deriving pathways from this type of interrogation.

Figure 44a-b. Gene interaction network for (a) ER and (b) LN status. Red connections indicate positive weightings between two genes, whilst blue connections show negative connections. The weight of the line indicates the level of interaction between two genes.

(a)



(b)

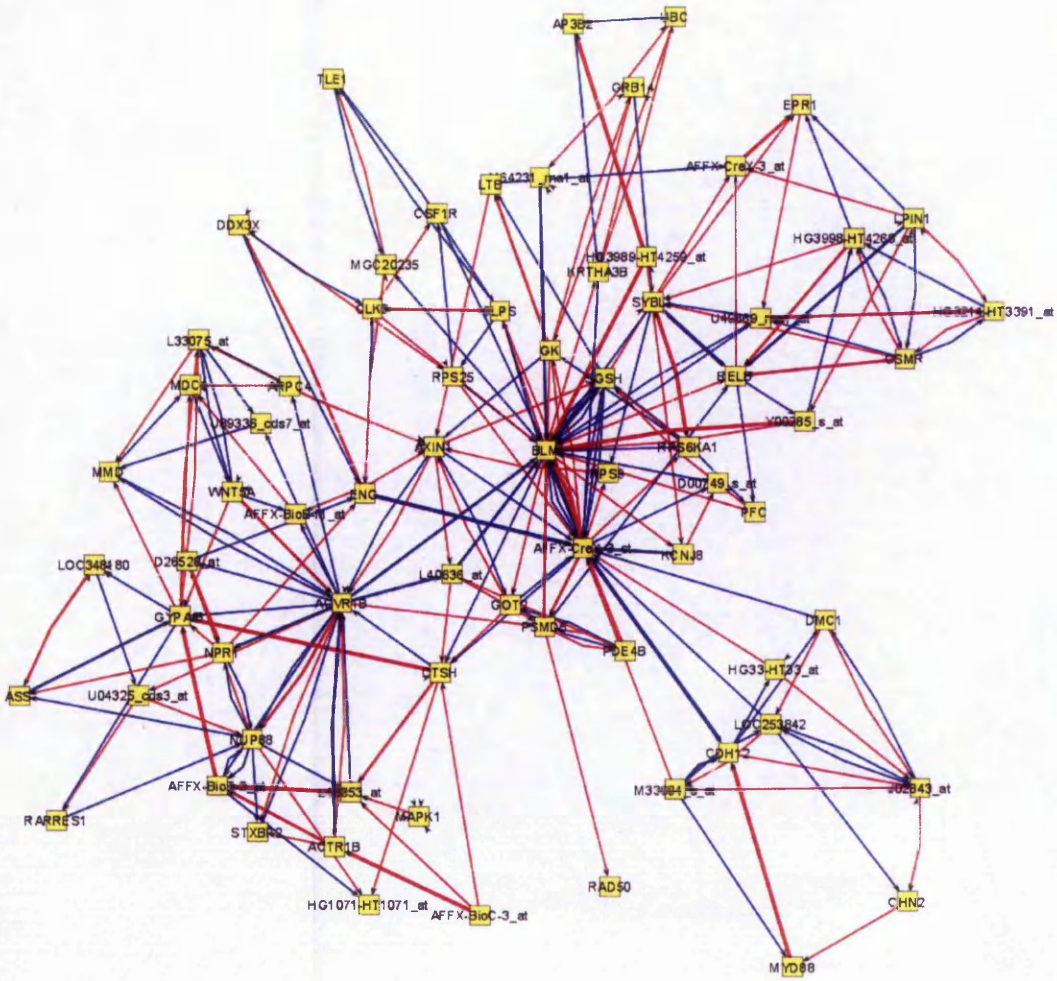

TLE1
EPB41
EPR1
DDX3X
CSF1R
CPN1
MGC20235
SYBU
L33075_at
GK
RPS25
GSH
BELP
DSMR
MDC1
ARPC4
AXIN2
BLM
MMP2
WNT5A
ENG
PFC
LOC348480
D26520_at
AFFX-Crex3-3_at
RCNJ8
GYPA
NPR1
PSMD8
PDE4B
DMC1
ASS
U04325_cds3_at
CTSH
HG33-HT33_at
LOC253842
CDH12
NUP88
RARRES1
STXBP2
ACTR1B
MAPK1
RAD50
CHN2
MYD88
HG1071-HT1071_at
AFFX-BioC-3_at

Figure 45. Interrogation of the interaction network. Interaction connections are shown together with the response curves of the genes of interest.

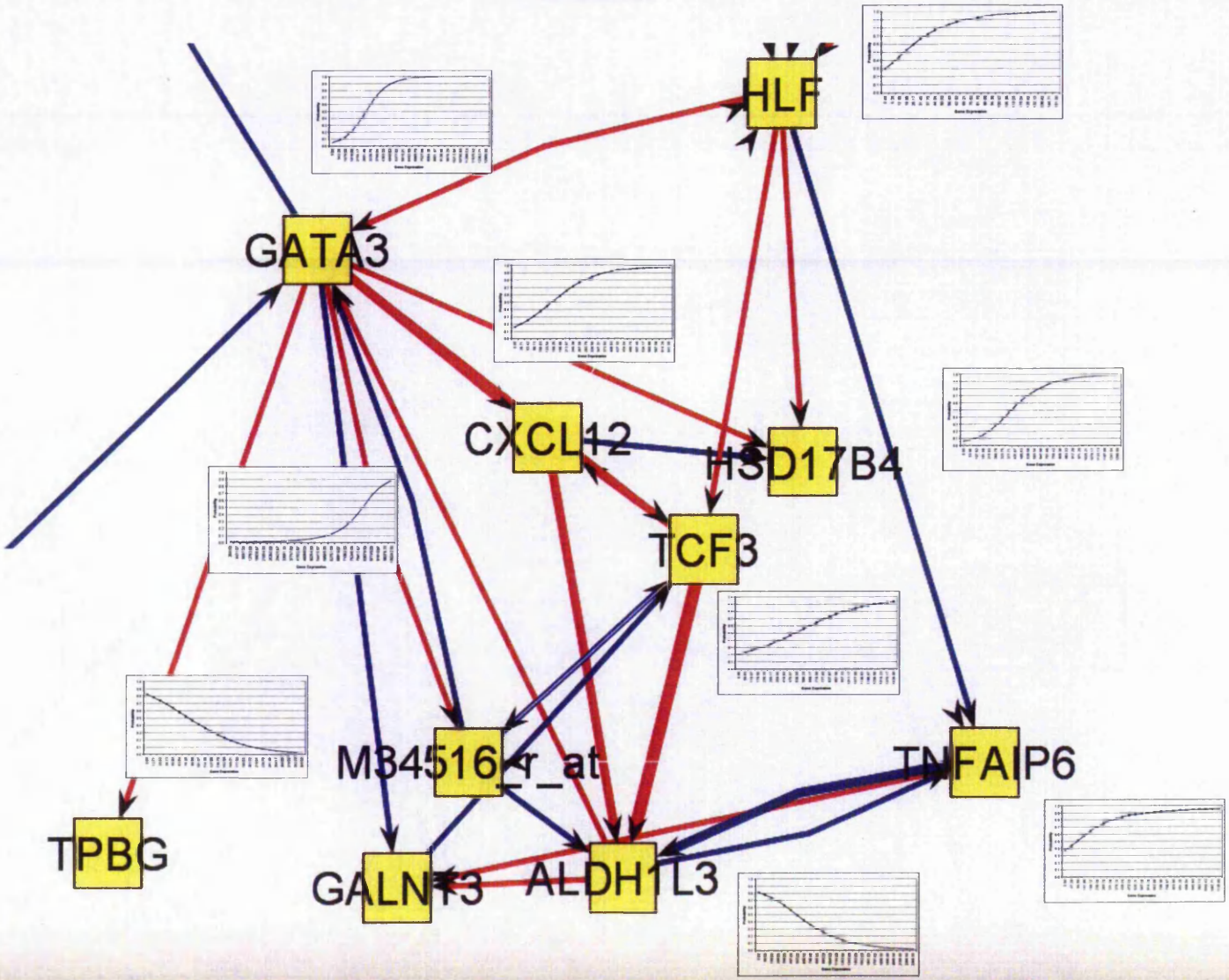
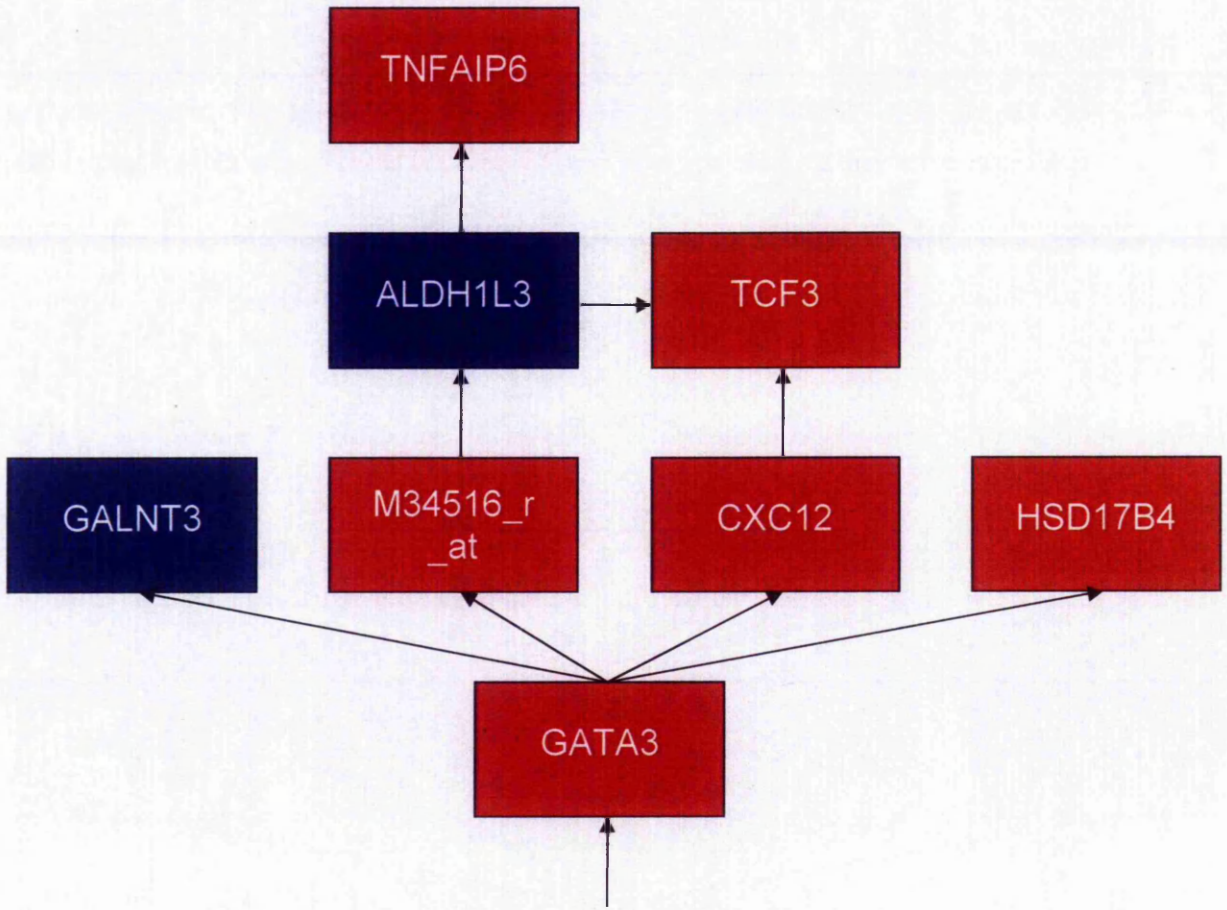


Figure 46. Summary of example theoretical pathway. Genes surrounded by a red box indicate they are up-regulated as a result of the gene at the previous step. Genes surrounded by a blue box indicate they are down-regulated.



5.5. Discussion

Analysis of the response curves for the individual biomarkers showed how the output of the ANN changed when the input value of a particular input variable was varied over a given range. This allowed for the biomarkers to be characterised by determining how an increase or decrease in their expression governs a disease outcome of interest. The response graphs showed that the response of these biomarkers could be positive, negative, discriminatory and co-factorial. The one limitation of this approach is that whilst one input is varied, all others remain fixed. This would not be the case *in vivo* where a change in expression of one gene would affect the downstream expression of others. However, the resulting directions of these responses appear to concur with what has been previously reported in the literature, which provides further support to the proposed roles these biomarkers may have in cancer progression, and that the information that these response curves are providing appears to be a true reflection on the biological nature of these genes. Additionally this method potentially enables rules to be extracted from the ANN model by establishing the biomarker threshold at which the prediction changes from one class to another. Simple IF AND THEN rules may then be extracted; for example:

IF gene x expression is $> i$ AND
gene y expression is $< j$ THEN
ANN output = 1

This would have a potential use in a clinical setting for molecular classification of tumours, and in turn helps to shed light on the often described ANN "black box". This shows that they are indeed not black boxes, as rules and even formulae can be easily extracted from trained ANN models, which can then be applied to future cases in a rapid and simple manner.

Analysis of the population structure within a dataset allows for the positioning of individual samples based upon their similarity with regards to the ANN model predictions over a number of RSCV sub-models. Using this method it becomes possible to be able to not only classify new cases into a particular group, but to pinpoint exactly where they lie within the population. Here there is the possibility of identifying new sub-groups within diseases of interest and potentially extending traditional clinical diagnosis of diseases such as cancer, whilst at the same time identifying sample outliers within the population. This

type of analysis can then be combined with that conducted in previous chapters using visualisation techniques such as PCA and cluster analysis in order to determine whether the population structures identified using a variety of methods show correlations with one another. This allows for the potential to identify cases that are diagnosed as negative but are also showing characteristics (based on their profiles) belonging to the positive population. These individuals may be then defined as a "higher risk" sub-population. This approach could be fundamental in the development of novel prognostic and diagnostic markers and a broadening of clinical classifications, not only reducing the risk of false positive diagnoses, but also identifying risk of disease development and progression in a patient specific manner.

An approach aimed towards the identification of potential interaction networks from identified gene subsets was then described. This provides information about the components involved in the system of interest, and how they might regulate this system by interacting with one another. This method was shown to be successful in accurately modelling the data on which it was trained, that is using gene expression values to model the expression values of other genes identified in the same gene subset. The method shows how genes appear to regulate one another, causing the up- or down-regulation of others, and how gene subsets are potentially linked together by individual genes. Additionally, the strength of these interactions can also be determined, and these strength thresholds may be varied according to the level of interaction one is interested in. For example the network may be altered to only include those interactions which are extremely strong. The one disadvantage here is that there are currently no known interaction networks for these systems for the network derived here to be validated against. However, the approach provides valuable information on how subsets of important biomarkers may link together, and how each motif appearing in the network may represent an individual pathway. The expression of genes within these pathways may be important to disease status, and potentially disease progression. Genes which appear to be the focal points of these networks may be potential targets since they appear to be at the root of the problem. For example those genes found to be important in lymph node metastasis may be targeted in an effort to prevent this disease spread. When the network is interrogated further, one is able to focus in on particular regions of interest, and derive potential pathways from this information. This was shown in this chapter, where a section of a gene network was investigated to generate a pathway of events. It was shown how the expression of one gene appears to affect the expression of others, which subsequently causes a cascade effect

of increased or decreased expression of many other genes. Interestingly, many of the genes examined here have been previously linked to breast cancer, further validating their identification here. For example, several ER positive human cancer cell lines have been shown to express the HSD17B4 gene (de Launoit and Adamski, 1999). Lacroix and colleagues (Lacroix, et al., 2004) described the GALNT3 gene to be inversely correlated to ER expression in tumours. Sreerama and Sladek (Sreerama and Sladek, 1997) showed that ALDH1 levels varied over a wide range in primary and malignant breast malignancies, and they proposed that measuring the level of this in normal breast tissue would be of high predictive potential with regards to treatment. Moreover, it has been shown how target organs for breast metastases are enriched in the chemokine CXCL12, which is overexpressed in the metastatic process of breast tumours (Ben-Baruch, 2003). Finally, TNF-alpha, involved in the inflammatory response (Azmy, et al., 2004), has been shown to be tumourigenic both *in vitro* (Komori, et al., 1993) and *in vivo* (Fujiki, et al., 1994; Suganuma, et al., 1999). Furthermore high plasma levels of TNF in cancer patients has been associated with poor prognosis (Nakashima, et al., 1998).

Various other methods for deriving gene regulatory networks have been proposed in the literature, which include rule based formalisms and Boolean networks (de Jong, 2002) although most applications appear to favour the use genetic algorithms to construct the network (Huang, et al., 2005). However genetic algorithms have been shown to perform particularly poorly on this type of complex data (Spieth, et al., 2005). Bar-Joseph (Bar-Joseph, 2004) discussed that current methods for deriving regulatory networks from gene expression data from DNA microarrays were limited due to the large numbers of genes involved, and as such only small networks have been inferred be computational means. This thesis introduced a method which overcomes this problem by initially identifying the optimal topology of the network using the stepwise approach described in the previous chapters. This then reduced the number of parameters to be used for creating the gene regulatory network to just the most important gene subsets capable of classification of the data. It is then these gene subsets which are used for the derivation of the gene interaction network, which considers all components together in a single network. Although it is accepted that these are preliminary models of the system which need validation by further research, the potential is evident for the development of these methodologies towards an integrated modelling, biomarker identification and systems biology approach to biological data analysis.

CHAPTER 6. OVERALL SUMMARY AND DISCUSSION

This study has focused on the development and validation of ANN based methodologies geared towards analysis of biological systems in order to gain an insight into diseases of interest. ANNs were chosen to be the main focus of these studies due to their many advantages within the context of the aims of this thesis. For example their ability to model complex nonlinear data is essential as this type of dataset is becoming commonplace in biological research. Their proven ability to be able to generalise well to new cases is crucial when developing models for diagnostic and prognosis purposes. Furthermore the way in which ANNs are accessible for interrogation was perhaps one of the most important reasons they were chosen here as this enables variable importance to be determined, and biomarkers to be identified and characterised. These may then be examined further to deduce the possible function of these markers relative to a disease of interest. It is well established that disease processes can be highly complex, and this research is aimed at developing methods which identify biomarkers that have prognostic and diagnostic potential. Furthermore, methods focused at interrogating these biomarkers have been described, leading to further understanding of how these may be changing in response to their role in disease and how they may interact with one another.

The main aims of this research were as follows:

- To use ANNs to develop and validate predictive models from real biological datasets.
- To determine the importance of individual variables within these models, thus deriving meaning from the ANN "black box".
- To design, write and implement software tools for the automated identification and characterisation of key subsets of biomarkers, and the subsequent validation of these methods.
- To show how the biomarkers identified can be interrogated further still, in order to characterise their behaviour with regards to the disease of interest.
- To investigate the potential use of ANN modelling in identifying gene networks and interactions between biomarkers identified in these complex systems.

6.1. ANNs in the development and validation of predictive models

Network parameter settings of the ANN architecture were trialled and tested to determine a set of parameters which were appropriate for the type of complex data used throughout this study. Based upon these findings, the basic ANN parameters used for the analysis carried out were:

Learning rate of 0.1.

Momentum value of 0.5.

Hidden layer size of between 2 and 5 nodes.

Data randomly split into subsets of 60 % training, 20 % test, and 20 % validation.

Initial weights of the network randomised between 0 and 1.

Training terminated once network fails to improve on the test error for 20,000 epochs.

Once these had been determined, predictive models were developed using absolute weightings analysis to parameterise input variables in order to reduce the complexity of the model whilst maintaining predictive performance. This was shown to be a successful approach in generating predictive models from MS analysis of bacterial pathogens. The models developed were shown to be extremely robust when validated against a further set of samples, classifying 98 % of these samples correctly. Additionally PCA and clustering approaches were utilised in order to gain further understanding of the data structure, Interestingly, the samples misclassified by the ANNs were shown by PCA and clustering to be more closely related to their ANN predicted class than their actual assigned class suggesting a population continuum existing between the different species. However, there are limitations to this approach caused by the dimensionality of the data generated with this type of analysis. With this type of parameterisation although one can obtain a measure of importance of each input variable used, the difference in the importance value from one input to another is often extremely small due to the magnitude and dimensionality of the data. Therefore the decision concerning which inputs to retain and which should be removed remains somewhat arbitrary when this high dimensionality is considered. The meta-pruning approach summarised in the next section was designed to address these problems by only retaining input variables which were consistently identified as important in classification over a number of methods, increasing the confidence that a given input is of true value to the model.

6.2. Determining the importance of input variables within the system of interest

A comparison of several input pruning approaches was conducted to assess how inputs identified as important in different approaches performed in the classification of different datasets. Here, weightings analysis using the actual values of the interconnecting weights produced the more accurate models when compared with absolute weightings analysis and sensitivity analysis. This in turn led to the assessment of a proposed "meta-pruning" approach. Here only those inputs of importance in all of the methods tested are selected, and are then combined together for further training. This identified candidate biomarkers which were consistently influential in the predictive performance of the model. This meta-pruning approach significantly out performed the traditional methods in both of the datasets used here for validation. This chapter of the study showed how information relating to input importance is readily accessible from the trained ANN models, and by using this information, complex datasets may be parameterised to generate simpler and more robust predictive models. Although this approach overcomes the limitations described in the previous section, it is still often difficult to determine how one balances model simplicity against model performance. This often makes it difficult to decide at which point to stop removing inputs which appear to be of less importance in the model, resulting in the final biomarker subset. Therefore the stepwise approach to biomarker identification summarised in the next section was designed to overcome these issues by identifying biomarker subsets whose inclusion in the model results in an increase in classification for blind data.

6.3. The design and implementation of software tools for the automated identification of key subsets of biomarkers

Novel stepwise ANN approaches to data mining of complex highly dimensional datasets have been developed to allow for the identification of a defined subset of biomarkers with prognostic and diagnostic potential. Software tools were created for the automation of identifying biomarkers that result in high classification accuracies. This approach was then applied to different datasets to validate its potential application. Here, subsets of biomarkers were identified which were capable of predicting disease status to very high accuracies (> 95 %) in a variety of analyses on different types of dataset which were focused toward the diagnosis and prognosis of different stages of melanoma and breast cancer. Further investigation of these biomarkers showed that many of these had

previously been associated with cancer. With the stepwise approach, biomarkers identified subsequently to the marker identified at step 1 are ordinal to one another, explaining different aspects of the variation in the data space. Therefore although they may have less overall influence on classification than the first marker, they are explaining variation not previously described. Inputs may be ranked at step 1 to give a relative importance value similar to that achieved when using the parameterisation methods described previously. Furthermore, the order in which these inputs appear when they are ranked according to their performance was shown to be consistent when validated by repeating the same analyses multiple times. The method was also validated using separate additional validation datasets, as a second order validation of the approach. Once again, high classification accuracies were seen adding further evidence that the markers identified are of true relevance to the system being modelled.

Additionally, it was shown that there is not a single solution to a particular problem, and many biomarker subsets exist within these systems which have the ability to predict the class of interest to high accuracies. Many of the genes identified here appeared in several biomarker signatures, further validating and highlighting the potential importance of their role in disease. Stepwise approaches in ANN modelling have been previously described (Addison, et al., 2004; Cottrell, et al., 1995), however these studies showed their use more specifically in dimensionality reduction by identification of the importance of variables within a system. The stepwise approach described in this research is aimed more specifically towards biomarker identification, and was designed to help overcome some of the limitations associated with approaches focused on the parameterisation of datasets. The stepwise approach is not affected by these limitations since it takes each input variable into consideration and monitors how its independent addition to the model affects network performance, and as such is not adversely affected by the inherent noise commonly associated with these highly dimensional datasets. It was also shown that this approach was capable of generalisation to new data not originally included in the model development process, and even to new datasets where the patients originated from a very different background to the original. This shows that these models are not merely overfitting the data, and limitations associated with linear based stepwise models discussed in Steyerberg et al. (Steyerberg, et al., 1999) are not applicable here.

Another limitation is the false detection of input variables of importance, so it was essential to show that the proposed stepwise approach developed here was not falsely

identifying random input subsets as a consequence of the highly dimensional datasets used. To achieve this, various validation exercises were conducted which showed that the biomarkers identified using these methodologies were not a random selection of input variables, but were consistently identified as important over multiple analyses. This, together with previous reports in the literature of the associations of these biomarkers with the disease of interest, showed that the biomarkers identified with this approach do indeed have biological relevance and confidence in their importance in differentiating between the classes of interest can be high, thus warranting their further investigation.

6.4. Interrogation and characterisation of biomarkers

Once a subset of biomarkers has been identified using the approaches described, it was of interest to investigate them further to understand how their increased or decreased expression affects the classification by the ANN. This was achieved by monitoring the change in ANN output when the input value of a particular biomarker of interest is altered. These responses could be positive, negative, discriminatory and co-factorial. Furthermore, it was highlighted that rules may be extracted from these response curves in order to generate simplistic models which have a potential use in a clinical setting for molecular classification of disease, as such illuminating the alleged ANN "black-box". The main limitation in the response curve analysis is that all input values remain fixed whilst one is varied. This would be untrue in a biological system where changes in the abundance and expression of proteins and genes would result in a change of many others. However many biomarkers identified from the cancer datasets were shown to have previous associations with cancer, and results from the analysis of the response curves were shown to coincide with findings in the literature regarding the changes in expression patterns of these markers. Therefore it was concluded that the information gained from the response curve analysis appears to be a true reflection on the biological role of these biomarkers.

It was also shown how the ANN predictions could be used to derive the population structure of the samples that were modelled. Here it was shown how various sub-groups of cases appeared within the data, which could subsequently be used to identify patients which were at risk of disease spread based upon their genetic profiles. A similar method was shown by West et al. (West, et al., 2001) where the authors looked at the probabilities from their classification model to identify samples that were identified as lymph node negative and whose disease may be poised to metastasise.

6.5. Identifying interactions between biomarkers in complex systems

It was proposed that ANNs could be used to derive potential interaction motifs from the subsets of biomarkers identified as important in the classification of disease. Here, each biomarker was in turn used as an output in an ANN model, with the remaining biomarkers used as inputs and trained to predict the expression level of the output gene. Once the model had trained sufficiently to predict this expression to high accuracies, the strength of the weightings between the input and output genes was monitored in order to determine if there were any regulatory interactions between genes, and if so, the strength of this interaction. This method showed that the data could be modelled in this way very precisely. When the weight matrices were used to derive an interaction network, several motifs appeared to be interacting with one another by clearly defined focal genes within the system. These focal genes look as if they are strongly involved in determining the disease status of an individual, and their expression could be crucial to providing a further understanding of the diseases of interest. Although this interaction work is in its infancy, and requires extensive validation from a biological perspective, these gene networks offer the potential for deriving pathways that show how the change in expression of one gene causes a resulting change in many others. This could be fundamental from a clinical perspective.

6.6. Future directions

Future research will continue to develop the methods described in this thesis, in order to gain more information from these complex disease processes. Some of the markers that were identified as important in the breast cancer and melanoma studies are currently being assessed for their value as diagnostic and prognostic indicators. For example CAIX and AGP are being validated by colleagues on new samples in order to determine their role in a true biological context. Methods are being developed to further interrogate the patient populations within these diseases and methods for extending traditional prognostic methods are underway, with a view to improving current methods in diagnosing diseases such as cancer. To improve the stepwise approach to identifying biomarkers, methods are currently being developed to distribute this processing over multiple machines, so that tens, or even hundreds of computers will simultaneously analyse the datasets, substantially shortening down the processing time required.

6.7. Summary

The development of proteomic and genomic applications for the research into different diseases has paved the way for the development of novel approaches for the way in which these systems can be investigated. As these technologies are relatively cheap and reliable, proteomic and gene expression profiles can be generated allowing for these diseases to be looked into from a whole new perspective. This provides new insight into how proteins and genes are being regulated with regards to different conditions. The limitation of these approaches is the sheer volume of data which they produce, with the majority of this data often noisy and redundant. Therefore these technologies must be coupled with appropriate computational approaches which would then enhance this research further by identifying components of these proteomic and genomic profiles that are the most important in differentiating between disease states of interest. The challenge here is to develop intelligent computational methods which are robust enough to be able to cope with data of this size and nature in order to provide further understanding of these complex proteomic and genomic patterns. This will in turn lead to methods for prognosis and diagnosis of diseases such as cancer, by providing an insight into the proteins and genes which are being expressed differently depending on the current status of the disease.

This thesis has described methods developed using ANNs which allow for the classification of cases based upon an outcome of interest. It was shown how complex datasets can be parameterised in order to identify components of importance between the classes of interest. These approaches were enhanced further with the development of a stepwise approach to analysis, whereby groups of biomarkers are identified which are capable of the accurate prediction of disease class. Models may then be interrogated to investigate how these biomarkers are being differentially expressed with regards to disease status. It was then shown that there are potentially many different subsets of biomarkers capable of the accurate classification of disease status, and that there is not only one solution to a problem of this complexity. These different subsets of biomarkers may then be used in determining gene networks through the accurate modelling of gene expression data. This was achieved by using the weights of these trained models to understand interactions between individual biomarkers and derive pathways based upon these interactions. Furthermore, it was shown how these motifs possibly link together by individual biomarkers which are strongly influential in the prediction of disease class. As with any system the analyses conducted during the course of this research are not without

limitations. These have been discussed and all of the methodological developments within this research have attempted to overcome these problems. Efforts have been taken to ensure that the models are extensively cross validated during development, so as to be generalised to new unseen data which can then be applied back to the model when it is made available. This has led to the training, testing and cross validation of millions of individual models. This enables confidence to be gained that the subsets of biomarkers identified are of relevance to the biological system being modelled and are not an artifact of the high dimensionality of the data itself.

To summarise, this research encompasses the design and implementation of a number of novel approaches combined together to gain a further understanding of disease systems using proteomic and genomic data, providing solutions for problems facing biologists with regards to handling these masses of data being produced. Therefore, the aims set out at the beginning of this research are substantiated by the work contained within this thesis. These aims have all been achieved. This includes the design, writing and subsequent implementation of software for use in the analysis of biological data and the identification of biomarker subsets which correlate strongly to disease status. This analysis has shown how ANNs can be used successfully to identify biomarkers which can then be interrogated further to gain an understanding of the system being modelled with respect to how these biomarkers are responding to a particular disease status, and how their change can be monitored for disease prognosis and diagnosis. As such the results from the experimentation carried out in this work have resulted in novel contributions to the field of bioinformatics.

BIBLIOGRAPHY

Abe, H., Ashizawa, K., Li, F., Matsuyama, N., Fukushima, A., Shiraishi, J., MacMahon, H. and Doi, K. (2004) Artificial neural networks (ANNs) for differential diagnosis of interstitial lung disease: results of a simulation test with actual clinical cases, *Acad Radiol*, 11, 29-37.

Adam, B.L., Qu, Y., Davis, J.W., Ward, M.D., Clements, M.A., Cazares, L.H., Semmes, O.J., Schellhammer, P.F., Yasui, Y., Feng, Z. and Wright, G.L., Jr. (2002) Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men, *Cancer Res*, 62, 3609-3614.

Addison, J.F.D., McGarry, K.J., Wermter, S. and MacIntyre, J. (2004) Stepwise linear regression for dimensionality reduction in network modelling. In Hamza, M.H. (ed), *Artificial Intelligence and Applications*. Innsbruck, Austria, 910.

Adib, T.R., Henderson, S., Perrett, C., Hewitt, D., Bourmpoulia, D., Ledermann, J. and Boshoff, C. (2004) Predicting biomarkers for ovarian cancer using gene-expression microarrays, *Br J Cancer*, 90, 686-692.

Aebersold, R. and Mann, M. (2003) Mass spectrometry-based proteomics, *Nature*, 422, 198-207.

Alexander, H., Stegner, A.L., Wagner-Mann, C., Du Bois, G.C., Alexander, S. and Sauter, E.R. (2004) Proteomic analysis to identify breast cancer biomarkers in nipple aspirate fluid, *Clin Cancer Res*, 10, 7500-7510.

Allison, J.S., Heo, J. and Iskandrian, A.E. (2005) Artificial neural network modeling of stress single-photon emission computed tomographic imaging for detecting extensive coronary artery disease, *Am J Cardiol*, 95, 178-181.

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

Ando, T., Suguro, M., Kobayashi, T., Seto, M. and Honda, H. (2003) Multiple fuzzy neural network system for outcome prediction and classification of 220 lymphoma patients on the basis of molecular profiling, *Cancer Sci*, 94, 906-913.

Andrighetto, C., De Dea, P., Lombardi, A., Neviani, E., Rossetti, L. and Giraffa, G. (1998) Molecular identification and cluster analysis of homofermentative thermophilic lactobacilli isolated from dairy products, *Res Microbiol*, 149, 631-643.

Attoh-Okine, N.O. (1999) Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance, *Advances in Engineering Software*, 30, 291-302.

Azmy, I.A., Balasubramanian, S.P., Wilson, A.G., Stephenson, T.J., Cox, A., Brown, N.J. and Reed, M.W. (2004) Role of tumour necrosis factor gene polymorphisms (-308 and -238) in breast cancer susceptibility and severity, *Breast Cancer Res*, 6, R395-400.

Baggerly, K.A., Morris, J.S. and Coombes, K.R. (2004) Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments, *Bioinformatics*, 20, 777-785.

Ball, G., Mian, S., Holding, F., Allibone, R.O., Lowe, J., Ali, S., Li, G., McCardle, S., Ellis, I.O., Creaser, C. and Rees, R.C. (2002) An integrated approach utilizing artificial neural networks and SELDI mass spectrometry for the classification of human tumours and rapid identification of potential biomarkers, *Bioinformatics*, 18, 395-404.

Bar-Joseph, Z. (2004) Analyzing time series gene expression data, *Bioinformatics*, 20, 2493-2503.

Barlow, T.W. (1995) Self-organizing maps and molecular similarity, *J Mol Graph*, 13, 24-27, 53-25.

Barrier, A., Lemoine, A., Boelle, P.Y., Tse, C., Brault, D., Chiappini, F., Breittschneider, J., Lacaine, F., Houry, S., Huguier, M., Van der Laan, M.J., Speed, T., Debuire, B., Flahault, A. and Dudoit, S. (2005) Colon cancer prognosis prediction by gene expression profiling, *Oncogene*, 24, 6155-6164.

Basheer, I.A. and Hajmeer, M. (2000) Artificial neural networks: fundamentals, computing, design, and application, *J Microbiol Methods*, 43, 3-31.

Beer, D.G., Kardia, S.L., Huang, C.C., Giordano, T.J., Levin, A.M., Misek, D.E., Lin, L., Chen, G., Gharib, T.G., Thomas, D.G., Lizyness, M.L., Kuick, R., Hayasaka, S., Taylor, J.M., Iannettoni, M.D., Orringer, M.B. and Hanash, S. (2002) Gene-expression profiles predict survival of patients with lung adenocarcinoma, *Nat Med*, 8, 816-824.

Ben-Baruch, A. (2003) Host microenvironment in breast cancer development: inflammatory cells, cytokines and chemokines in breast cancer progression: reciprocal tumor-microenvironment interactions, *Breast Cancer Res*, 5, 31-36.

Berrar, D.P., Downes, C.S. and Dubitzky, W. (2003) Multiclass cancer classification using gene expression profiling and probabilistic neural networks, *Pac Symp Biocomput*, 5-16.

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

Bicciato, S., Pandin, M., Didone, G. and Di Bello, C. (2003) Pattern identification and classification in gene expression data using an autoassociative neural network model, *Biotechnol Bioeng*, 81, 594-606.

Bieche, I., Girault, I., Urbain, E., Tozlu, S. and Lidereau, R. (2004) Relationship between intratumoral expression of genes coding for xenobiotic-metabolizing enzymes and benefit from adjuvant tamoxifen in estrogen receptor alpha-positive postmenopausal breast carcinoma, *Breast Cancer Res*, 6, R252-263.

Bishop, C. (1995) *Neural networks for pattern recognition*. Oxford University Press.

Bleasby, A.J., Knowles, J.C. and Cooke, N.J. (1985) Microheterogeneity of alpha 1-acid glycoprotein: lack of discrimination between benign and malignant inflammatory disease of the lung, *Clin Chim Acta*, 150, 231-235.

Boon, M.E., Kok, L.P., Nygaard-Nielsen, M., Holm, K. and Holund, B. (1994) Neural network processing of cervical smears can lead to a decrease in diagnostic variability and an increase in screening efficacy: a study of 63 false-negative smears, *Mod Pathol*, 7, 957-961.

Bourquin, J., Schmidli, H., van Hoogevest, P. and Leuenberger, H. (1998) Advantages of Artificial Neural Networks (ANNs) as alternative modelling technique for data sets showing non-linear relationships using data from a galenical study on a solid dosage form, *Eur J Pharm Sci*, 7, 5-16.

Braga-Neto, U. and Dougherty, E. (2005) Exact performance of error estimators for discrete classifiers., *Pattern Recognition*, 38, 1799-1814.

Brown, P.O. and Botstein, D. (1999) Exploring the new world of the genome with DNA microarrays, *Nat Genet*, 21, 33-37.

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

Burnside, E.S. (2005) Bayesian networks: computer-assisted diagnosis support in radiology, *Acad Radiol*, 12, 422-430.

Calin, G., Herlea, V., Barbanti-Brodano, G. and Negrini, M. (1998) The coding region of the Bloom syndrome BLM gene and of the CBL proto-oncogene is mutated in genetically unstable sporadic gastrointestinal tumors, *Cancer Res*, 58, 3777-3781.

Cao, Y. and Wu, J. (2002) Projective ART for clustering data sets in high dimensional spaces, *Neural Netw*, 15, 105-120.

Carpenter, G.A. and Grossberg, S. (1988) The ART of adaptive pattern recognition by a self-organising neural network, *IEEE Computer*, 21, 77-88.

Carsberg, C.J., Myers, K.A., Evans, G.S., Allen, T.D. and Stern, P.L. (1995) Metastasis-associated 5T4 oncofoetal antigen is concentrated at microvillus projections of the plasma membrane, *J Cell Sci*, 108 (Pt 8), 2905-2916.

Chen, Y.D., Zheng, S., Yu, J.K. and Hu, X. (2004) [Application of serum protein pattern model in diagnosis of colorectal cancer], *Zhonghua Zhong Liu Za Zhi*, 26, 417-420.

Chernushevich, I.V., Loboda, A.V. and Thomson, B.A. (2001) An introduction to quadrupole-time-of-flight mass spectrometry, *J Mass Spectrom*, 36, 849-865.

Chia, S.K., Wykoff, C.C., Watson, P.H., Han, C., Leek, R.D., Pastorek, J., Gatter, K.C., Ratcliffe, P. and Harris, A.L. (2001) Prognostic significance of a novel hypoxia-regulated marker, carbonic anhydrase IX, in invasive breast carcinoma, *J Clin Oncol*, 19, 3660-3668.

Chu, F. and Wang, L. (2005) Applications of support vector machines to cancer classification with microarray data, *Int J Neural Syst*, 15, 475-484.

Colantonio, D.A. and Chan, D.W. (2005) The clinical application of proteomics, *Clin Chim Acta*, 357, 151-158.

Cottrell, M., Girard, B., Girard, Y., Mangeas, M. and Muller, C. (1995) Neural Modeling for Time Series: A Statistical Stepwise Method for Weight Elimination, *IEEE Transactions on neural networks*, 6, 1355-1364.

Cowper, M.R., Mulgrew, B. and Unsworth, C.P. (2002) Nonlinear prediction of chaotic signals using a normalised radial basis function network, *Signal Processing*, 82, 775-789.

Cox, B., Kislinger, T. and Emili, A. (2005) Integrating gene and protein expression data: pattern analysis and profile mining, *Methods*, 35, 303-314.

Craven, D.E., Frasch, C.E., Robbins, J.B. and Feldman, H.A. (1978) Serogroup identification of Neisseria meningitidis: comparison of an antiserum agar method with bacterial slide agglutination, *J Clin Microbiol*, 7, 410-414.

Crisianini, N. and Shawe-Taylor, J. (2000) *An introduction to support vector machines (and other kernel-based learning methods)*
Cambridge University Press.

Cytoscape (http://www.cytoscape.org)

D'Amato, R.F., Eriquez, L.A., Tomfohrde, K.M. and Singerman, E. (1978) Rapid identification of Neisseria gonorrhoeae and Neisseria meningitidis by using enzymatic profiles, *J Clin Microbiol*, 7, 77-81.

Dayhoff, J. (1990) *Neural Network Architectures An Introduction*. VNR Press.

de Jong, H. (2002) Modeling and simulation of genetic regulatory systems: a literature review, *J Comput Biol*, 9, 67-103.

de Launoit, Y. and Adamski, J. (1999) Unique multifunctional HSD17B4 gene product: 17beta-hydroxysteroid dehydrogenase 4 and D-3-hydroxyacyl-coenzyme A dehydrogenase/hydratase involved in Zellweger syndrome, *J Mol Endocrinol*, 22, 227-240.

de Tommaso, M., De Carlo, F., Difruscolo, C., Massafra, R., Sciruicchio, V. and Belloti, R. (2003) Detection of subclinical brain electrical activity changes in Huntington's disease using artificial neural networks, *Clinical Neurophysiology*, 113, 1237-1245.

Debouck, C. and Goodfellow, P.N. (1999) DNA microarrays in drug discovery and development, *Nat Genet*, 21, 48-50.

Deichmann, M., Benner, A., Bock, M., Jackel, A., Uhl, K., Waldmann, V. and Naher, H. (1999) S100-Beta, melanoma-inhibiting activity, and lactate dehydrogenase discriminate progressive from nonprogressive American Joint Committee on Cancer stage IV melanoma, *J Clin Oncol*, 17, 1891-1896.

Demicco, E.G., Kavanagh, K.T., Romieu-Mourez, R., Wang, X., Shin, S.R., Landesman-Bollag, E., Seldin, D.C. and Sonenshein, G.E. (2005) RelB/p52 NF-kappaB complexes rescue an early delay in mammary gland development in transgenic mice with targeted superrepressor IkappaB-alpha expression and promote carcinogenesis of the mammary gland, *Mol Cell Biol*, 25, 10136-10147.

Dhingra, V., Gupta, M., Andacht, T. and Fu, Z.F. (2005) New frontiers in proteomics research: a perspective, *Int J Pharm*, 299, 1-18.

Diamandis, E.P. and van der Merwe, D.E. (2005) Plasma protein profiling by mass spectrometry for cancer diagnosis: opportunities and limitations, *Clin Cancer Res*, 11, 963-965.

Djavan, B., Zlotta, A., Kratzik, C., Remzi, M., Seitz, C., Schulman, C.C. and Marberger, M. (1999) PSA, PSA density, PSA density of transition zone, free/total PSA ratio, and PSA velocity for early detection of prostate cancer in men with serum PSA 2.5 to 4.0 ng/mL, *Urology*, 54, 517-522.

Djukanovic, D., Hofmann, U., Sucker, A., Rittgen, W. and Schadendorf, D. (2000) Comparison of S100 protein and MIA protein as serum marker for malignant melanoma, *Anticancer Res*, 20, 2203-2207.

Domeniconi, C., Papadopoulos, D., Gunopulos, D. and Ma, S. (2004) Subspace clustering of high dimensional data In Berry, M.W., Dayal, U., Kamath, C. and Skillicorn, D. (eds), *4th SIAM International Conference on Data Mining*. SIAM, 3600 UNIV CITY SCIENCE CENTER, PHILADELPHIA, PA 19104-2688 USA Lake Buena Vista, FL, 517-521.

Dreiseitl, S., Ohno-Machado, L., Kittler, H., Vinterbo, S., Billhardt, H. and Binder, M. (2001) A comparison of machine learning methods for the diagnosis of pigmented skin lesions, *J Biomed Inform*, 34, 28-36.

Duche, J.C., Urien, S., Simon, N., Malaurie, E., Monnet, I. and Barre, J. (2000) Expression of the genetic variants of human alpha-1-acid glycoprotein in cancer, *Clin Biochem*, 33, 197-202.

Duggan, D.J., Bittner, M., Chen, Y., Meltzer, P. and Trent, J.M. (1999) Expression profiling using cDNA microarrays, *Nat Genet*, 21, 10-14.

Duh, M.S., Walker, A.M. and Ayanian, J.Z. (1998) Epidemiologic interpretation of artificial neural networks, *Am J Epidemiol*, 147, 1112-1122.

Eftekhar, B., Mohammad, K., Ardebili, H.E., Ghodsi, M. and Ketabchi, E. (2005) Comparison of artificial neural network and logistic regression models for prediction of mortality in head trauma based on initial clinical data, *BMC Med Inform Decis Mak*, 5, 3.

Eszlinger, M., Wiench, M., Jarzab, B., Krohn, K., Beck, M., Lauter, J., Gubala, E., Fujarewicz, K., Swierniak, A. and Paschke, R. (2006) Meta- and reanalysis of gene expression profiles of hot and cold thyroid nodules and papillary thyroid carcinoma for gene groups, *J Clin Endocrinol Metab*.

Ferrari, L., Seraglia, R., Rossi, C.R., Bertazzo, A., Lise, M., Allegri, G. and Traldi, P. (2000) Protein profiles in sera of patients with malignant cutaneous melanoma, *Rapid Commun Mass Spectrom*, 14, 1149-1154.

Fidler, I.J. (2003) The pathogenesis of cancer metastasis: the 'seed and soil' hypothesis revisited, *Nat Rev Cancer*, 3, 453-458.

Fletcher, D. and Goss, E. (1993) Forecasting With Neural Networks: An Application Using Bankruptcy Data, *Information and Management*, 24, 159-167.

Franzke, A., Probst-Kepper, M., Buer, J., Duensing, S., Hoffmann, R., Wittke, F., Volkenandt, M., Ganser, A. and Atzpodien, J. (1998) Elevated pretreatment serum levels of soluble vascular cell adhesion molecule 1 and lactate dehydrogenase as predictors of survival in cutaneous metastatic malignant melanoma, *Br J Cancer*, 78, 40-45.

Fujiki, H., Suganuma, M., Komori, A., Yatsunami, J., Okabe, S., Ohta, T. and Sueoka, E. (1994) A new tumor promotion pathway and its inhibitors, *Cancer Detect Prev*, 18, 1-7.

Gao, W.M., Kuick, R., Orchekowski, R.P., Misek, D.E., Qiu, J., Greenberg, A.K., Rom, W.N., Brenner, D.E., Omenn, G.S., Haab, B.B. and Hanash, S.M. (2005) Distinctive serum protein profiles involving abundant proteins in lung cancer patients based upon antibody microarray analysis, *BMC Cancer*, 5, 110.

Garcia-Closas, M., Brinton, L.A., Lissowska, J., Chatterjee, N., Peplonska, B., Anderson, W.F., Szeszenia-Dabrowska, N., Bardin-Mikolajczak, A., Zatonski, W., Blair, A., Kalaylioglu, Z., Rymkiewicz, G., Mazepa-Sikora, D., Kordek, R., Lukaszek, S. and Sherman, M.E. (2006) Established breast cancer risk factors by clinically important tumour characteristics, *Br J Cancer*, 95, 123-129.

Garson, G.D. (1991) Interpreting neural-network connection weights, *Artificial Intelligence Expert*, 6, 47-51.

Geman, S., Bienenstock, E. and Doursat, R. (1992) Neural Networks and the Bias/Variance Dilemma, *Neural Computation*, 4, 1-58.

German, J. (1997) Bloom's syndrome. XX. The first 100 cancers, *Cancer Genet Cytogenet*, 93, 100-106.

Gery, S. and Koeffler, H.P. (2003) Repression of the TMEFF2 promoter by c-Myc, *J Mol Biol*, 328, 977-983.

Gery, S., Sawyers, C.L., Agus, D.B., Said, J.W. and Koeffler, H.P. (2002) TMEFF2 is an androgen-regulated gene exhibiting antiproliferative effects in prostate cancer cells, *Oncogene*, 21, 4739-4746.

Goodacre, R., Timmins, E.M., Burton, R., Kaderbhai, N., Woodward, A.M., Kell, D.B. and Rooney, P.J. (1998) Rapid identification of urinary tract infection bacteria using hyperspectral whole-organism fingerprinting and artificial neural networks, *Microbiology*, 144 (Pt 5), 1157-1170.

Goodenough, D.J., Rossmann, K. and Lusted, L.B. (1974) Radiographic applications of receiver operating characteristic (ROC) curves, *Radiology*, 110, 89-95.

Greene, D. and Cunningham, P. (2005) Producing accurate interpretable clusters from high-dimensional data In Jorge, A., Torgo, L., Brazdil, P., Camacho, R. and Gama, J. (eds), *9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*. Porto, Portugal.

Guffanti, A. (2002) Modeling molecular networks: a systems biology approach to gene function, *Genome Biol*, 3, reports4031.

Guo, X., Cao, S.M., Yu, J.K., Hong, M.H., Zhang, C.Q., Li, N.W., Xiang, Y.Q., Qian, C.N., Hu, X., Min, H.Q. and Zeng, Y.X. (2005) Distinct serumal proteomic patterns between ascending and descending types of loco-regionally advanced nasopharyngeal carcinoma assessed by surface enhanced laser desorption ionization and artificial neural network analyses, *Chin Med J (Engl)*, 118, 1912-1917.

Haykin, S. (1999) *Neural networks: a comprehensive foundation*. Prentice-Hall. Inc.

Hilario, M., Kalousis, A., Prados, J. and Binz, P.-A. (2004) Data mining for mass spectra-based cancer diagnosis and biomarker discovery, *Drug Discovery Today: Biosilico*, 2, 214-222.

Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities, *Proc Natl Acad Sci U S A*, 79, 2554-2558.

Hu, Y., Zhang, S., Yu, J., Liu, J. and Zheng, S. (2005) SELDI-TOF MS: the proteomics and bioinformatics approaches in the diagnosis of breast cancer, *Breast*, 14, 250-255.

Hu, Y., Zhang, S.Z., Yu, J.K., Liu, J., Zheng, S. and Hu, X. (2005) [Diagnostic application of serum protein pattern and artificial neural network software in breast cancer.], *Ai Zheng*, 24, 67-71.

Huang, C.-F., Tiee, M. and Forst, C.V. (2005) Pathway optimization of biological drug response networks *2nd IEEE Symposium on Computational Intelligence in Bioformatics and Computational Biology*. IEEE, 345 E 47TH ST, NEW YORK, NY 10017 USA La Jolla, CA, 82-89.

Huang, E., Cheng, S.H., Dressman, H., Pittman, J., Tsou, M.H., Horng, C.F., Bild, A., Iversen, E.S., Liao, M., Chen, C.M., West, M., Nevins, J.R. and Huang, A.T. (2003) Gene expression predictors of breast cancer outcomes, *Lancet*, 361, 1590-1596.

Hunter, A., Kennedy, L., Henry, J. and Ferguson, I. (2000) Application of neural networks and sensitivity analysis to improved prediction of trauma survival, *Comput Methods Programs Biomed*, 62, 11-19.

Ioannidis, J.P. (2005) Microarrays and molecular research: noise discovery?, *Lancet*, 365, 454-455.

Jain, A.K., Mao, J. and Mohiuddin, K. (1996) Artificial Neural Networks: A Tutorial, *IEEE Computer*, 29, 56-63.

Janda, W.M., Morello, J.A. and Bohnhoff, M. (1984) Use of the API NeIdent system for identification of pathogenic Neisseria spp. and Branhamella catarrhalis, *J Clin Microbiol*, 19, 338-341.

Jeffrey, S.S., Lonning, P.E. and Hillner, B.E. (2005) Genomics-based prognosis and therapeutic prediction in breast cancer, *J Natl Compr Canc Netw*, 3, 291-300.

Jerez-Aragones, J.M., Gomez-Ruiz, J.A., Ramos-Jimenez, G., Munoz-Perez, J. and Alba-Conejo, E. (2003) A combined neural network and decision trees model for prognosis of breast cancer relapse, *Artif Intell Med*, 27, 45-63.

Jiang, Y., Harlocker, S.L., Molesh, D.A., Dillon, D.C., Stolk, J.A., Houghton, R.L., Repasky, E.A., Badaro, R., Reed, S.G. and Xu, J. (2002) Discovery of differentially expressed genes in human breast cancer using subtracted cDNA libraries and cDNA microarrays, *Oncogene*, 21, 2270-2282.

Kaneko, H., Inoue, R., Yamada, Y., Sukegawa, K., Fukao, T., Tashita, H., Teramoto, T., Kasahara, K., Takami, T. and Kondo, N. (1996) Microsatellite instability in B-cell lymphoma originating from Bloom syndrome, *Int J Cancer*, 69, 480-483.

Kang-Ning, L. and Perez, R.A. (1996) A new system identification technique using Kalman filtering and multilayer neural networks, *Artificial Intelligence in Engineering*, 10, 1-8.

Kennedy, S. (2002) The role of proteomics in toxicology: identification of biomarkers of toxicity by protein expression analysis, *Biomarkers*, 7, 269-290.

Khan, J., Wei, J.S., Ringner, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C. and Meltzer, P.S. (2001) Classification and

diagnostic prediction of cancers using gene expression profiling and artificial neural networks, *Nat Med*, 7, 673-679.

Kim, B. and Park, S. (2001) An optimal neural network plasma model: a case study, *Chemometrics and Intelligent Laboratory Systems*, 56, 39-50.

Kohonen, T. (1989) *Self-organization and associative memory*. Springer, Berlin.

Komori, A., Yatsunami, J., Suganuma, M., Okabe, S., Abe, S., Sakai, A., Sasaki, K. and Fujiki, H. (1993) Tumor necrosis factor acts as a tumor promoter in BALB/3T3 cell transformation, *Cancer Res*, 53, 1982-1985.

Koomen, J.M., Li, D., Xiao, L.C., Liu, T.C., Coombes, K.R., Abbruzzese, J. and Kobayashi, R. (2005) Direct tandem mass spectrometry reveals limitations in protein profiling experiments for plasma biomarker discovery, *J Proteome Res*, 4, 972-981.

Kremmer, T., Szollosi, E., Boldizsar, M., Vincze, B., Ludanyi, K., Imre, T., Schlosser, G. and Vekey, K. (2004) Liquid chromatographic and mass spectrometric analysis of human serum acid alpha-1-glycoprotein, *Biomed Chromatogr*, 18, 323-329.

Kuo, R.J. (2001) A sales forecasting system based on fuzzy neural network with initial weights generated by genetic algorithm, *European Journal of Operational Research*, 129, 496-517.

Lacroix, M. and Leclercq, G. (2004) About GATA3, HNF3A, and XBP1, three genes co-expressed with the oestrogen receptor-alpha gene (ESR1) in breast cancer, *Mol Cell Endocrinol*, 219, 1-7.

Lacroix, M., Toillon, R.A. and Leclercq, G. (2004) Stable 'portrait' of breast tumors during progression: data from biology, pathology and genetics, *Endocr Relat Cancer*, 11, 497-522.

Lancashire, L., Schmid, O., Shah, H. and Ball, G. (2005) Classification of bacterial species from proteomic data using combinatorial approaches incorporating artificial neural networks, cluster analysis and principal components analysis, *Bioinformatics*, 21, 2191-2199.

Lee, J.S., Chu, I.S., Mikaelyan, A., Calvisi, D.F., Heo, J., Reddy, J.K. and Thorgeirsson, S.S. (2004) Application of comparative functional genomics to identify best-fit mouse models to study human cancer, *Nat Genet*, 36, 1306-1311.

Li, J., Zhang, Z., Rosenzweig, J., Wang, Y.Y. and Chan, D.W. (2002) Proteomics and bioinformatics approaches for identification of serum biomarkers to detect breast cancer, *Clin Chem*, 48, 1296-1304.

Li, L., Tang, H., Wu, Z., Gong, J., Gruidl, M., Zou, J., Tockman, M. and Clark, R.A. (2004) Data mining techniques for cancer detection using serum proteomic profiling, *Artif Intell Med*, 32, 71-83.

Liao, S.Y., Brewer, C., Zavada, J., Pastorek, J., Pastorekova, S., Manetta, A., Berman, M.L., DiSaia, P.J. and Stanbridge, E.J. (1994) Identification of the MN antigen as a diagnostic biomarker of cervical intraepithelial squamous and glandular neoplasia and cervical carcinomas, *Am J Pathol*, 145, 598-609.

Liebler, D.C. (2002) *Introduction to Proteomics: Tools for the New Biology*. Humana Press Inc.

Lisboa, P.J. and Taktak, A.F. (2006) The use of artificial neural networks in decision support in cancer: A systematic review, *Neural Netw*.

Liu, Z., Chen, D. and Bensmail, H. (2005) Gene expression data classification with Kernel principal component analysis, *J Biomed Biotechnol*, 2005, 155-159.

Lusted, L.B. (1971) Decision-making studies in patient management, *N Engl J Med*, 284, 416-424.

Maier, H.R. and Dandy, G.C. (1998) The effect of internal parameters and geometry on the performance of back-propagation neural networks: an empirical study, *Environmental Modelling and Software*, 13, 193-209.

Manel, S., Dias, J.-M. and Ormerod, S.J. (1999) Comparing discriminant analysis, neural networks and logistic regression for predicting species distributions: a case study with a Himalayan river bird *Ecological Modelling*, 120, 337-347.

Marengo, E., Robotti, E., Righetti, P.G., Campostrini, N., Pascali, J., Ponzoni, M., Hamdan, M. and Astner, H. (2004) Study of proteomic changes associated with healthy

and tumoral murine samples in neuroblastoma by principal component analysis and classification methods, *Clin Chim Acta*, 345, 55-67.

Martin, M. (2006) Molecular biology of breast cancer, *Clin Transl Oncol*, 8, 7-14.

Masters, J.R. and Lakhani, S.R. (2000) How diagnosis with microarrays can help cancer patients, *Nature*, 404, 921.

McCulloch, W.S. and Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 7, 115-133.

Meiners, S., Brinkmann, V., Naundorf, H. and Birchmeier, W. (1998) Role of morphogenetic factors in metastasis of mammary carcinoma cells, *Oncogene*, 16, 9-20.

Metz, C.E. (1978) Basic principles of ROC analysis, *Semin Nucl Med*, 8, 283-298.

Mian, S., Ball, G., Hornbuckle, J., Holding, F., Carmichael, J., Ellis, I., Ali, S., Li, G., McArdle, S., Creaser, C. and Rees, R. (2003) A prototype methodology combining surface-enhanced laser desorption/ionization protein chip technology and artificial neural network algorithms to predict the chemoresponsiveness of breast cancer cell lines exposed to Paclitaxel and Doxorubicin under in vitro conditions, *Proteomics*, 3, 1725-1737.

Mian, S., Ugurel, S., Parkinson, E., Schlenzka, I., Dryden, I., Lancashire, L., Ball, G., Creaser, C., Rees, R. and Schadendorf, D. (2005) Serum proteomic fingerprinting discriminates between clinical stages and predicts disease progression in melanoma patients, *J Clin Oncol*, 23, 5088-5093.

Michiels, S., Koscielny, S. and Hill, C. (2005) Prediction of cancer outcome with microarrays: a multiple random validation strategy, *Lancet*, 365, 488-492.

Minsky, M. and Papert, S. (1969) *Perceptrons*. MA: MIT Press, Cambridge.

Mitchell, T.M. (1997) *Machine Learning*. McGraw-Hill.

Mittal, G.S. and Zhang, J. (2000) Prediction of freezing time for food products using neural network, *Food Research Int*, 33, 557-562.

Mittal, G.S. and Zhang, J. (2000) Prediction of temperature and moisture content of frankfurters during thermal processing using neural network, *Meat Sci*
55, 13-24.

Moody, J. and Darkin, C. (1989) Fast learning in networks of lacally-tuned processing units, *Neural Computation*, 1, 281-294.

Murphy, N., Millar, E. and Lee, C.S. (2005) Gene expression profiling in breast cancer: towards individualising patient management, *Pathology*, 37, 271-277.

Mylonas, P., Wallace, M. and Kollias, S. (2004) Using k-nearest neighbor and feature selection as an improvement to hierarchical clustering. *3rd Hellenic Conference on Artificial Intelligence*. SPRINGER-VERLAG BERLIN, HEIDELBERGER PLATZ 3, D-14197 BERLIN, GERMANY, Samos, Greece.

Nakashima, J., Tachibana, M., Ueno, M., Miyajima, A., Baba, S. and Murai, M. (1998) Association between tumor necrosis factor in serum and cachexia in patients with prostate cancer, *Clin Cancer Res*, 4, 1743-1748.

Narayanan, A., Keedwell, E.C., Gamalielsson, J. and Tatineni, S. (2004) Single-layer artificial neural networks for gene expression analysis, *Neurocomputing*, 61, 217-240.

Nomura, T. and Katunuma, N. (2005) Involvement of cathepsins in the invasion, metastasis and proliferation of cancer cells, *J Med Invest*, 52, 1-9.

Nour, M.A. and Madey, G.R. (1996) Heuristic and optimization approaches to extending the Kohonenself organizing algorithm, *Eur J Oper Res*, 93, 428-448.

O'Neill, M.C. and Song, L. (2003) Neural network analysis of lymphoma microarray data: prognosis and diagnosis near-perfect, *BMC Bioinformatics*, 4, 13.

Olden, J.D., Joy, M.K. and Death, R.G. (2004) An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data, *Ecological Modelling*, 178, 389-397.

Osuna, E. and Girosi, F. (1999) Reducing run-time complexity in support vector machines. In Schölkopf, B., Burges, C.J.C. and Smola, A.J. (eds), *Advances in Kernel Methods: Support Vector Learning*. The MIT Press 392.

Palmer-Brown, D., Wee Lee, S., Tepper, J. and Roadknight, C. (2003) Fast learning neural nets with adaptive learning styles. In Al-Dabass, D. (ed), *Foundations for successful modelling and simulation. 17th European simulation multiconference*. Nottingham, UK.

Paweletz, C.P., Gillespie, J.W., Ornstein, D.K., Simone, N.L., Brown, M.R., Cole, K.A., Wang, Q.-H., Huang, J., Hu, N., Yip, T.-T., Rich, W.E., Kohn, E.C., Linehan, W.M., Weber, T., Taylor, P., Emmert-Buck, M.R., Liotta, L.A. and Petricoin, E.F. (2000) Rapid Protein Display Profiling of Cancer Progression Directly from Human Tissue Using a Protein Biochip, *Drug Development Research*, 49, 34-42.

Petricoin, E.F., Ardekani, A.M., Hitt, B.A., Levine, P.J., Fusaro, V.A., Steinberg, S.M., Mills, G.B., Simone, C., Fishman, D.A., Kohn, E.C. and Liotta, L.A. (2002) Use of proteomic patterns in serum to identify ovarian cancer, *Lancet*, 359, 572-577.

Petricoin, E.F. and Liotta, L.A. (2002) Proteomic analysis at the bedside: early detection of cancer, *Trends Biotechnol*, 20, S30-34.

Petricoin, E.F., Zoon, K.C., Kohn, E.C., Barrett, J.C. and Liotta, L.A. (2002) Clinical proteomics: translating benchside promise into bedside reality, *Nat Rev Drug Discov*, 1, 683-695.

Pomeroy, S.L., Tamayo, P., Gaasenbeek, M., Sturla, L.M., Angelo, M., McLaughlin, M.E., Kim, J.Y., Goumnerova, L.C., Black, P.M., Lau, C., Allen, J.C., Zagzag, D., Olson, J.M., Curran, T., Wetmore, C., Biegel, J.A., Poggio, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, G., Louis, D.N., Mesirov, J.P., Lander, E.S. and Golub, T.R. (2002) Prediction of central nervous system embryonal tumour outcome based on gene expression, *Nature*, 415, 436-442.

Porkka, K., Saramaki, O., Tanner, M. and Visakorpi, T. (2002) Amplification and overexpression of Elongin C gene discovered in prostate cancer by cDNA microarrays, *Lab Invest*, 82, 629-637.

Pritchard, L. and Dufton, M.J. (2000) Do proteins learn to evolve? The Hopfield network as a basis for the understanding of protein evolution *J Theor Biol*, 202, 77-86.

Ragazzi, E., Vogliardi, S., Allegri, G., Costa, C.V., Lise, M., Rossi, C.R., Seraglia, R. and Traldi, P. (2003) Cluster analysis of serum proteins in malignant cutaneous melanoma: search for disease markers, *Rapid Commun Mass Spectrom*, 17, 1511-1515.

Raimundo jr, I.M. and Narayanaswamy, R. (2003) Simultaneous determination of Zn(II), Cd(II) and Hg(II) in water
Sensors and Actuators B, 90, 189-197.

Rebuffo, C.A., Schmitt, J., Wenning, M., von Stetten, F. and Scherer, S. (2006) Reliable and rapid identification of Listeria monocytogenes and Listeria species by artificial neural network-based Fourier transform infrared spectroscopy, *Appl Environ Microbiol*, 72, 994-1000.

Reddick, A. (1975) A simple carbohydrate fermentation test for identification of the pathogenic Neisseria, *J Clin Microbiol*, 2, 72-73.

Roadknight, C., Palmer-Brown, D. and Al-Dabass, D. (2001) Simulation of correlation activity pruning methods to enhance transparency of ANNs. *UKSIM2001*. Emmanuel College, Cambridge, UK., 56-62.

Robinson, M.J. and Oberhofer, T.R. (1983) Identification of pathogenic Neisseria species with the RapID NH system, *J Clin Microbiol*, 17, 400-404.

Rohrer, J.W., Barsoum, A.L., Dyess, D.L., Tucker, J.A. and Coggin, J.H., Jr. (1999) Human breast carcinoma patients develop clonable oncofetal antigen-specific effector and regulatory T lymphocytes, *J Immunol*, 162, 6880-6892.

Rosenblatt, F. (1958) The perceptron: a probabilistic model for information storage and organization in the brain, *Psychol Rev*, 65, 386-408.

Rosenwald, A., Wright, G., Chan, W.C., Connors, J.M., Campo, E., Fisher, R.I., Gascoyne, R.D., Muller-Hermelink, H.K., Smeland, E.B., Giltnane, J.M., Hurt, E.M., Zhao, H., Averett, L., Yang, L., Wilson, W.H., Jaffe, E.S., Simon, R., Klausner, R.D., Powell, J., Duffey, P.L., Longo, D.L., Greiner, T.C., Weisenburger, D.D., Sanger, W.G., Dave, B.J., Lynch, J.C., Vose, J., Armitage, J.O., Montserrat, E., Lopez-Guillermo, A., Grogan, T.M., Miller, T.P., LeBlanc, M., Ott, G., Kvaloy, S., Delabie, J., Holte, H., Krajci, P., Stokke, T. and Staudt, L.M. (2002) The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma, *N Engl J Med*, 346, 1937-1947.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Learning representations by back-propagating errors, *Nature*, 323, 533-536.

Saarnio, J., Parkkila, S., Parkkila, A.K., Haukipuro, K., Pastorekova, S., Pastorek, J., Kairaluoma, M.I. and Karttunen, T.J. (1998) Immunohistochemical study of colorectal tumors for expression of a novel transmembrane carbonic anhydrase, MN/CA IX, with potential value as a marker of cell proliferation, *Am J Pathol*, 153, 279-285.

Sargent, D.J. (2001) Comparison of artificial neural networks with other statistical approaches: results from medical data sets, *Cancer*, 91, 1636-1642.

Sasaki, K., Sato, K., Akiyama, Y., Yanagihara, K., Oka, M. and Yamaguchi, K. (2002) Peptidomics-based approach reveals the secretion of the 29-residue COOH-terminal fragment of the putative tumor suppressor protein DMBT1 from pancreatic adenocarcinoma cell lines, *Cancer Res*, 62, 4894-4898.

Seibert, V., Wiesner, A., Buschmann, T. and Meuer, J. (2004) Surface-enhanced laser desorption ionization time-of-flight mass spectrometry (SELDI-TOF-MS) and ProteinChip technology in proteomics research, *Pathol Res Pract*, 200, 83-94.

Seraglia, R., Vogliardi, S., Allegri, G., Comai, S., Lise, M., Rossi, C.R., Mocellin, S., Scalerta, R., Ragazzi, E. and Traldi, P. (2005) Search for melanoma markers in plasma and serum samples, *Eur J Mass Spectrom (Chichester, Eng)*, 11, 353-360.

Shen, H.B., Yang, J. and Chou, K.C. (2005) Fuzzy KNN for predicting membrane protein types from pseudo-amino acid composition, *J Theor Biol*.

Smith, A.E., Nugent, C.D. and McClean, S.I. (2003) Evaluation of inherent performance of intelligent medical decision support systems: utilising neural networks as an example, *Artif Intell Med*, 27, 1-27.

Smith, K.A., Abramson, D. and Duke, D. (2003) Hopfield neural networks for timetabling: formulations, methods, and comparative results, *Computers and Industrial Engineering* 44, 283-305.

Song, X., Mitnitski, A., Cox, J. and Rockwood, K. (2004) Comparison of machine learning techniques with classical statistical models in predicting health outcomes, *Medinfo*, 11, 736-740.

Sorace, J.M. and Zhan, M. (2003) A data review and re-assessment of ovarian cancer serum proteomic profiling, *BMC Bioinformatics*, 4, 24.

Sotiriou, C., Neo, S.Y., McShane, L.M., Korn, E.L., Long, P.M., Jazaeri, A., Martiat, P., Fox, S.B., Harris, A.L. and Liu, E.T. (2003) Breast cancer classification and prognosis based on gene expression profiles from a population-based study, *Proc Natl Acad Sci U S A*, 100, 10393-10398.

Spieth, C., Streichert, F., Supper, J., Speer, N. and Zell, A. (2005) Feedback memetic algorithms for modeling gene regulatory networks *2nd IEEE Symposium on Computational Intelligence in Bioformatics and Computational Biology*. IEEE, 345 E 47TH ST, NEW YORK, NY 10017 USA La Jolla, CA, 61-67.

Srecnik, G., Debeljak, Z., Cerjan-Stefanovic, S., Novic, M. and Bolancab, T. (2002) Optimization of artificial neural networks used for retention modelling in ion chromatography, *J Chromatogr A*, 973, 47-59.

Sreerama, L. and Sladek, N.E. (1997) Cellular levels of class 1 and class 3 aldehyde dehydrogenases and certain other drug-metabolizing enzymes in human breast malignancies, *Clin Cancer Res*, 3, 1901-1914.

Starzynska, T., Rahi, V. and Stern, P.L. (1992) The expression of 5T4 antigen in colorectal and gastric carcinoma, *Br J Cancer*, 66, 867-869.

Stekel, D. (2003) *Microarray Bioinformatics*. Cambridge University Press.

Steyerberg, E.W., Eijkemans, M.J. and Habbema, J.D. (1999) Stepwise selection in small data sets: a simulation study of bias in logistic regression analysis, *J Clin Epidemiol*, 52, 935-942.

Subasi, A. and Ercelebi, E. (2005) Classification of EEG signals using neural network and logistic regression, *Comput Methods Programs Biomed*, 78, 87-99.

Suganuma, M., Okabe, S., Marino, M.W., Sakai, A., Sueoka, E. and Fujiki, H. (1999) Essential role of tumor necrosis factor alpha (TNF-alpha) in tumor promotion as revealed by TNF-alpha-deficient mice, *Cancer Res*, 59, 4516-4518.

Tibshirani, R. (1996) A comparison of some error estimates for neural network models, *Neural Computation*, 8, 152-163.

Tsai, M.S., Shamon-Taylor, L.A., Mehmi, I., Tang, C.K. and Lupu, R. (2003) Blockage of heregulin expression inhibits tumorigenicity and metastasis of breast cancer, *Oncogene*, 22, 761-768.

Tung, W.L., Quek, C. and Cheng, P. (2004) GenSo-EWS: a novel neural-fuzzy based early warning system for predicting bank failures, *Neural Netw*, 17, 567-587.

Ultsch, A. and Roske, F. (2002) Self-organizing feature maps predicting sea levels *Information Sciences*, 144, 91-125.

van 't Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., van der Kooy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R. and Friend, S.H. (2002) Gene expression profiling predicts clinical outcome of breast cancer, *Nature*, 415, 530-536.

Wagner, M., Naik, D.N., Pothen, A., Kasukurti, S., Devineni, R.R., Adam, B.L., Semmes, O.J. and Wright, G.L., Jr. (2004) Computational protein biomarker prediction: a case study for prostate cancer, *BMC Bioinformatics*, 5, 26.

Wall, R., Cunningham, P., Walsh, P. and Byrne, S. (2003) Explaining the output of ensembles in medical decision support on a case by case basis, *Artif Intell Med*, 28, 191-206.

Ward Systems Group, I. (1993) Neuroshell 2 Help. In Inc., W.S.G. (ed).

Warnat, P., Eils, R. and Brors, B. (2005) Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes, *BMC Bioinformatics*, 6, 265.

Wasinger, V.C., Cordwell, S.J., Cerpa-Poljak, A., Yan, J.X., Gooley, A.A., Wilkins, M.R., Duncan, M.W., Harris, R., Williams, K.L. and Humphery-Smith, I. (1995) Progress with gene-product mapping of the Mollicutes: Mycoplasma genitalium, *Electrophoresis*, 16, 1090-1094.

Watt, P.M. and Hickson, I.D. (1996) Failure to unwind causes cancer. Genome stability, *Curr Biol*, 6, 265-267.

Welsh, J.B., Zarrinkar, P.P., Sapinoso, L.M., Kern, S.G., Behling, C.A., Monk, B.J., Lockhart, D.J., Burger, R.A. and Hampton, G.M. (2001) Analysis of gene expression profiles in normal and neoplastic ovarian tissue samples identifies candidate molecular markers of epithelial ovarian cancer, *Proc Natl Acad Sci U S A*, 98, 1176-1181.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J.A., Jr., Marks, J.R. and Nevins, J.R. (2001) Predicting the clinical status of human breast cancer by using gene expression profiles, *Proc Natl Acad Sci U S A*, 98, 11462-11467.

White, C.N., Chan, D.W. and Zhang, Z. (2004) Bioinformatics strategies for proteomic profiling, *Clin Biochem*, 37, 636-641.

Widrow, B. and Hoff, M.E. (1960) Adaptive switching circuits, *IRE WESCON Convention Record*, 96-104.

Wysocki, V.H., Resing, K.A., Zhang, Q. and Cheng, G. (2005) Mass spectrometry of peptides and proteins, *Methods*, 35, 211-222.

Xiao, Z., Prieto, D., Conrads, T.P., Veenstra, T.D. and Issaq, H.J. (2005) Proteomic patterns: their potential for disease diagnosis, *Mol Cell Endocrinol*, 230, 95-106.

Xu, W.H., Chen, Y.D., Hu, Y., Yu, J.K., Wu, X.G., Jiang, T.J., Zheng, S. and Zhang, S.Z. (2006) Preoperatively molecular staging with CM10 ProteinChip and SELDI-TOF MS for colorectal cancer patients, *J Zhejiang Univ Sci B*, 7, 235-240.

Yang, J., Xu, G., Hong, Q., Liebich, H.M., Lutz, K., Schmulling, R.-M. and Wahl, H.G. (2004) Discrimination of Type 2 diabetic patients from healthy controls by using metabonomics method based on their serum fatty acid profiles, *Journal of chromatography B*, 813, 53-58.

Yu, J.K., Chen, Y.D. and Zheng, S. (2004) An integrated approach to the detection of colorectal cancer utilizing proteomics and bioinformatics, *World J Gastroenterol*, 10, 3127-3131.

Yu, J.S., Ongarello, S., Fiedler, R., Chen, X.W., Toffolo, G., Cobelli, C. and Trajanoski, Z. (2005) Ovarian cancer identification based on dimensionality reduction for high-throughput mass spectrometry data, *Bioinformatics*, 21, 2200-2209.

APPENDIX

```
'This function compares i to input list
' if i is already selected, go to next i

Function FindString(ListOfChars As String, Char As String) As Boolean

Dim LengthOfList As Integer
Dim ToCompare As String
Dim TempChar As String


ToCompare = ""

For i = 1 To Len(ListOfChars)
   TempChar = Mid(ListOfChars, i, 1)
   If TempChar <> "," Then
      ToCompare = ToCompare + TempChar
   Else
      If ToCompare = Char Then
         FindString = True
         Exit Function
      End If
      ToCompare = ""
   End If
Next i

FindString = False

End Function

Private Sub Check1_Click()
If Check1.Value = 1 Then
   lblterm2.Visible = True
```

```
    TxtTerm.Visible = True
End If

If Check1.Value = 0 Then
    lblterm2.Visible = False
    TxtTerm.Visible = False
End If


End Sub

'exit program
Private Sub CmdExit_Click()
End

End Sub


Private Sub Command1_Click()

'Create directories to save analysis files into
MkDir (TxtPath.Text)
MkDir (TxtPath.Text + "\Bootstrapping_Results\")
MkDir (TxtPath.Text + "\Predictions\")
MkDir (TxtPath.Text + "\Sensitivity_Analysis\")
MkDir (TxtPath.Text + "\Overall_Summary\")
MkDir (TxtPath.Text + "\Overall_Averages\")
MkDir (TxtPath.Text + "\Network_Models\")

Dim inputlist As String
Dim z As Integer
Dim Additions As Integer
Dim Bootstraps As Integer
Dim Improvement As Integer
```

```
Improvement = 0


Additions = CInt(txtAdditive.Text)

'This creates an inputlist which is empty to begin with

If z <= 1 Then
inputlist = "," & txtaddinputs.Text & ","
End If

For z = 1 To Additions

Application.Option.Output.MaxOutputDocs = 10000000

Dim AO As AnalysisOutput
Dim AWB As Workbook

'set amount of bootstraps to equal combobox

Bootstraps = CInt(txtBoot.Text)


'declare variables

Dim f As Integer            'columns for mean analysis
Dim g As Integer            'rows to copy for mean analysis, = no. bootstraps ie 20
Dim h As Integer            'row to copy mean values
Dim l As Integer            'row ID for mean analysis to copy to averages sheet
Dim m As Integer            'row ID to copy results averages to averages sheet
Dim n As Integer            'row ID to paste from means analysis to averages sheet
Dim p As Integer            'row ID to paste input ID
Dim q As Integer            'row ID to paste sensitivity
Dim r As Integer            'column "" ""
Dim s As Integer            'end row to paste sensitivity
```

```
Dim HiddenLayers As Integer
Dim Inputs As Integer
Dim i As Integer


g = 1
h = Bootstraps + 1
l = 1
m = 1 + Bootstraps
n = 1
p = 1
q = 1
r = 1
s = 1


'number of hidden layers in analysis
'will equal amount set in text box

HiddenLayers = txtHidden.Text

'INPUT SELECTION
'number of inputs is specified in text box
'number of additions is specified also

Inputs = txtInputs.Text
For i = 1 To Inputs

'Create spreadsheets for analysis

Dim Results As New Spreadsheet
Dim Averages As New Spreadsheet
Dim Sensitivity As New Spreadsheet
Dim Summary As New Spreadsheet
Dim character As String
```

```
Dim tempsens As New Spreadsheet
Dim TempSpread As New Spreadsheet

character = ","

Dim SourceData As Spreadsheet

'INSERT PATH OF SPREADSHEET TO ANALYSE

Set SourceData = Spreadsheets.Open(lblSelected.Caption)

'this code declares the number of samples to analyse as the number
'of cases in the spreadsheet file
'this code also specifies the % split of training/test and validation

Dim Ncases As Long
Dim Samples As Integer
Dim Training As Integer
Dim Validation As Integer
Dim test As Integer



Ncases = SourceData.NumberOfCases
Samples = Ncases
Training = Samples * (txtTraining.Text / 100)
Validation = Samples * (lblValidation_Percentage.Caption / 100) / 2
test = Validation
'this code refers back to original function
'in order to determine if an input is already
'predent in the input list to analyse

Dim InputLoop2 As Boolean
InputLoop2 = True
```

```
Do While InputLoop2
If i > CInt(Inputs) Then
   InputLoop2 = False
Else

                    'Searching string for inputs already being used
      If FindString(inputlist, CStr(i)) = False Then
      InputLoop2 = False
      Else
         i = i + 1
      End If

End If

Loop
'this code creates spreadsheet to use for analysis, to speed up loading times

Dim TotalInputs As Integer
Dim DataArray As Double
Dim inputlist3 As String
Dim tempInput As String

TotalInputs = 1

inputlist3 = ""
inputlist3 = i & inputlist      'inputlist to read from for pasting variables to be used


Dim OutputColumn As Integer
Dim CurrentColumn As Integer
Dim InputNumber As Integer
Dim ToCompare As String
Dim TempChar As String
Dim count As Integer
```

```
InputLoop3 = True
InputNumber = 0
OutputColumn = txtOutput.Text              'identifies column countaining output variable
CurrentColumn = 1

TempSpread.Activate
TempSpread.AddVariables "", TempSpread.NumberOfVariables, 500

If z = 1 Then
   If i = 1 Then
      TempSpread.AddCases 10, SourceData.NumberOfCases - 10
   End If
End If

ToCompare = ""

If inputlist = ",," Then

SourceData.Activate
      Set Cells = SourceData.CellsRange(1, i, SourceData.NumberOfCases, i)
      Cells.Activate
      SourceData.CopyWithHeaders

      'Paste the contents of the column into the new speadsheet
      TempSpread.Activate
      Set Cells = TempSpread.CellsRange(1, CInt(CurrentColumn),
SourceData.NumberOfCases, CInt(CurrentColumn))
      Cells.Select
      TempSpread.Paste
      TempSpread.AutoFitVariables
      TotalInputs = TotalInputs + 1

      CurrentColumn = CurrentColumn + 1

Else
```

```
'loop around until the end of inputlist
For count = 1 To Len(inputlist3)
   TempChar = Mid(inputlist3, count, 1)
   If TempChar <> "," Then
      ToCompare = ToCompare + TempChar
   Else
      'Found a comma, use this column
      'Copy the cells from the opened spreadsheet in this column
      SourceData.Activate
      Set Cells = SourceData.CellsRange(1, CInt(ToCompare),
SourceData.NumberOfCases, CInt(ToCompare))
      Cells.Activate
      SourceData.CopyWithHeaders

      'Paste the contents of the column into the new speadsheet
      TempSpread.Activate
      Set Cells = TempSpread.CellsRange(1, CInt(CurrentColumn),
SourceData.NumberOfCases, CInt(CurrentColumn))
      Cells.Select
      TempSpread.Paste
      TempSpread.AutoFitVariables
      TotalInputs = TotalInputs + 1

      CurrentColumn = CurrentColumn + 1   'Advance the column number of the new
speadsheet
      InputNumber = InputNumber + 1       'Advance the input number

      ToCompare = ""                      'reset the comparison
   End If
Next count

                                  'copy output column
End If
```

```
' Paste the output column from the old to new spreadsheet
SourceData.Activate
Set Cells = SourceData.CellsRange(1, OutputColumn, SourceData.NumberOfCases,
OutputColumn)
Cells.Activate
SourceData.CopyWithHeaders

TempSpread.Activate
TempSpread.Visible = True
Set Cells = TempSpread.CellsRange(1, CurrentColumn, 1, CurrentColumn)
Cells.Select
TempSpread.Paste
TempSpread.AutoFitVariables
SourceData.SelectCaseNamesOnly
SourceData.Copy
TempSpread.SelectCaseNamesOnly
TempSpread.Paste
TempSpread.DeleteVariables CurrentColumn + 1, TempSpread.NumberOfVariables
Set Cells = TempSpread.CellsRange(1, 1, 1, 1)
Cells.Select

'main analysis code

Dim ANNanalysis As Analysis
Set ANNanalysis = Analysis(scNeuralNetworks, TempSpread)

ANNanalysis.NeverShowDialog = True
ANNanalysis.Visible = False
ANNanalysis.AnalysisOption.ShowWarnings = False

If i <= CInt(Inputs) Then

'input selection
```

```
If CurrentColumn = 2 Then

With ANNanalysis.Dialog
   .VariablesAndTypes = "1 | 2"
   .Variables = "2 | 1"

   .TypeOfAnalysis = scSNNCustomNetworkDesigner
   .IssueMissingDataWarning = False
   .ApplyMemoryLimit = False

End With

Else

With ANNanalysis.Dialog
   .VariablesAndTypes = "1-" & (CurrentColumn - 1) & " | " & CurrentColumn
   .Variables = CurrentColumn & "  |  1-" & (CurrentColumn - 1)

   .TypeOfAnalysis = scSNNCustomNetworkDesigner
   .IssueMissingDataWarning = False
   .ApplyMemoryLimit = False

End With
End If

ANNanalysis.Run

With ANNanalysis.Dialog
   .MLPNetwork = True
   .MLPNoHiddenLayers = 1
   .MLPNoHiddenUnitsLayer1 = HiddenLayers
   .MLPErrFnClassificationSOS = True                    'classification
  ' .MLPRegressionOutputFnLogistic = True                'regression
  ' .MLPRegressionOutputFnLogisticRange = 0.9            'regression
   .RBFNormalize = False
```

```
    .RBFNoHiddenUnits = 9
    .RBFClassificationErrFnEntropy = True
    .PNNNormalize = False
    .PNNIncludeLossMatrix = False
    .GRNNNormalize = False
    .GRNNNoHiddenUnits = 16
    .SOFMNormalize = False
    .SOFMWidth = 7
    .SOFMHeight = 2
    .PCNNoPrincipalComponents = 1
    .CNNormalize = False
    .CNNoHiddenUnits = 9
    .MakeTimeSeriesNetwork = False
End With

ANNanalysis.Run

With ANNanalysis.Dialog
    .MLPTrainingPhaseOne = True
    .MLPTrainingPhaseOneAlgorithm = 0
    .MLPTrainingPhaseOneEpochs = 3000
    .MLPTrainingPhaseOneLearningRate = txtLearning.Text
    .MLPTrainingPhaseTwo = False
    .MLPTrainingReinitializeBeforeTraining = True
    .MLPTrainingUniformReinitialize = True
    .MLPTrainingMinimumMean = 0
    .MLPTrainingMaximumSD = 1
    .MLPTrainingTrackBestNetwork = True
    .MLPTrainingTrainingErrorStop = 0
    .MLPTrainingSelectionErrorStop = 0
    .MLPTrainingTrainingErrorImprove = 0
    .MLPTrainingSelectionErrorImprove = 0
    .MLPTrainingImproveWindow = 1000
    .MLPTrainingPruneInputsAndHiddenUnits = False
    .MLPTrainingPruneInputsWithLowSensitivity = False
```

```
    .MLPTrainingClassificationSpecified = True
    .MLPTrainingClassificationAcceptRatio = 0.5
    .MLPTrainingClassificationRejectRatio = 0.5
    '.MLPTrainingClassificationMinLoss = True                    'classification
    '.MLPTrainingClassificationMinLossValue = 1                  'classification
    ' .MLPTrainingAssignToHighestConfidence = True               'regression
    .MLPTrainingApplyWeightDecayPhaseOne = False
    .MLPTrainingUseInteractiveDisplay = False
    .MLPTrainingBP1AdjustLRAndMomentum = False
    .MLPTrainingBP1LRInitValue = 0.1
    .MLPTrainingBP1MomentumInitValue = txtMomentum.Text
    .MLPTrainingBP1ShuffleCaseOrder = True
    .MLPTrainingBP1AddGaussianNoise = False
    .MLPTrainingBP2AdjustLRAndMomentum = False
    .MLPTrainingBP2LRInitValue = 0.01
    .MLPTrainingBP2MomentumInitValue = 0.3
    .MLPTrainingBP2ShuffleCaseOrder = True
    .MLPTrainingBP2AddGaussianNoise = False
    .MLPTrainingDBD1InitialLR = 0.01
    .MLPTrainingDBD1LRIncrement = 0.01
    .MLPTrainingDBD1Decay = 0.8
    .MLPTrainingDBD1Smoothing = 0.5
    .MLPTrainingDBD1AddGaussianNoise = False
    .MLPTrainingDBD2InitialLR = 0.01
    .MLPTrainingDBD2LRIncrement = 0.01
    .MLPTrainingDBD2Decay = 0.8
    .MLPTrainingDBD2Smoothing = 0.5
    .MLPTrainingDBD2AddGaussianNoise = False
    .MLPTrainingQP1LR = 0.01
    .MLPTrainingQP1Acceleration = 2
    .MLPTrainingQP1AddGaussianNoise = False
    .MLPTrainingQP2LR = 0.01
    .MLPTrainingQP2Acceleration = 2
    .MLPTrainingQP2AddGaussianNoise = False
End With
```

```
Set optionsdialog = ANNanalysis.Dialog.GetUserCaseSelection
With optionsdialog
   .SelectSpecifiedCases = True
   .UserSpecifiedCases = "1-" & Samples
End With

ANNanalysis.Dialog.SetUserCaseSelection optionsdialog
Set optionsdialog = Nothing

Set optionsdialog = ANNanalysis.Dialog.GetSampling
With optionsdialog
 .SamplingMethodUseRandomSamplingAdvanced = True
   .NoSamplesValueAdvanced = Bootstraps
   .UseFromEnsembleAdvanced = False
   .TrainingSubsetSizeCrossValidation = 96
   .SelectionSubsetSizeCrossValidation = 48
   .TrainingSubsetSizeBoot = 112
   .TrainingSubsetSizeBootstrap = 56
   .ResampleSelectionOfSubsetsRandom = True
   .AssignmentOfSubsetsOnceRandom = True
   .TrainingSubsetSizeRandom = Training
   .SelectionSubsetSizeRandom = test
   .TestSubsetSizeRandom = Validation
   .CasewiseDeletionOfMD = True
End With


ANNanalysis.Dialog.SetSampling optionsdialog
Set optionsdialog = Nothing

ANNanalysis.Run

With ANNanalysis.Dialog
```

```
'this specifies whether the user has selected
'either training/test or validation for analysis

  If OpTrain.Value = True Then
     .UseTrainingSubset = True
     Else
    ' .UseTrainingSubset = False
' End If
  If OpTest.Value = True Then
     .UseTestSubset = True
     Else
    ' .UseTestSubset = False
' End If
  If OpVal.Value = True Then
     .UseSelectionSubset = True
    ' Else
    ' .UseSelectionSubset = False
  End If
  End If
  End If

  .GenerateSummaryStatistics = True
  .GenerateConfusionMatrix = True                'classification
  .ShowPredictions = True
  .ShowConfidenceLevels = False                  'classification
  .IncludeObservedInPredictionsSpreadsheet = True
  .IncludeIndependentsInPredictionsSpreadsheet = False
  .IncludeSubsetVariableInPredictionsSpreadsheet = False
  .IncludeUserVariablesInPredictionsSpreadsheet = False
  .ShowRawResiduals = True
  .ShowResidualsSquared = False
  .ShowAbsoluteResiduals = False
  .ShowStandardResiduals = False
  .IncludePredictionsInResidualsSpreadsheet = True
  .IncludeObservedInResidualsSpreadsheet = False
```

```
    .IncludeIndependentsInResidualsSpreadsheet = False
    .IncludeSubsetVariableInResidualsSpreadsheet = False
    .IncludeUserVariablesInResidualsSpreadsheet = False
    .ShowRatioAndRankingSensitivityMetrics = True
    .SelectXaxisVariable = 1
    .SelectYaxisVariable = 0
    .CasewiseDeletionOfMD = True
End With

Set optionsdialog = ANNanalysis.Dialog.GetUserCaseSelection
With optionsdialog
    .SelectSpecifiedCases = True
    .UserSpecifiedCases = "1-" & Samples
End With

ANNanalysis.Dialog.SetUserCaseSelection optionsdialog
Set optionsdialog = Nothing

Set optionsdialog = ANNanalysis.Dialog.GetMultipleModelSelection
With optionsdialog
    .ShowOnlyStandaloneNetworks = True
    .SelectTheEnsembleItself = True
    .SelectedModels = "1-" & Bootstraps
End With

ANNanalysis.Dialog.SetMultipleModelSelection optionsdialog
Set optionsdialog = Nothing

'Summary Spreadsheet

Dim S1 As Spreadsheet
Set S1 = ANNanalysis.Dialog.PrintModelSummarySpreadsheet(1)
S1.Visible = True

'S1.Activate
```

```
Set Cells = S1.CellsRange(1, 1, Bootstraps, 12)

Cells.Select
S1.CopyWithHeaders

S1.Close


Results.Activate
Set Cells = Results.CellsRange(1, 1, 1, 1)
Cells.Select
Results.Paste



Results.Value(1, 1) = SourceData.VariableName(i)
   If i = 1 Then
   Results.AddCases Bootstraps, 20
   End If

'Calculating MEAN


Dim columns As Integer
columns = 7

For f = 2 To columns

   Results.Value(h, f) = 1 - 1

   g = h - Bootstraps

Dim Bstraps As Integer
Bstraps = g + Bootstraps - 1
```

```
For g = g To Bstraps

   Results.Value(h, f) = Results.Value(h, f) + Results.Value(g, f)

Next g

   Results.Value(h, f) = Results.Value(h, f) / Bootstraps          'divide by number of
bootstraps


Next f

Results.Activate                                    'copying mean results into new
spreadsheet
   Set Cells = Results.CellsRange(1, 1, 1, 1)
   Cells.Select
   Results.Copy


Averages.Activate
   Set Cells = Averages.CellsRange(l, 1, l, 1)
   Cells.Select
   Averages.Paste

l = l + 1

Results.Activate
   Set Cells = Results.CellsRange(m, 2, m, 7)
   Cells.Select
   Results.Copy

Averages.Activate
   Set Cells = Averages.CellsRange(n, 2, n, 7)
   Cells.Select
```

```
   Averages.Paste

n = n + 1

Averages.VariableName(1) = "m/z value"                'label variable names accordingly
Averages.VariableName(2) = "Train Perf."
Averages.VariableName(3) = "Select Perf."
Averages.VariableName(4) = "Test Perf."
Averages.VariableName(5) = "Train Error"
Averages.VariableName(6) = "Select Error"
Averages.VariableName(7) = "Test Error"
Averages.VariableName(8) = "Input Number"

Averages.Value(p, 8) = i                              'change variable name width

Dim AveragesWidth As Integer
AveragesWidth = 8
For AveragesVariableWidth = 1 To AveragesWidth

Averages.VariableWidth(AveragesVariableWidth) = 1

Next AveragesVariableWidth

   If i = 1 Then
   Averages.AddCases 10, Inputs
   End If

p = p + 1


   If i <= CInt(Inputs) Then


   If ChkBoot.Value = 1 Then
```

```
    Dim Path_Results As String
    Path_Results = TxtPath.Text + "\Bootstrapping_Results\"
    'Path_Results = "C:\Additive_Approach\Results\"
    Path_Results = Path_Results + CStr(z) + "InputModels," + CStr(i) + ".sta"
    Results.SaveAs Path_Results, True
    End If


    Dim HowManyVariables As Integer
    HowManyVariables = Results.Value(1, 10)

    Results.SelectAll
    Results.EntireRange.Clear                    'save results and clear for next z


'MERGING AND SAVING PREDICTION FILES
If ChkSens.Value = 1 Then

    Dim MergedPred As Spreadsheet
    Dim TempPred As Spreadsheet

    Dim MergeLoop As Integer
    MergeLoop = 1
    Set TempPred = ANNanalysis.Dialog.PrintPredictionsSpreadsheet(MergeLoop)

    Set MergedPred = TempPred
    MergeLoop = MergeLoop + 1

    For MergeLoop = MergeLoop To Bootstraps
        Set TempPred = ANNanalysis.Dialog.PrintPredictionsSpreadsheet(MergeLoop)

        On Error Resume Next
        'ANNanalysis.NeverShowDialog = True

        TempPred.SaveAs txtTempPred.Text, True
```

```
        MergedPred.MergeCases (txtTempPred.Text)
        TempPred.Close

    Next MergeLoop

    MergedPred.SortData 0, 0, 1

    Dim Path_Pred As String
        Path_Pred = TxtPath.Text + "\Predictions\"
        'Path_Pred = "C:\Additive_Approach\Predictions\"
        Path_Pred = Path_Pred + CStr(z) + "InputModel," + CStr(i) + ".sta"
        MergedPred.SaveAs Path_Pred, True

End If


 'sensitivity analysis
If ChkSens.Value = 1 Then

    Dim Sens As Spreadsheet
        Set Sens = ANNanalysis.Dialog.PrintSensitivityAnalysisSpreadsheet(1)



Dim SensAverage As New Spreadsheet
Dim SensSTDev As New Spreadsheet
Dim Median As New Spreadsheet


Sens.AddCases Bootstraps * 2, 5


'The below code calculates the average ratio for each input
```

```
Dim RatioCases As Integer
Dim VariableNumber As Integer
Dim NumberOfVariables As Integer

VariableNumber = HowManyVariables
SensSTDev.Activate

If VariableNumber > 10 Then
   SensSTDev.AddVariables "New", 10, VariableNumber - 10
   Median.AddVariables "New", 10, VariableNumber - 10
End If

For NumberOfVariables = 1 To VariableNumber

Dim TotalCases As Integer
Dim NumberOfCases As Integer

TotalCases = Bootstraps * 2
Sens.Value(TotalCases + 1, NumberOfVariables) = "0"

For NumberOfCases = 1 To TotalCases

Sens.CaseName(TotalCases + 1) = "Ratio Averages"

Sens.Value(TotalCases + 1, NumberOfVariables) = Sens.Value(TotalCases + 1,
NumberOfVariables) + Sens.Value(NumberOfCases, NumberOfVariables)


NumberOfCases = NumberOfCases + 1

Next NumberOfCases

Sens.Value(TotalCases + 1, NumberOfVariables) = Sens.Value(TotalCases + 1,
NumberOfVariables) / Bootstraps
```

```
'the following will compute STDev for ratios
'Sens.Visible = True

'SensSTDev.Visible = False
SensSTDev.AddCases 10, 100

Dim SDevTotalVar As Integer
Dim SDevVar As Integer

Dim SDevCase As Integer
Dim SDevTotalCases As Integer
Dim AverageCase As Integer

SDevCase = 1
SDevTotalCases = TotalCases
For AverageCase = 1 To SDevTotalCases


'Sum and square averages from actual values

SensSTDev.Value(SDevCase, NumberOfVariables) = (Sens.Value(AverageCase,
NumberOfVariables) - Sens.Value(TotalCases + 1, NumberOfVariables))

SDevCase = SDevCase + 1
AverageCase = AverageCase + 1
Next AverageCase


'sum squared averages


Dim TotalSquared As Integer
Dim Squared As Integer
Dim cases1 As Integer
```

```
Dim cases2 As Integer
Dim cases3 As Integer
Dim cases4 As Integer
Dim cases5 As Integer
Dim cases6 As Integer
Dim cases7 As Integer

cases1 = Bootstraps + 1
cases2 = Bootstraps + 2
cases3 = Bootstraps + 3
cases4 = TotalCases + 1
cases5 = TotalCases + 2
cases6 = TotalCases + 3
cases7 = TotalCases + 4


SensSTDev.Value(cases1, NumberOfVariables) = "0"

TotalSquared = Bootstraps
For Squared = 1 To TotalSquared


SensSTDev.Value(cases1, NumberOfVariables) = SensSTDev.Value(cases1,
NumberOfVariables) + SensSTDev.Value(Squared, NumberOfVariables) ^ 2

Next Squared

'Squared value over N cases

SensSTDev.Value(cases2, NumberOfVariables) = SensSTDev.Value(cases1,
NumberOfVariables) / Bootstraps

'SQRT sqaured value over N to equal Standard Deviation
```

```
SensSTDev.Value(cases3, NumberOfVariables) = Sqr(SensSTDev.Value(cases2,
NumberOfVariables))


SensSTDev.Activate
'Copy into original spreadsheet

Sens.Value(cases5, NumberOfVariables) = SensSTDev.Value(cases3,
NumberOfVariables)
Sens.CaseName(cases5) = "Ratio ST Dev"

'calculate standard error

Sens.Value(cases6, NumberOfVariables) = Sqr((Sens.Value(cases5, NumberOfVariables)
^ 2 / Bootstraps))
Sens.CaseName(cases6) = "Standard error mean"

'calculate 95% ci

Sens.Value(cases7, NumberOfVariables) = Sens.Value(cases6, NumberOfVariables) *
1.96
Sens.CaseName(cases7) = "95% ci"

'Calculating the median value from the ranks

Median.Activate
Median.AddCases 10, 15

Dim MedTotalCases As Integer
Dim MedCases As Integer
Dim CasePaste As Integer


MedTotalCases = Bootstraps * 2
```

```
Dim cases8 As Integer
cases8 = MedTotalCases + 5

CasePaste = 1
For MedCases = 2 To MedTotalCases

Median.Value(CasePaste, NumberOfVariables) = Sens.Value(MedCases,
NumberOfVariables)

CasePaste = CasePaste + 1

MedCases = MedCases + 1
Next MedCases

Median.SortData NumberOfVariables, 0, 0


'Determining whether number is divisible by 2,
'to in turn determine whether it is even or odd
'so that correct median value can be calculated

Dim MyNum As Integer
Dim EvenNumber As Boolean

MyNum = CInt(txtBoot.Text)


If MyNum Mod 2 Then
   EvenNumber = False
Else
   EvenNumber = True
End If

If EvenNumber = True Then
```

```
   Sens.Value(cases8, NumberOfVariables) = (Median.Value((Bootstraps / 2),
NumberOfVariables) + Median.Value((Bootstraps / 2) + 1, NumberOfVariables)) / 2
   Sens.CaseName(cases8) = "Median Rank"
   Sens.CaseNameWidth = 1.5

Else
   Sens.Value(cases8, NumberOfVariables) = (Median.Value((Bootstraps / 2),
NumberOfVariables))
   Sens.CaseName(cases8) = "Median Rank"
   Sens.CaseNameWidth = 1.5
End If

Next NumberOfVariables

SensSTDev.EntireRange.Clear
Median.EntireRange.Clear

End If
End If
End If
If ChkSens.Value = 1 Then


   Dim Path_Sens As String
      Path_Sens = TxtPath.Text + "\Sensitivity_Analysis\"
      'Path_Sens = "C:\Additive_Approach\Sens\"
      Path_Sens = Path_Sens + CStr(z) + "InputModels," + CStr(i) + ".sta"
      Sens.SaveAs Path_Sens, True



MergedPred.EntireRange.Clear

End If
Results.EntireRange.Clear
```

```
Dim Path_Averages As String
   Path_Averages = TxtPath.Text + "\Overall_Averages\"
   'Path_Averages = "C:\Additive_Approach\Averages\"
   Path_Averages = Path_Averages + CStr(z) + ".sta"
   Averages.SaveAs Path_Averages, True

Next i

If z = 1 Then
   Averages.DeleteVariables 9, 10                    'delete unneccessary columns
End If


'sort data by selected training/test or validation results

If OpTrain.Value = True Then
   Averages.SortData 5, 0, 0
End If

If OpTest.Value = True Then
   Averages.SortData 7, 0, 0
End If

If OpVal.Value = True Then
   Averages.SortData 6, 0, 0
End If


'Averages.Visible = True

Averages.Cells(1, 1).EntireRow.Select
   Averages.Copy

Summary.Activate
```

```
   Summary.Cells(z, 1).EntireRow.Select
   Summary.Paste


If z = 1 Then
   Summary.AddCases 10, 100 + Inputs
End If


If z = 1 Then

   Dim SummaryWidth As Integer                    'change variable name width
   SummaryWidth = 8
   For SummaryVariableWidth = 1 To SummaryWidth

   Summary.VariableWidth(SummaryVariableWidth) = 1

   Next SummaryVariableWidth

   Summary.VariableName(1) = "m/z value"          'label variable names
accordingly
   Summary.VariableName(2) = "Train Perf."
   Summary.VariableName(3) = "Select Perf."
   Summary.VariableName(4) = "Test Perf."
   Summary.VariableName(5) = "Train Error"
   Summary.VariableName(6) = "Select Error"
   Summary.VariableName(7) = "Test Error"
   Summary.VariableName(8) = "Input Number"

   Summary.DeleteCases 9, 10
End If

'SAVING DATA
```

```
Dim Path_Summary As String
   Path_Summary = TxtPath.Text + "\Overall_Summary\"
   'Path_Summary = "C:\Additive_Approach\Summary\"
   Path_Summary = Path_Summary + "Summary.sta"
   Summary.SaveAs Path_Summary, True

   Path_Averages = TxtPath.Text + "\Overall_Averages\"
   'Path_Averages = "C:\Additive_Approach\Averages\"
   Path_Averages = Path_Averages + CStr(z) + ".sta"
   Averages.SaveAs Path_Averages, True

If inputlist = ",," Then
inputlist = ","
End If

inputlist = inputlist + CStr(Averages.Cells(1, 8))

inputlist = inputlist + ","

'Correlation MATRIX

If ChkSens.Value = 1 Then

Dim Corr As New Spreadsheet
Dim CaseName As Integer
CaseName = 1
Dim VarNameOrig As Integer
'corr.Visible = True
'Sens.Visible = True
Averages.Visible = True


If z <= 1 Then

   Corr.AddCases 10, SourceData.NumberOfVariables
```

```
    Corr.AddVariables 10, CInt(Additions)

        For VarNameOrig = 1 To SourceData.NumberOfVariables
        Corr.CaseName(CaseName) = SourceData.VariableName(VarNameOrig)
        CaseName = CaseName + 1
        Next VarNameOrig
    Corr.CaseNameWidth = 1
End If

tempsens.Activate
Dim CorrVar As Integer
Dim Tempz As Integer
Tempz = 1
Dim corrpath As New Spreadsheet

'this opens up the spreadsheet from the best input in the previous addition

Set corrpath = Spreadsheets.Open(TxtPath.Text + "\Sensitivity_Analysis\" + CStr(z) +
"InputModels," + CStr(Averages.Value(1, 8)) + ".sta")
corrpath.Activate
'corrpath.Visible = True
'tempsens.Visible = True

tempsens.Value(z, 1) = Averages.Value(1, 8) 'this creates list of best inputs at each
addition
tempsens.SortData 1, 0, 0

'writing sens ratios to correlation matrix
For CorrVar = 1 To z
    Corr.Value(tempsens.Value(Tempz, 1), z) = corrpath.Value(Bootstraps * 2 + 1,
CorrVar)
    Tempz = Tempz + 1
Next CorrVar

Dim Matrix As Spreadsheet
```

```
Set Matrix = Corr

'assign blue to variable importance if it declines
'assign red if variable importance increases

Dim fontcase As Integer
Dim fontvar As Integer

For fontvar = 2 To Matrix.NumberOfVariables

For fontcase = 1 To Matrix.NumberOfCases

If Matrix.Value(fontcase, fontvar - 1) > 0 Then
   If Matrix.Value(fontcase, fontvar) > 0 Then

      If Matrix.Value(fontcase, fontvar) < Matrix.Value(fontcase, fontvar - 1) Then
      Matrix.Cells(fontcase, fontvar).Font.Background = RGB(0, 0, 255)
      Matrix.Cells(fontcase, fontvar).Font.Color = RGB(255, 255, 255)
      End If

      If Matrix.Value(fontcase, fontvar) > Matrix.Value(fontcase, fontvar - 1) Then
      Matrix.Cells(fontcase, fontvar).Font.Background = RGB(255, 0, 0)
      End If
   End If
End If

Next fontcase
Next fontvar

Matrix.EntireRange.Font.Bold = True

Corr.SaveAs TxtPath.Text + "\correlation_matrix.sta", True

End If
```

```
Averages.EntireRange.Clear

'this code will check for model imporvements, and terminate
'analysis if model shows no improvement for x amount of input additions

If Check1.Value = 1 Then

Dim Best As New Spreadsheet
Best.Activate
Best.Visible = True

  If z > 1 Then

    If z = 2 Then
      Best.Value(1, 1) = Summary.Value(1, 6)
    End If

    If Summary.Value(z, 6) > Best.Value(1, 1) Then
      Improvement = Improvement + 1
    End If

    If Summary.Value(z, 6) < Best.Value(1, 1) Then
      Best.Value(1, 1) = Summary.Value(z, 6)
      Improvement = 0
    End If

  End If

  If Improvement = CInt(TxtTerm.Text) Then
    GoTo MSGBOXEND
  End If
End If

Next z
```

```
MSGBOXEND:

'this code creates a graph or MSE and percentage of samples
'predicted correctly against number of inputs in the model

Summary.Activate
Set Summary = ActiveDataSet

Dim Graph As Analysis
Set Graph = Analysis(sc2dLinePlots, Summary)

With Graph.Dialog

   If OpTrain.Value = True Then
      .Variables = "2 5"
   End If
   If OpTest.Value = True Then
      .Variables = "4 7"
   End If
   If OpVal.Value = True Then
      .Variables = "3 6"
   End If

   .GraphType = scLineMultiplePlot
   .FitType = scFitOff
   .AggregateSpan = 5
   .AggregateValueRangeBar = True
   .DisplayPoint = scLine2DPointStyleAuto
   .IgnoreMissingData = False
   .IgnoreOutOfFramePoints = True
   .GraphStyle = "Normal Graph"
   .DocumentStyle = "Normal Document Size"
End With
```

```
With Graph.Dialog.CategoryOne
   .EnableCategory = False
End With


With Graph.Dialog.CategoryTwo
   .EnableCategory = False
End With


With Graph.Dialog.Options
   .DisplayDefaultTitle = False
   .TitlePosition = scTitleTop

   If OpTrain.Value = True Then
      .Title = "Training Error and Training % Correct"
   End If

   If OpTest.Value = True Then
      .Title = "Test Error and Test % Correct"
   End If

   If OpVal.Value = True Then
      .Title = "Validation Error and Validation % Correct"
   End If

   .DisplayDefaultFootnote = False
   .Footnote = ""
   .DisplayCaseLabels = scCaseLabelOff
   .CaseLabelsVariable = "1"
   .DisplayTextValuesAsAxisValues = True
   .CoordinateSystem = scCoordinateStandard
   .XYAxisPosition = scAxisStandard
   .DisplayFitExpressionInTitle = scFitOptionInTitle
   .PolynomialOrder = scQuadraticOrder
```

```
    .LogBase = scLogOptionTenBase
    .DisplayProgressBar = True
    .RandomSampling = False
    .RandomSamplingMethod = scByNumberOfCase
    .RandomSamplingSubsetSize = 1
    .AxisType(scX) = scLinearScale
    .AxisType(scY) = scLinearScale
    .AxisType(scZ) = scLinearScale
    .AxisType(scV) = scLinearScale
End With


Set AO = Graph.RouteOutput(Graph.Dialog.Graphs)
'AO.Visible = True
If AO.HasWorkbook Then
    Set AWB = AO.Workbook
Else
    Set AWB = Nothing
End If

AO.Workbook.SaveAs TxtPath.Text + "\SummaryGraph.stw", True

'At end of analysis, display message box to user

'ANNanalysis.NeverShowDialog = False


MsgBox "Analysis Complete. Results saved to desired folders"


'ANNanalysis.Run

'With ANNanalysis.Dialog
  ' .VariablesAndTypes = txtconcat.Text
   '.Variables = txtOutput.Text & " | " & InputList & i
```

```
  ' .TypeOfAnalysis = 8
  ' .NetworkListView = " 1 "
  ' .IssueMissingDataWarning = False
  ' .ApplyMemoryLimit = False

'End With

'Dim Path_Network As String
     ' Path_Network = TxtPath.Text + "\Network_Models\"
     '  Path_Network = Path_Network + CStr(z) + "InputModel," + CStr(i) + ".snn"
      ' ANNanalysis.Dialog.SaveNetworkFile Path_Network, True



End Sub
Private Sub Form_Load()
Form1.Visible = False
Form2.Show
Command1.Visible = False
lblterm2.Visible = False
TxtTerm.Visible = False
LblTemp.Visible = False
txtTempPred.Visible = False

End Sub

Private Sub Image1_Click()
CD1.Filter = "Statistica Spreadsheet Files (*.sta)|*.sta|All files (*.*)|*.*"
CD1.ShowOpen
lblSelected.Caption = CD1.FileName

Dim Autofill As Spreadsheet

Set Autofill = Spreadsheets.Open(lblSelected.Caption)
txtInputs.Text = Autofill.NumberOfVariables - 1
```

```
txtAdditive.Text = txtInputs.Text
txtconcat.Text = "1-" & txtInputs.Text & " | " & txtInputs.Text + 1
txtOutput.Text = txtInputs.Text + 1

End Sub

Private Sub chksens_click()

If ChkSens.Value = 1 Then
   LblTemp.Visible = True
   txtTempPred.Visible = True
End If

If ChkSens.Value = 0 Then
   LblTemp.Visible = False
   txtTempPred.Visible = False
End If

End Sub

Private Sub Image2_Click()
lblValidation_Percentage.Caption = 100 - txtTraining.Text
Command1.Visible = True

End Sub

Private Sub lblSens_Click()
End Sub

Private Sub lblAverages_Click()
End Sub
```