

https://dspace.org/

Machine learning and soft computing approaches to microarray differential expression analysis and feature selection.

Perez, Meir

2012-09-12
http://hdl.handle.net/10539/11932

Downloaded from a DSpace repository. https://dspace.org/

MACHINE LEARNING AND SOFT COMPUTING APPROACHES TO MICROARRAY DIFFERENTIAL EXPRESSION ANALYSIS AND FEATURE SELECTION

Meir Perez

A thesis submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Doctor of Philosophy.

Johannesburg, 2011

DECLARATION

I declare that this thesis is my own unaided work. It is being submitted to the Degree of Doctor of Philosophy to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any other degree or examination to any other University.

……
(Signature of Candidate)

…….day of …….year …….
day *month* *year*

ABSTRACT

Differential expression analysis and feature selection is central to gene expression microarray data analysis. Standard approaches are flawed with the arbitrary assignment of cut-off parameters and the inability to adapt to the particular data set under analysis. Presented in this thesis are three novel approaches to microarray data feature selection and differential expression analysis based on various machine learning and soft computing paradigms. The first approach uses a Separability Index to select ranked genes, making gene selection less arbitrary and more data intrinsic. The second approach is a novel gene ranking system, the Fuzzy Gene Filter, which provides a more holistic and adaptive approach to ranking genes. The third approach is based on a Stochastic Search paradigm and uses the Population Based Incremental Learning algorithm to identify an optimal gene set with maximum inter-class distinction.

All three approaches were implemented and tested on a number of data sets and the results compared to those of standard approaches. The Separability Index approach attained a K-Nearest Neighbour classification accuracy of 92%, outperforming the standard approach which attained an accuracy of 89.6%. The gene list identified also displayed significant functional enrichment. The Fuzzy Gene Filter also outperformed standard approaches, attaining significantly higher accuracies for all of the classifiers tested, on both data sets ($p < 0.0231$ for the prostate data set and $p < 0.1888$ for the lymphoma data set). Population Based Incremental Learning outperformed Genetic Algorithm, identifying a maximum Separability Index of 97.04% (as opposed to 96.39%).

Future developments include incorporating biological knowledge when ranking genes using the Fuzzy Gene Filter as well as incorporating a functional enrichment assessment in the fitness function of the Population Based Incremental Learning algorithm.

ACKNOWLEDGEMENTS

The author would like to acknowledge the contributions made by each of his supervisors towards his understanding and appreciation of their respective fields. Many thanks to Professor Tshilidzi Marwala for enriching the author's understanding of Machine Learning and Soft Computing and for his support. Much appreciation is due to Professor David Rubin for exposing the author to the world of Biomedical Engineering and for facilitating the integration of the Engineering and Biological aspects of the project. To Professor Lesley Scott and Professor Wendy Stevens, thanks for facilitating the author's exposure to the world of Molecular Medicine and specifically to microarrays, as well as for the support over the duration of the author's employment in the department. The author would also like to acknowledge the contribution made by the National Research Foundation towards the funding of the project.

Contents

Declaration i
Abstract ii
Acknowledgements iii
List of Figures vi
List of Tables vii
1 Introduction 1
1.1 Microarrays 1
1.2 Microarray Data Analysis 4
1.2.1 Data Pre-processing 4
1.2.2 Gene Selection 5
1.2.3 Co-expression Analysis 11
1.2.4 Functional Enrichment/Biological Pathway Analysis 12
1.2.5 Classification 14
1.3 Problem Statement, Research Questions and Contributions 15
1.4 Data Sets 17
1.5 Implementation and Testing Framework 17
1.6 Conclusion 18
2 Differential Expression Analysis Using Separability Index 19
2.1 Separability Index 19
2.2 SI for Differentially Expressed Gene Set Selection 23
2.3 Implementation and Testing Framework 24
2.3.1 Classifier 25
2.3.2 Pathway Analysis 26
2.4 Results and Analysis 26
2.4.1 Test Data Classification 27
2.4.2 Functional Enrichment and Biological Pathway Analysis 27
2.5 Conclusion 30
3 Fuzzy Gene Filter 32
3.1 Fuzzy Gene Filter Design 33
3.1.1 Input Layer 34
3.1.2 Input Fuzzy Membership Functions 34
3.1.3 Fuzzy Parameter Optimisation 35
3.1.4 Fuzzy Rule Block and Output Fuzzy Membership Functions 38

3.2 Experimental Design 39
3.2.1 Cross-validation 39
3.2.2 Data Sets 41
3.3 Results and Discussion 42
3.3.1 Prostate Data Set Results 42
3.3.2 Lymphoma Data Set Results 43
3.3.3 Discussion 44
3.4 Conclusion 46
4 Stochastic Search Gene Selection 47
4.1 Stochastic Search Algorithms for Feature Selection 47
4.2 Population-Based Incremental Learning (PBIL) 49
4.3 Implementation 52
4.4 Results and Discussion 54
4.5 Conclusion 56
5 Conclusion and Recommendations 57
References 59
Appendix A Supervised Classifiers for Microarray Data Classification 68
Appendix B Fuzzy Inference 75
Appendix C Genetic Algorithm 81

LIST OF FIGURES

Figure 1.1: Affymetrix GeneChip Human Genome Expression Microarray 2
Figure 1.2: Microarray scan (.dat image). 3
Figure 1.3: Intensity Value Probability Density Functions. 6
Figure 1.4: ROC curve indicating the Random classifier slope. 9
Figure 1.5: Heat Map generated from Hierarchical Clustering 12
Figure 2.1: Feature set (axes) with perfect class separability. 22
Figure 2.2: Feature set (axes) with poor class separability. 22
Figure 2.3: Feature set (axes) also with a large hypothesis margin. 23
Figure 2.4: A depiction of how an optimal gene set is selected. 24
Figure 2.5: SI variation as a function of the number of top ranking probe sets 28
Figure 2.6: Hierarchical clustergram of the training samples. 28
Figure 3.1: Overview of the Fuzzy Gene Filter [63]. 34
Figure 3.2: Input fuzzy membership functions. 36
Figure 3.3: FGF Parameter optimisation overview. 38
Figure 3.4: Output fuzzy membership functions 39
Figure 3.5: Butterfly diagram for the prostate data set,. 42
Figure 4.1: Flow Diagram of the PBIL algorithm 50
Figure 4.2: Fitness variation of the fittest individual. 55
Figure A.1: A Three layered Multilayer Perceptron. 68
Figure A.2: Data distributed in a 2 dimensional vector space 70
Figure A.3: Two class data distributed in a 2 dimensional vector space 71
Figure A.4: Three dimensional space projection of Figure A.3. 72
Figure A.5: K-nearest neighbour classifier. 73
Figure B.1: A Crisp Set and a Fuzzy Set describing the height of a person 76
Figure B.2: Fuzzification of the Input Variables 77
Figure B.3: Antecedent to the Consequent Mapping 78
Figure B.4: Aggregation of each Fuzzy. 79
Figure C.1: Individual consisting of a number of genes using binary encoding 81
Figure C.2: Simple Crossover. 83

LIST OF TABLES

Table 1.1: Gene 1 Rank Table..... 7
Table 1.2: Gene 2 Rank Table..... 8
Table 2.1: BIOPAX pathways significantly..... 29
Table 3.1: Prostate data set classification accuracies. 42
Table 3.2: Lymphoma data set classification accuracies. 43
Table 4.1: PBIL\GA\ANOVA Results Summary 54

1 INTRODUCTION

The challenge of identifying genes which characterise types and sub-types of cancers is central to microarray data analysis [1]. Identifying these genes entails implementing a feature selection algorithm whereby genes which are differentially expressed are identified [2]. This thesis describes the development, implementation and testing, of three novel approaches to microarray data feature selection using soft computing and machine learning paradigms. Presented in this chapter is an introduction to microarrays and microarray data analysis. A review of current techniques for microarray data feature selection is presented, highlighting their major problems. The chapter also includes an overview of the data used for the project as well as the implementation and testing framework. The chapter concludes with an overview of the remainder of the thesis.

1.1 Microarrays

Microarrays have revolutionised the way we analyse genomic composition and expression by allowing for high-throughput analysis of a tissue's genome and transcriptome [3]. A microarray consists of thousands of oligonucleotide probes (short sequences of DNA (Deoxyribonucleic acid)) bound on a chip substrate (glass or silicon). There are a number of different types of microarrays. Some microarrays, such as those manufactured by Agilent Technologies, use a dual colour system whereby mRNA (messenger Ribonucleic acid) content from two different sources are labelled with different colour reporter molecules and bound on the same chip. Other microarrays, such as those manufactured by Affymetrix, use a single colour system whereby mRNA from a single tissue type is bound on the chip (Figure 1.1). The microarray data used in this study were generated using the Affymetrix platform. Each oligonucleotide probe on an Affymetrix expression microarray consists of 25 base pairs sampled from the 3' end of an annotated gene.



Figure 1.1: Affymetrix GeneChip Human Genome Expression Microarray

One of the main applications of microarrays is gene expression analysis [1]. Microarrays allow for the simultaneous quantification of the expression levels of thousands of genes; the result being a Gene Expression Profile (GEP) for the sample under analysis. GEPs of different sample-types are compared and gene-sets which are differentially expressed between the samples can be identified. Microarrays have also been applied to genome and proteome analysis. The techniques described in this thesis are applied to gene expression analysis only and focus on identifying differentially expressed genes.

A microarray gene expression experiment is conducted as follows [3]: mRNA is extracted from a tissue sample. The mRNA is then amplified using PCR (Polymerase Chain Reaction), labelled using a coloured fluorescent reporter molecule and hybridised onto the microarray. During hybridisation, a particular mRNA sequence binds to its corresponding probe on the microarray. If the mRNA sequence is abundant then the probe shines bright when placed under fluorescent light. The microarray is then scanned, resulting in an image of the array (Figure 1.2). Bright pixels on the image correspond to probes with a high density of mRNA. The image is then analysed, producing an expression value for each probe set present on the chip.

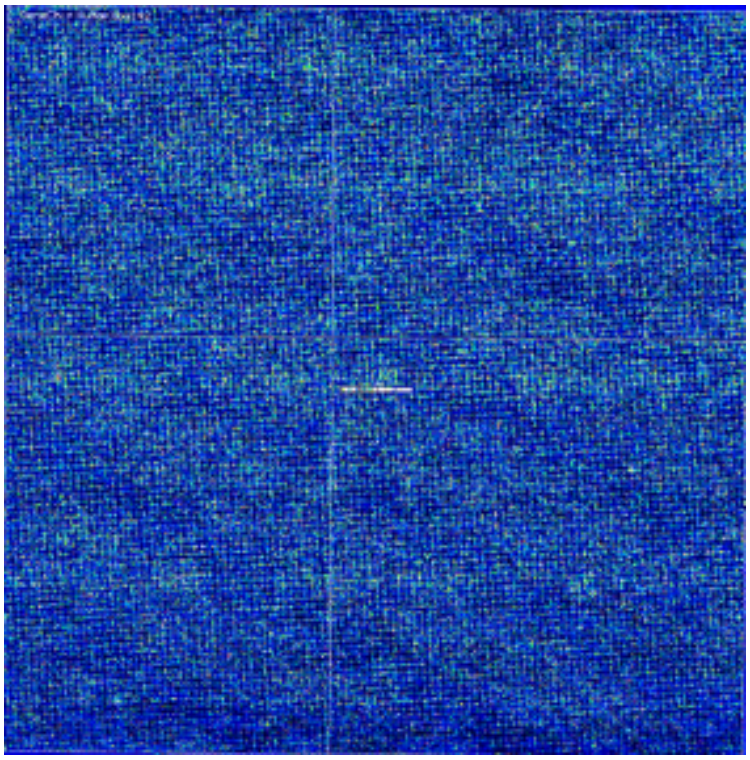


Figure 1.2: Microarray scan (.dat image).

Expression analysis has been extensively applied in cancer research. This is due to the fact that the primary cause of most cancers is genetic mutation (specifically in oncogenes and tumour suppressor genes). Genetic mutation also alters the mRNA content of a cell. One example of this is the t(9;22) translocation which occurs in Chronic Myeloid Leukaemia (CML) [3]. The ABL oncogene from chromosome 9 fuses onto the BCR gene on chromosome 22. The result of such a mutation is a hyperactive form of tyrosine kinase (encoded by ABL), resulting in the cell becoming highly sensitive to growth factor. This causes the cell to undergo mitosis before it is fully mature, resulting in the unstable, positive feedback production of immature myeloid cells. Therefore, CML can be identified by abnormal ABL and BCR expression [4, 5]. Similarly, Acute Myeloid Leukaemia (AML) can be identified by abnormal MYC oncogene expression [6].

Microarrays have been applied to a variety of cancers ranging from Leukaemia [7, 8] to breast cancer [9] and prostate cancer [10]. The fundamental purpose of most of these studies has been the identification of differentially expressed genes for

diagnostic and/or prognostic purposes, as well as for personalised gene-therapy treatment.

The techniques described in this thesis have been applied to expression array data generated from various cancers. Nevertheless, they can be extended to any expression array experiment.

1.2 Microarray Data Analysis

A typical expression array experiment results in data which consists of thousands of expression values per sample processed. Most microarray data analysis packages implement [11-14] five distinct steps in microarray data analysis [1, 5, 15].

1. Data pre-processing (intra-chip and inter-chip normalisation).
2. Gene Selection (identifying differentially expressed genes).
3. Clustering (identifying common expression patterns – co-expression analysis).
4. Functional Enrichment/Biological Pathway analysis (identifying the biological significance of the selected genes).
5. Classification (developing a classification system for unclassified samples).

1.2.1 Data Pre-processing

Before the data can be analysed, it is necessary to [1]:

a) Counteract any technical variation that might be present in the data, such as non-specific binding of mRNA and scanner noise (background correction and summarisation).
b) Normalise data generated from different chips so that samples can be compared to one another (inter-chip normalisation).

The most common algorithms used to achieve this (on Affymetrix expression array data) are MAS 5.0 [16], RMA (Robust Multi-array Analysis) [17] and gcRMA (Genechip RMA) [18]. MAS 5.0 does not implement inter-chip normalisation and only corrects for intra-chip variation. RMA corrects for inter-

chip variation but does not correct for non-specific binding. gcRMA combines inter-chip and intra-chip normalisation and is therefore the algorithm used to normalise all the data used in this study. Microarray data normalisation is not the focus of this thesis, hence, for a thorough treatment of expression array data normalisation techniques the reader is referred to Bolstad et. al. [19].

1.2.2 Gene Selection

Once the data has been normalised and summarised, it is necessary to identify genes which are differentially expressed [1, 2]. The most primitive metric for differential expression is fold change [20]. If a particular gene, on average, is under-expressed for one class of samples and is over-expressed for another class then it is identified as being a class differentiating gene. The problem with fold change is that it does not take into account the variance of a particular gene within a class, thus leading to more appropriate parametric ranking techniques based on hypotheses testing [1, 2]. Hypothesis testing is a method of inferring a numerical fact about a population based on statistical evidence attained from a sample [2].

For two class problems, a two tailed t-test is generally used [1, 2] (Figure 1.3) and for multiclass problems a multivariate, one way Analysis of Variance (ANOVA) is used [1, 2, 21].

The Student t-test was first proposed by William Sealy Gosset (who published under the pen name 'Student') in 1908 [22]. The two-sample t-test is a parametric hypothesis test which examines whether two data-sets were sampled from the same distribution (or have the same mean).

In the context of differential expression analysis, it is assumed that, for a particular gene, the expression values across two classes are of an unequal sample size and have an unequal variance. Hence an unpaired t-test is generally implemented on expression array data [1, 2]. The two sample, unpaired *t* statistic is calculated using the following equation:

$$\boldsymbol{t} = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{s_x^2}{n} - \frac{s_y^2}{m}}} \qquad (1.1)$$

Where:

t = the t statistic

$\bar{x}$ = sample mean of class x

$\bar{y}$ = sample mean of class y

s_x^2 = sample standard variation of class x (intra-class standard deviation)

s_y^2 = sample standard variation of class y (intra-class standard deviation)

n = the number of samples in class x

m = the number of samples in class y

Small intra-class standard deviations and a large inter-class mean difference (the numerator of equation 1.1) is indicative of a good class differentiating gene (small p-value), as is evident from Figure 1.3 A). A p-value is determined based on the overlap of the distributions. If the p-value is less than an arbitrary assigned p-value cut-off (defined by the required confidence interval) then the gene is classified as being differentially expressed. A more recent approach combines the t-statistic with a Support Vector Machine to identify the differentially expressed gene [23-25].

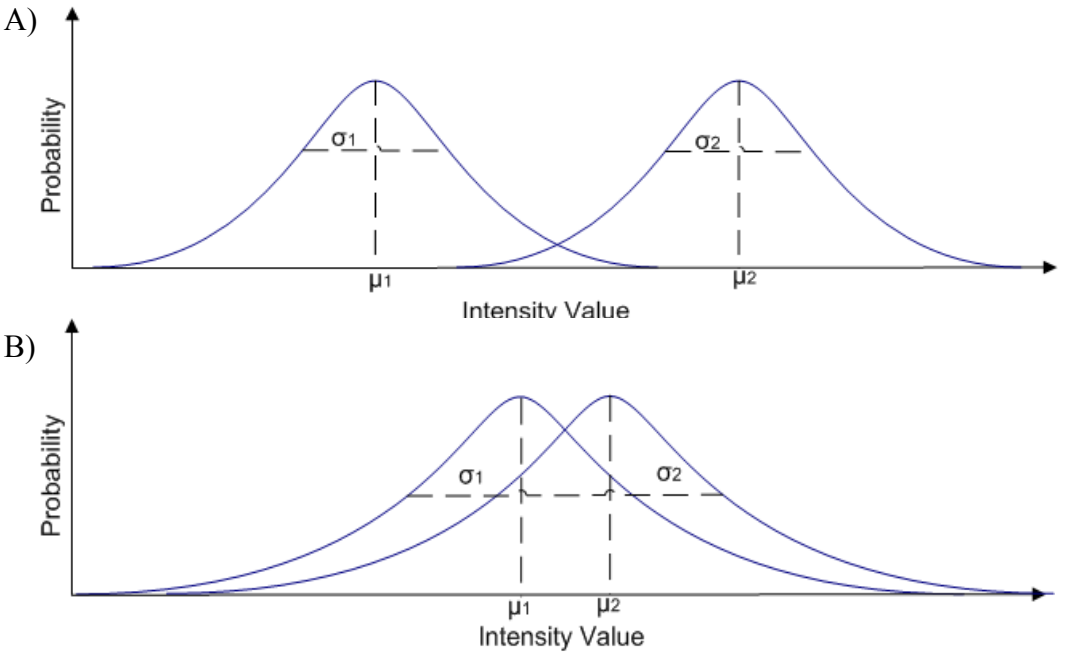


Figure 1.3: Intensity Value Probability Density Functions of a gene which is differentially expressed. B) Intensity Value Probability Density Functions of a gene which is not differentially expressed. Each curve depicts the distribution of intensity values of the two classes.

ANOVA, which was first advanced by Ronald A. Fisher in 1918 [26], assesses whether data from several groups share a common mean. In its most basic form, ANOVA is a generalisation of the two-sampled t-test for sampled data which consist of more than two classes [2, 21, 26]. ANOVA compares the sample variances of each of the data classes, or Mean Square Error (MSE), with the variance of the entire data set, or Mean Square Between (MSB). If MSB is similar to the MSE values then the null hypothesis stands (the classes of data was sampled from the same distribution) and the p-value is high. If they are different then at least one of the classes was sampled from a different distribution, indicated by a small p-value.

In the context of gene ranking, the null hypothesis is that, for a particular gene, there is no difference in mean intensity values between samples from different classes [21]. Hence the smaller the p-value generated from the test, the better the gene's class differentiating ability.

Non-parametric techniques, such as the Wilcoxon test, have also been used on microarray data [27]. The Wilcoxon test, first advanced by Frank Wilcoxon in 1945 [28], is a non-parametric hypothesis test which sums the ranks of samples of a particular class and based on the rank sum, determines a p-value. For a particular gene, the samples are ranked in order of increasing intensity value. The rank values of the samples from each class are then summed. If the sum-of-ranks are similar then the gene does not differentiate between samples of different classes and hence will have a high p-value. If the rank sums are different then gene is differentially expressed. The following hypothetical example illustrates how the Wilcoxon test works for two genes.

Table 1.1: Gene 1 Rank Table

Sample Class	A	B	B	A	B	A
Expression Value	-4.6	-4.1	1.6	2.8	3.6	5.1
Rank	1	2	3	4	5	6

Table 1.2: Gene 2 Rank Table

Sample Class	A	A	A	B	B	B
Expression Value	-7.1	-6.8	-6.4	4.3	4.6	5.1
Rank	1	2	3	4	5	6

From Table 1.1, the rank sum for the samples in class A for gene 1 is 11 and for those in class B is 10. Since the rank sums are similar, it cannot be assumed that the expression values of gene 1 for class A samples were sampled from a different distribution to those in class B. Hence gene 1 will have a large p-value. On the other hand, from Table 1.2, the rank sum for the samples in class A for gene 2 is 6 and for those in class B is 15. This is indicative of the expression values of gene 2 for samples from class B being sampled from a different distribution to those in class A. Hence gene 2 will have a low p-value.

A more recent non-parametric approach for p-value estimation involves the use of Receiver operating characteristic (ROC) analysis [29]. ROC analysis is used to assess the performance of a classifier by depicting the tradeoffs between hit rates and false alarm rates [30]. A ROC curve (Figure 1.4) is a plot of the sensitivity (true positive rate) of a classifier vs. the false positive rate [30].

ROC analysis was originally used in signal detection theory to assess the accuracy of correctly classifying radar signals [29, 30]. It has also been extensively applied to medical diagnostic performance analysis [31]. ROC analysis has been recently applied to microarray gene ranking [32], where each gene is assigned a p-value based on its performance as a classifier. In the context of Machine Learning, it has been used for model comparison by assessing the area under the ROC curve (AUC) and hence deriving the ROC AUC statistic [29]. This approach has been criticised since AUC is a noisy classification measure [33] and has been proven to be problematic in model selection [34].

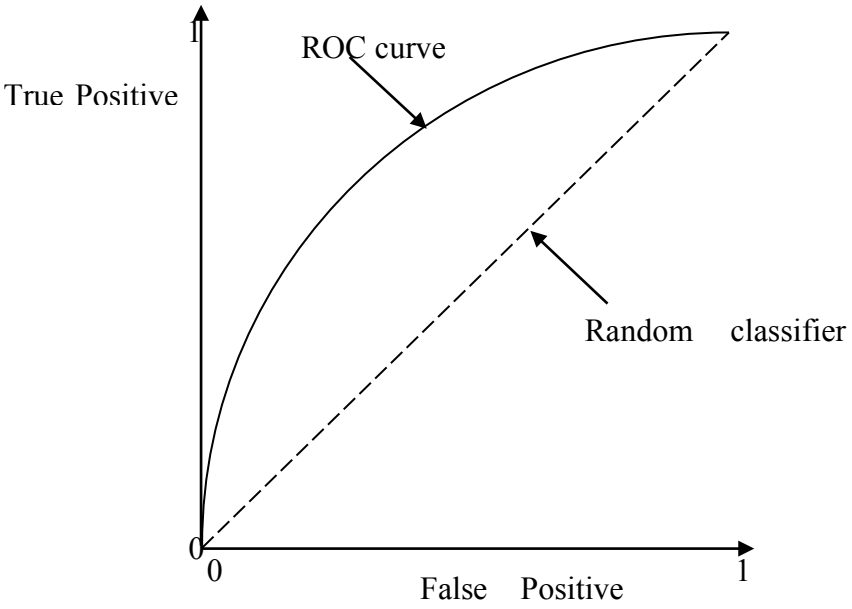


Figure 1.4: ROC curve indicating the Random classifier slope.

An improvement of the ROC AUC statistic involves evaluating the area between the classifier's ROC curve and the non-discriminatory line or random classifier slope [30], as depicted in Figure 1.4. Based on this area, a p-value is generated, evaluating the class distinctive performance of the classifier as compared to randomly guessing the class distribution. If the number of correctly guessed samples is the same the number of false alarms then the classifier is no better than randomly assigning class labels to each sample [30]. This is represented by the random classifier slope. Hence, the area between the ROC curve and the random classifier slope evaluates the randomness associated with the classifier. If the area is large, then the classifier demonstrates a high positive hit rate and a low false alarm rate, indicative of a good classifier, also demonstrating a low level of randomness. This will result in the classifier being assigned a small p-value. If, however, the area is small then the classifier is no better than randomly assigning class labels, resulting in a small p-value.

Depending on whether or not microarray data is normally distributed determines whether a parametric or a non-parametric test is suited for differential expression

analysis. A number of studies [35-37] have demonstrated that different hypothesis tests can produce different sets of differentially expressed genes on the same data set with some sets differing by more than 50 % (50 % of genes found by one algorithm were not identified by the other). This can in turn have a significant effect on classification accuracy as well as pathway analysis. One of the central themes in this thesis is the development of a holistic gene selection algorithm which combines both parametric and non-parametric features of the data (see Section 1.3).

Since the data produced by an expression array experiment consists of the expression values of thousands of genes, multiple hypothesis tests are carried out [37]. The problem with using multiple hypothesis tests is that the number of false positives detected increases with the number of tests implemented [38]. For example, if a 95% confidence is required and 1000 differentially expressed genes are identified, then typically 50 genes are false positives (classified as differentially expressed when they are actually not). Hence a multiple hypothesis correction is required [37]. One of the most stringent multiple hypothesis correction is the Bonferroni correction [37] whereby the cut-off p-value is divided by the number of tests carried out, generating a more stringent p-value cut-off. For example, if a p-value cut-off of 0.05 is used (for 95% confidence) and there are 25000 genes being analysed, then the corrected cut-off is $2x10^{-6}$.

The problem with this approach is that some genes which are differentially expressed could be excluded due to the stringency of the corrected p-value cut-off [37]. As a result, a new statistic was introduced by Storey and Tibshirani [39]: the q-value. The q-value is based on the False Discovery Rate, introduced by Benjamini and Hochburg [40] and corrects for the number of false positives detected by a standard t-test. Selecting genes for classification based on their q-values has become the gold standard for feature selection.

The fundamental problem with all the approaches described in this subsection is that they all rely on assigning an arbitrary q-value/p-value cut-off [41]. One of the major themes of this thesis is the advancement of non-arbitrary approaches to gene selection (see Section 1.3).

Once the differentially expressed genes have been identified, they are analysed for co-expression and functional enrichment and are used to develop classifiers for diagnostic and prognostic purposes.

1.2.3 Co-expression Analysis

Co-expression analysis is carried out by identifying clusters of genes which have similar expression patterns across samples and samples which have similar expression patterns across genes [42, 43]. There are number of clustering techniques that are used for co-expression analysis. The most common technique is hierarchical clustering [44]. Specifically, agglomerative (as opposed to divisive) hierarchical clustering (Figure 1.5) has been used to discover new types and subtypes of cancers as well as to investigate tumorigenesis mechanisms [45].

Agglomerative hierarchical clustering works as follows [45]: each data point (gene or sample) is placed in its own cluster. The distances between data points are determined, based on a distance metric (Euclidean distance, 1 – Pearson correlation coefficient or entropy). Initially, closer data points are paired forming similar clusters. Closer clusters are then also paired based on their proximity. The linkage scheme defines the point in the cluster from which distances are measured (single linkage – the nearest points, complete linkage – the furthest point, or average linkage – the centre of the cluster). This process is iterated until all samples end up in one cluster.

Clustering is represented graphically in the form of a dendogram or cluster tree (Figure 1.5). Each branch on the tree represents a cluster of data points. Within the same tier of branches, the proximity of two clusters indicates their similarity. The clustered data is finally represented in the form of a heat map (Figure 1.5) where a colour map is used to grade expression values (typically, red represents high expression values while green represents low expression values).

Hierarchical clustering is prevalent in microarray literature because it does not require a-priori knowledge of the number of clusters present in the data [45], making it completely unsupervised (the number of gene clusters are typically unknown).

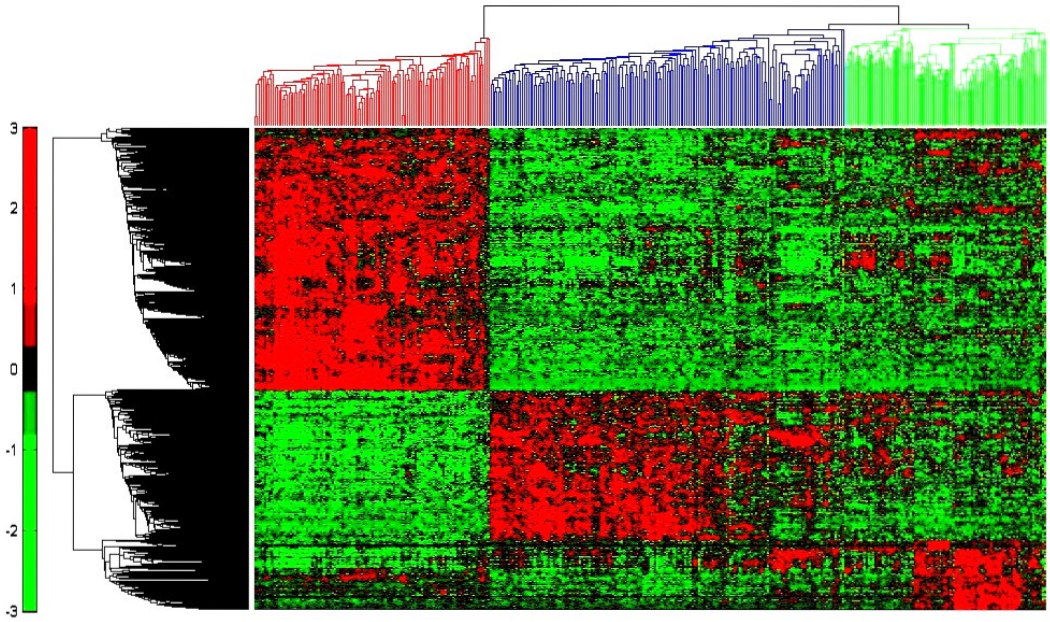


Figure 1.5: Heat Map generated from Hierarchical Clustering. The cluster trees on the side and top of the heat map indicate proximity between different samples and clusters. Red represents over-expression of a particular gene (relative to the median) while green represents under-expression.

This is in contrast to other clustering algorithms, such as k-means clustering, PCA (Principal Component Analysis) and fuzzy-means clustering, where data is clustered into a predefined number of clusters (or selected randomly and then optimised). The downside of using hierarchical clustering is that it does not allow for the refinement of clusters [45]: once a set of clusters is formed, their configuration cannot be optimised. Nevertheless, there is no consensus as to the best clustering algorithm. In this study, agglomerative hierarchical clustering was used for all clustering due to its prevalence in microarray literature.

1.2.4 Functional Enrichment/Biological Pathway Analysis

Once co-expressed gene clusters have been identified, it is necessary to examine whether they have an over-representation of genes which are involved in a particular biological, molecular or cellular process [1, 46, 47].

Gene ontology (GO) classifies gene function according to biological process, molecular function and cellular location. The GO terms are ordered in a hierarchical structure of relationships where vertical order is an assignment of specificity [47, 48].

Functional enrichment analysis compares the representation of genes from a particular GO [49] term in a list of genes to its representation in the rest of the genome [47, 48]. This is done by implementing a Fisher's exact test for a 2x2 contingency table in order to establish the significance of the over-representation of the GO term [46]. This is followed by a multiple hypothesis test correction factor, based on the False Discovery Rate.

A GO term is a group of genes which share a particular function [49]. Each GO term is assigned to one of the three primary ontologies: molecular, biological or cellular function [49]. Each term is also placed at a particular level, corresponding to the specialisation of its functionality: the higher the level, the more specialised the pathway. Each term has at least one parent (a functionally related less specialised level term). For example, the GO apoptotic term is a parent of the GO anti-apoptotic regulation term.

Biological Pathway analysis [1, 46, 47] is similar to functional enrichment analysis except that genes which belong to common biological pathways are grouped together and defined as a functional set.

A Fisher's exact test is used to analyse a contingency table where the samples sizes are small [46]. A contingency table is used to record and analyse the relationship between two or more variables. From the perspective of functional enrichment analysis, the two variables are 1) the number of genes present in a particular GO term/pathway compared to the number of genes that aren't present and 2) the number of genes in the generated list compared to the rest of the genome [46]. Another approach used to analyse over-representation of genes belonging to a particular genetic pathway is the hyper-geometric test [50].

Functional enrichment/biological pathway analysis serves two purposes [1]. It allows one to assess whether the differentially expressed group of genes has biological relevance thus providing a better understanding of the disease under examination. It can also serve as a validation of the feature selection algorithm by determining the biological significance of the selected features.

An alternative to functional enrichment analysis is Gene Set Enrichment Analysis (GSEA) [51]. GSEA assesses genes as a group by assigning them into a priori functional categories and then correlating them with their class labels. This approach has shown to identify subtle changes of gene expression by assessing global changes of a functional group [51].

1.2.5 Classification

The application of supervised classification systems for cancer diagnosis using microarray data has become prevalent [52, 53] and most microarray studies incorporate supervised classification as an indication of diagnostic feasibility [7, 8, 54-58]. A number of studies have shown relatively high classification accuracies on types and subtypes of cancer samples ranging from lymphomas [54] to prostate cancer [10].

A wide range of supervised learning algorithms have been applied to microarray data for sample classification. Techniques ranging from Artificial Neural Networks (ANN) [59] to Support Vector Machines (SVM) [57], from K Nearest Neighbour (KNN) [60] to Naïve Bayesian Classifiers (NBC) [56] have been applied to microarray data classification and their adequacy assessed. One of the most thorough studies on the subject was carried out by Statnikov et. al. [57] who conclude that the best classifier architecture for cancer expression classification is the One versus Rest Multiclass Support Vector Machine. The reader is referred to Appendix A for a technical overview of the most common microarray data classifiers.

Microarray data is an example of the 'curse of dimensionality' [61] where there are more features per sample than there are samples. Hence feature selection is crucial for microarray data classification. The identification of suitable features to use for classification is just as important as identifying the best classifier architecture: if the best features are chosen then even the simplest of classifiers can achieve high accuracies [61]. In microarray data, the most common choice of features is the top differentially expressed genes [7, 8, 54-57]. As mentioned, current techniques for identifying differentially expressed genes suffer from arbitrarily choosing p-value cut-offs.

Alternatively, the genes can be ranked in order of differential expression and a validation scheme can be used to select the best gene set [57]. The problem with this approach is that it requires the data to be split up into three (training, validation and testing), which is a problem in most microarray experiments where the sample size is much smaller than the feature space. Even the suitability of Leave-k-out cross validation schemes has been questioned as to its ability to successfully reduce over-fitting in microarray data.

1.3 Problem Statement, Research Questions and Contributions

It is evident from the overview of microarray data analysis that the most crucial stage in the analysis process is gene selection [1, 2]. Gold standard gene selection techniques currently rely on an arbitrary assigned p-value cut-off [41] which does not necessarily yield the optimal gene set for classification or for functional enrichment analysis. This is due to the fact that microarray data consists of thousands of features per sample. While a pre-assigned p-value cut off is makes sense when considering a data set with a high sample to feature ratio, in the case of microarray data it would make more sense to rank and select a feature by considering its relationship to the entire feature space.

The first research question addressed in this thesis is: Can a non-arbitrary approach to differentially expressed gene *selection* be implemented which selects features based on their contribution to class differentiation. This approach would also need to outperform current gold standard approaches with regards to classification accuracy, while also displaying functional enrichment. In this approach, a feature is selected based on its comparison to the other features in the data set as opposed to an arbitrary assigned feature set, leading to a more data-centric, non-arbitrary, approach.

In Chapter 2, a novel, previously untested approach to microarray gene selection, based on Separability Index (SI), is presented. This approach addresses the first research question, resulting in a non-arbitrary, data intrinsic, technique for selecting differentially expressed genes. SI gene selection comprises the first

contribution of knowledge towards the discipline of microarray data analysis described in this thesis and is described in a paper by Perez et.al [62].

The second research question addressed in this thesis is whether a more intuitive, holistic, data intrinsic feature *ranking* algorithm can be implemented which, while using statistical features, is not limited by the rigid distinction between parametric and non-parametric approaches. In other words, can we develop a feature ranking technique which incorporates both parametric and non-parametric data-features? Such a technique would also have to be flexible enough to adapt to the specific data set under analysis. As discussed in Section 1.2, the ranking algorithm has a significant effect on classification accuracies as well as functional analysis and hence should be optimised accordingly.

In Chapter 3, a gene ranking algorithm, based on Fuzzy Inference, termed the Fuzzy Gene Filter (FGF) is presented. The FGF is a novel, previously untested approach to gene ranking and attempts to incorporate both parametric and non-parametric features for ranking genes, making it more holistic than pure parametric and non-parametric approaches. The FGF parameters are also optimised for the specific data set under analysis, allowing for a more data-centric approach to gene ranking. FGF forms the second contribution of knowledge towards the discipline of microarray data analysis described in this thesis and is described in a paper by Perez et.al [63].

The final research question addressed in this thesis is whether stochastic optimisation algorithms can be used for microarray feature selection to select better features than the standard rank select approach. In the context of microarray gene selection, a stochastic optimisation algorithm can be used to identify the optimal combination of genes to be used for classification. This approach is also based on identifying the optimal gene set by considering the entire feature space when selecting genes. In Chapter 4, the Population Based Incremental Learning Algorithm (PBIL) is used for microarray feature selection. This approach had previously not been tested on microarray data and hence forms the third contribution of this thesis, also described in a paper by Perez et.al [64].

There are three appendices in this thesis. Appendix A describes the details of the classifiers used to facilitate the comparison of the various feature ranking algorithms described in Chapter 3. Appendix B details the Fuzzy Inference System used in the design of the FGF. Appendix C describes the Genetic Algorithm used to optimise the FGF in Chapter 3 and compared to the PBIL algorithm in Chapter 4.

1.4 Data Sets

A number of microarray data sets were used throughout the project. Initially, online publically available data-sets provided by Statnikov et.al. [65], were used to test some of the techniques presented in this thesis, specifically those presented in Chapters 3 and 4. The microarray data bank consists of 11 sets of microarray data ranging from leukaemia to prostate microarray data sets. The primary problem with this data is that only sample labels and expression values are provided. The data does not include gene annotation information hence functional enrichment analysis could not be implemented on this data.

Hence, another data set, provided by the organisers of the IEEE's Eighth International Conference on Machine Learning and Applications for the conference challenge [66], was used in Chapter 2. The data set, generated using the Affymetrix HG U133 plus 2.0 GeneChip is extensive, comprising 400 training samples: 200 breast cancers, 130 colon cancers and 70 lung cancers. Another 250 unlabelled test samples (50 lung cancers, 100 colon cancers, and 100 breast cancers) were also provided.

1.5 Implementation and Testing Framework

All of the techniques described in this thesis were implemented and tested in MATLAB 7.6.0 (R2008a) on an Intel Core 2 Quad 2.4GHz PC with 8 GB RAM, using the Statistics [67] and Bioinformatics toolboxes [68]. Functional enrichment and pathway analysis were implemented on GeneSpring GX 10.0 [15], querying GO terms and BioPAX Pathways.

In order to assess the performance of the algorithms described in this thesis, they are compared to their respective gold standard counterparts. The SI feature

selection approach is compared to selecting features based on an arbitrary assigned p-value based in classification accuracy. The FGF is compared to gold standard feature ranking approaches, across multiple classifiers (described in Appendix A). PBIL is compared to Genetic Algorithm (described in Appendix C) and the ANOVA rank\select approach, based on maximum SI.

1.6 Conclusion

Microarray data analysis consists of five steps: Normalisation, gene selection, clustering, classification and functional enrichment analysis. Gene selection is the most crucial step. Current gene selection techniques rely on arbitrary rigid statistics. Using various Machine Learning and Soft Computing paradigms, non-arbitrary, holistic, intuitive algorithms for microarray feature selection are implemented, tested and presented in this thesis.

2 DIFFERENTIAL EXPRESSION ANALYSIS USING SEPARABILITY INDEX

The first research question addressed in this thesis pertains to the arbitrary nature of gold standard approaches for differential expression analysis. Presented in this chapter is an approach to feature selection based on Separability Index (SI). SI is used since it is a non-arbitrary quantification of data separability and can be used to compare various feature sets for optimal class separability. The approach presented is a 'rank select' feature selection paradigm, similar to gold standard differential expression analysis, the only difference being the criteria by which the gene set is selected.

As mentioned in the introduction (Section 1.2), classically, a differentially expressed gene set is selected based on a confidence interval, defining a cut-off p-value. The cut-off is thus based on the arbitrary bias of the scientist/biologist as opposed to an inherent feature of the data. This could result in a suboptimal gene set [36], which could adversely affect classification accuracy as well as give a false indication of the biological framework of the disease under examination.

This chapter introduces the concept of a SI. Its application to gene selected is then presented. Finally, a testing framework for the approach is described and empirical results are presented, evaluating the suitability of SI to gene selection. Results are compared to gold standard gene selection approaches.

This chapter is based on a paper which was presented at the IEEE's Eighth International Conference on Machine Learning and Applications in Miami, Florida in December 2009 and is published in the conference proceedings [62].

2.1 Separability Index

A SI indicates the extent of separation between data from two or more classes, based on a group of features [69]. From a classification perspective, the more

class-separable the data, the simpler the classifier required and the better the classification accuracy obtained [69].

There are a number of possible approaches to quantifying separability. As with Hypothesis testing, SI techniques can be roughly divided into parametric and non-parametric approaches. Parametric approaches (such as those based on Gaussian mixture models) involve approximating density functions which best characterise the class distinction assigned to the data [61]. Non-parametric approaches include those advanced by Zighed el. al. [70] (which is based on a Cut Edge Weight statistic) and by Thornton [69].

The SI implemented here is based on the one described by Thornton [69], due to its relative simplicity and computational efficiency. The SI is calculated by determining the fraction of instances which have nearest neighbours belonging to the same class [69]:

$$SI = \frac{1}{N} \sum_{n=1}^{N} NH(n) \tag{2.1}$$

Where:
SI = the Separability Index
N = the total number of instances
n = the n^{th} instance or data point
NH(n) = a 1 bit binary number indicating whether the n^{th} instance has a nearest neighbour belonging to its own class (1) or does not (0).

Thus, if a data set consists of classes which are completely separate (Figure 2.1) the index will have a value of 1 indicating 100% separability. If data is not class-separable (Figure 2.2) then the index is less than 1 depending on the percentage of instances with nearest neighbours of the same class.

The problem with this approach, as described by Mthembu and Marwala [71], is that there is no way of quantifying the degree of separability in feature-spaces where the classes are already separate: the data in Figure 2.1 has the same SI as that of Figure 2.3, even though the classes in the former are 'more' separate. This is remedied by incorporating a Hypothesis Margin (HM) in assessing the extent of

class separability once the classes are already separate. HM measures the distance between an instance's nearest hit (nearest neighbour of the same class) and its nearest miss (nearest neighbour of a different class). All the nearest hits and nearest misses are summed separately and the HM is presented as a ratio of the two sums. Once the classes are completely separated, the modified SI is calculated by taking the HM ratio relative to when the classes were minimally 100% separate. Using the modified SI, the data in Figure 2.3 has a larger SI than that of Figure 2.1.

In the context of microarray gene selection, the SI indicates the extent that a particular set of genes differentiate between groups of samples under different conditions. It is a meaningful metric in quantifying the effectiveness of a set of genes in differentiating classes.

Due to the biological nature of the data, it is not true to assume that only genes which show large class-separability are biologically significant [37]. For example, a gene which controls the expression of other genes, such as transcription factors [72], might be slightly differentially expressed, but the dependence of the other genes on its expression makes it biologically important.

Therefore, for the purposes of microarray feature selection, the unmodified version of the SI is used, treating all feature-spaces which are 100% class-separate equally, so as to allow for the inclusion of minimally differentially expressed genes (various clusters of genes in Figure 2.6, specifically amongst the breast cancer samples, indicate the inclusion of genes which are marginally differentially expressed). This also reduces the computational complexity of calculating the SI.

In the context of Evolutionary Optimisation, the SI can be used as a fitness function, where the feature-space is optimised to maximum SI [73]. This is discussed in greater detail in Chapter 4.

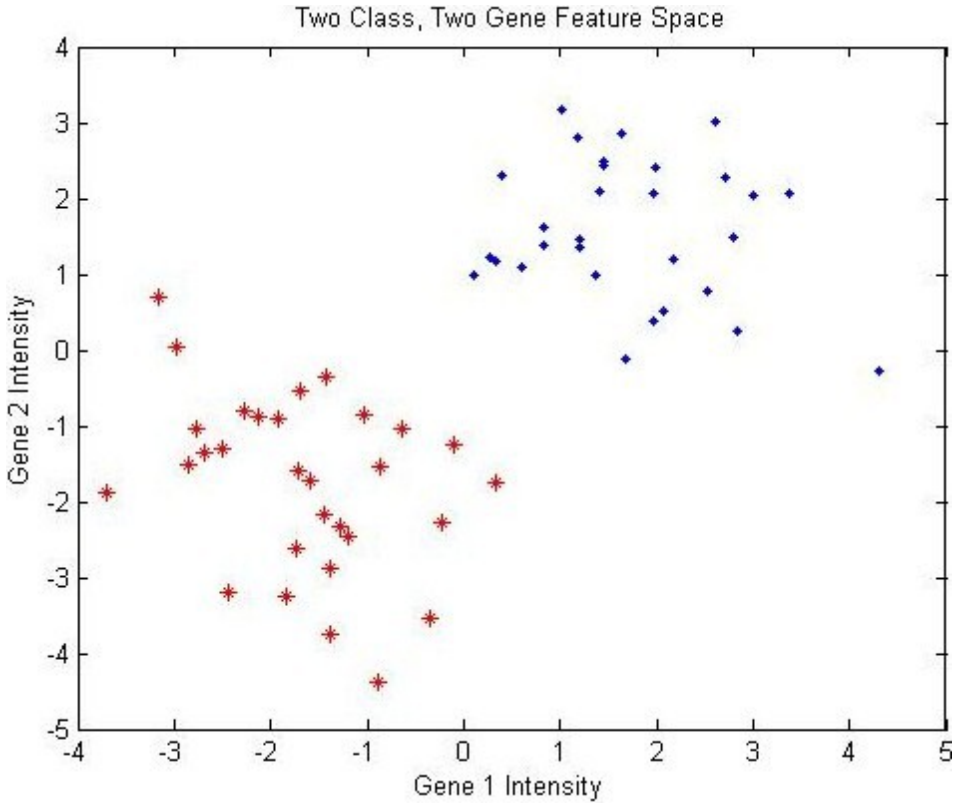


Figure 2.1: Feature set (axes) with perfect class separability hence an SI of 1 but with a small hypothesis margin.

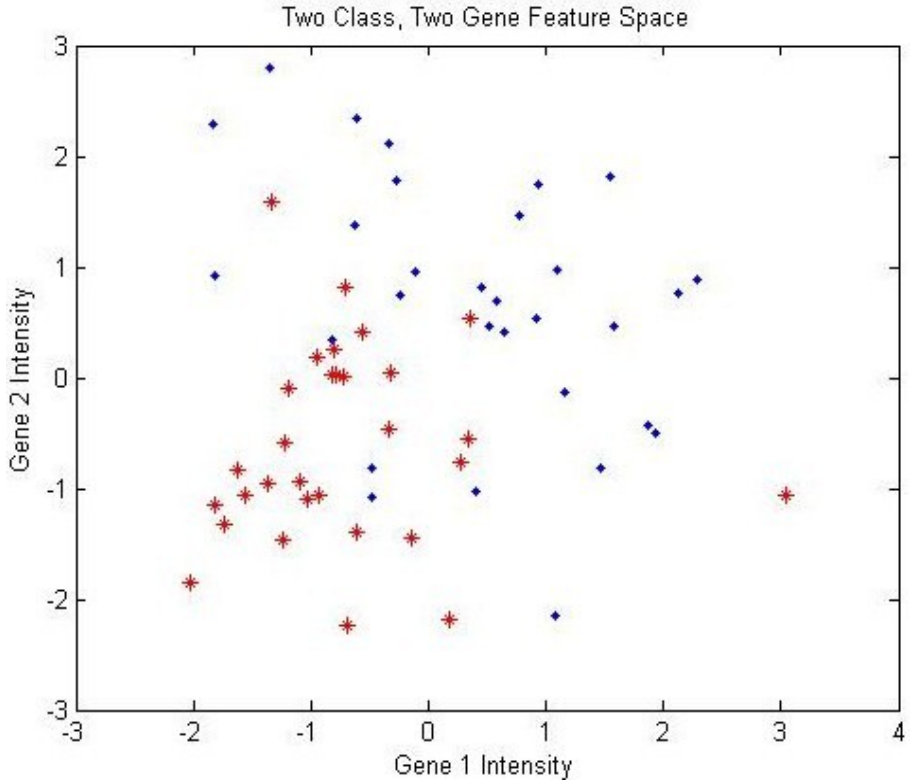


Figure 2.2: Feature set (axes) with poor class separability hence a low SI.

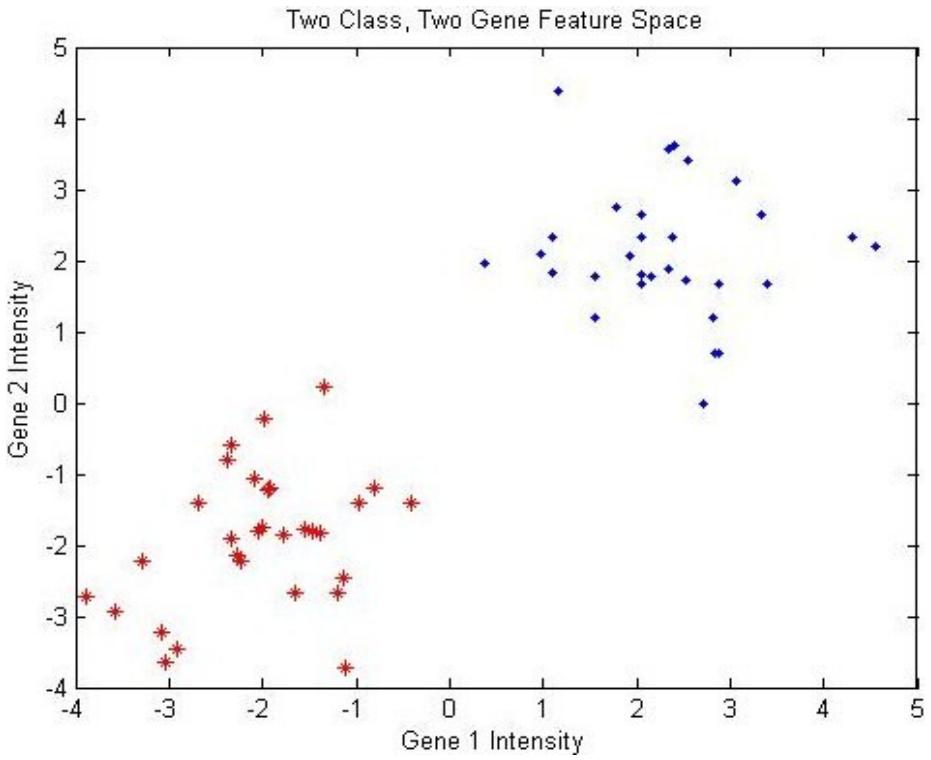


Figure 2.3: Feature set (axes) also with an SI of 1 but with a large hypothesis margin.

2.2 SI for Differentially Expressed Gene Set Selection

The application of SI to microarray data analysis is fairly novel. Besides for the author's publication on the subject [62], more recent publications include those by Unger et. al [74] and Costa et. al. [75], who examine the linear separability of gene expression data.

In the context of differential expression analysis, SI is used to select the optimal number of top ranking differentially expressed genes for classifier training and for functional enrichment analysis. The genes are ranked using either a parametric or non-parametric hypothesis test [2]. Beginning with the top ranking gene, the number of top-ranking genes used to calculate the SI is iteratively incremented, generating an SI vector.

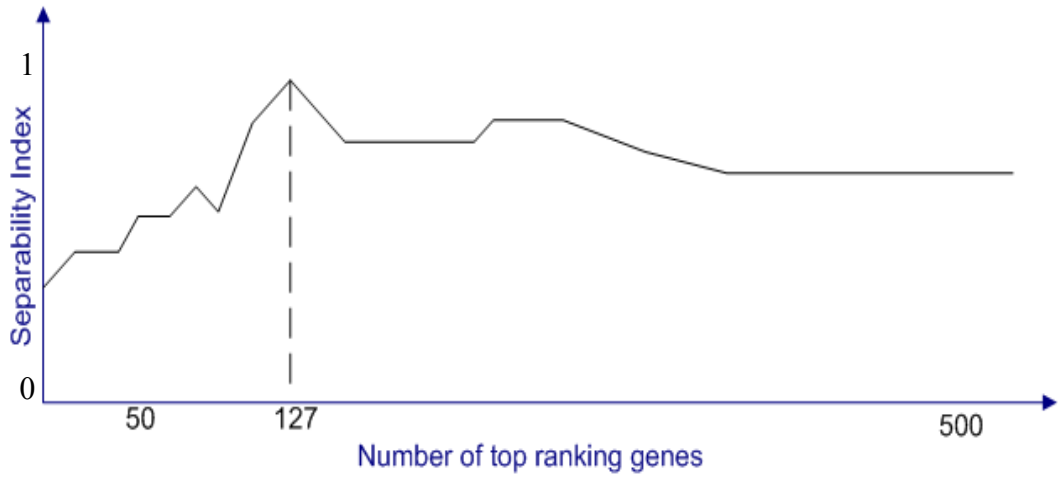


Figure 2.4: A depiction of how an optimal gene set is selected. Maximum SI occurs when the 127 top ranking genes is used. Hence the 127 top-ranking genes are selected for classifier training and functional enrichment analysis.

The entry in the SI vector with the highest SI value corresponds to the gene set which is optimally differentiated between samples from the different classes. If the n^{th} entry is the maximum SI then the top n ranking genes form the optimal gene set for classification and functional enrichment analysis. An example is depicted in Figure 2.4: the 127 top ranking genes is the optimal number to use for classification. The exclusion of the other genes is thus justifiable since they reduce the class differentiating ability of the set of genes.

The advantage of using SI for gene selection is that it serves as a data intrinsic parameter for deciding which genes to include in the gene list. Whereas hypothesis testing assesses each gene for differential expression, SI assesses the differentiability of a set of genes.

2.3 Implementation and Testing Framework

As mentioned in Section 1.4, the technique described here was implemented in MATLAB using a data set comprising 400 training samples: 200 breast cancers, 130 colon cancers and 70 lung cancers. Another 250 unlabelled test samples (50 lung cancers, 100 colon cancers, and 100 breast cancers) were used to test the classification accuracy of the features selected using this approach. The data set can be obtained from [66].

Since the data consists of more than two classes, a multivariate one way ANOVA (see Section 1.2.2) is used to rank the genes in order of differential expression. Once the data has been ranked, the maximum SI cut-off is used to select the genes, as described in Section 2.2. This approach is compared to assigning a confidence interval of 99%, after a Bonferroni correction (see Section 1.2.2), resulting in the selection of genes with $p < 1.8311 \times 10^{7}$. Bonferroni is used since it is more stringent than the FDR correction, and since cancers from different tissues are compared, a large number of differentially expressed genes are expected, requiring a more stringent feature selection.

Two criteria are used to assess the performance of the approach suggested in this chapter (as compared to the standard approach): classification accuracy and pathway analysis.

- Can the features identified using this approach outperform classical statistical techniques when used to train and test a classifier – classification accuracy.
- Amongst the genes selected using this technique, is there an over-representation of genes belonging to a common biological pathway – pathway analysis.

2.3.1 Classifier

A thorough review of supervised classifiers for microarray data is presented by Statnikov et. al [57] (for a summary and a full literature review on classifiers for microarray data see Appendix A). The classifier used for testing is the K-nearest neighbour (K-NN) classifier [76]. The K-NN classifier is an unsupervised, non-parametric classification technique which assigns an unknown instance to the class with the majority of K nearest instances, where K is pre-specified or calculated (for details see Appendix A).

K-NN has proven to produce high accuracies in classifying microarray data [57] and is one of the easiest classifiers to implement [76]. K-NN is used here since the feature set is chosen based on the fraction of instances which have nearest

neighbours belonging to the same class (see Section 2.2), which lends itself to K-NN classification.

2.3.2 Pathway Analysis

As mentioned in Section 1.2, the principal methods of assessing the biological significance of gene lists are functional enrichment and pathway analysis. Functional enrichment tests for the representation of gene ontology terms within a list. Gene ontology (GO) classifies gene function according to biological process, molecular function and cellular location. The GO terms are ordered in a hierarchical structure of relationships where vertical order is an assignment of specificity [47, 48].

To test for functional enrichment, a statistical test (e.g. Fisher's exact) is employed which compares, for each GO term, the number of associated genes within a list to the number of genes associated to that term present in the genome of study [46]. Functional enrichment, therefore, serves to both validate microarray results and to improve biological understanding of phenotypes under investigation.

Pathway analysis functions similarly to functional enrichment analysis, however, for pathway analysis a gene list is assessed for the enrichment of genes found within a database of biological pathways. This provides a deeper understanding of the condition of interest as pathways include specific gene entities as well as interactions and relationships [77].

2.4 Results and Analysis

After implementing the SI feature selection, a maximum SI of 91% was achieved using the top 4222 ranking probe sets. The SI plot in Figure 2.5 indicates how the SI varies as a function of the number of top ranking probe sets used to calculate the SI varies. Figure 2.5 only depicts the SI variation up to the top 5000 ranking probe sets. Anything above 5000 yields lower SI values. The data cursor indicates the point of maximum separability. These probe sets were selected for further analysis. After applying a Bonferroni multiple hypothesis correction, 8610 differentially expressed genes were identified.

2.4.1 Test Data Classification

The K-NN classifier implemented using the genes selected based on maximum SI, classified 96 testing samples as colon cancer, 107 as breast cancer and 47 as lung cancer. The first 100 samples in the testing set are breast cancer samples, the second 100 samples colon cancer and the final 50 samples lung cancer, resulting in an accuracy of 92% was achieved. The K-NN classifier implemented on the genes selected based on a p-value cut-off of $p < 1.8311 \times 10^{7}$ (the corrected p-value corresponding to a 5% confidence interval after implementing a Bonferroni correction) achieved an accuracy of 89.6%.

The experiment was repeated 100 times, each time attaining the same results (100% repeatability). This is due to the fact that the ranking and selection algorithms are deterministic (stochastic approaches are discussed in chapter 4). If the same ranking algorithm is used, the same features are always selected, resulting in identical classification accuracies.

Classification results indicate that genes selected based on the SI criterion attain a higher classification accuracy than those selected using classical statistical approaches (at least for the data set used here). This is attributed to the non-arbitrary nature of the selected features: the features are selected based on a property inherent to the data (maximum class separability), as opposed to an arbitrary assigned cut-off p-value.

2.4.2 Functional Enrichment and Biological Pathway Analysis

The hierarchical clustergram of the training set, using the 4222 top ranking probe sets, is depicted in Figure 2.6. The clusters distinguishing each of the three types of cancers were empirically identified and examined for functional enrichment and the presence of significant biological pathways. A summary of the significant pathways identified from the analysis is presented in Table 2.1.

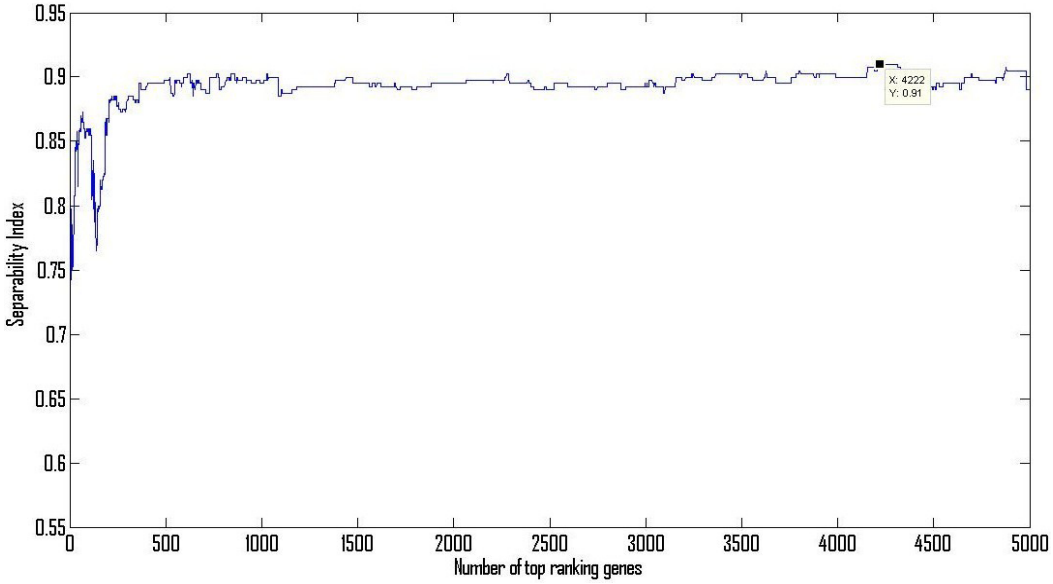


Figure 2.5: SI variation as a function of the number of top ranking probe sets used to calculate the SI increases.

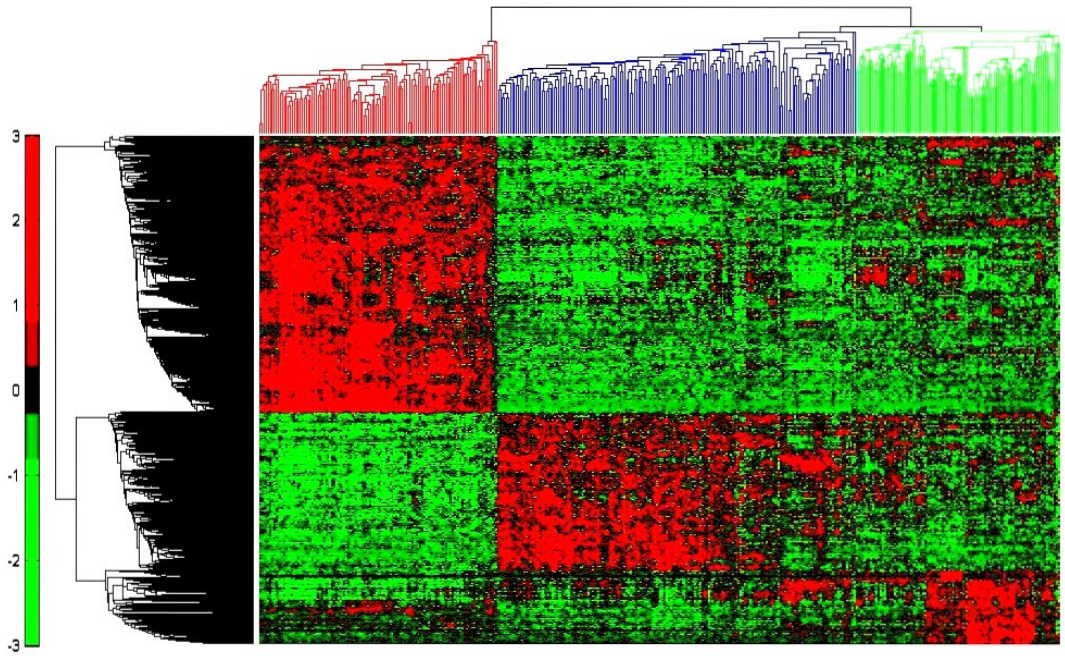


Figure 2.6: Hierarchical clustergram of the training samples, using the top 4222 differentially expressed probe sets. The red cluster contains predominantly colon cancer samples, the blue breast cancer and the green lung cancer.

Table 2.1: BIOPAX pathways significantly (p < 0:05) enriched for each cancer.

Breast	**Colon**	**Lung**
Androgen receptor	Alpha6Beta4Integrin	BCR
	EGFR1	IL5
	Hedgehog	IL6
	TCR	TCR
	TGFBR	
	TNF alpha/NF-kB	

The homogeneity of the expression patterns of the three types of cancers is indicative of their subtype variation. Colon cancer has relatively few molecular sub-classifications resulting in a highly homogenous expression pattern for all colon cancer samples. On the other hand, breast and lung cancers have more sub-classifications, hence their expression patterns show variations, allowing for common subtypes to cluster closer to one another.

The analysis is done in the context of identifying class differentiating genes and is, therefore, limited to differential expression unique to the specific cancer or tissue. Aberrant expression of genes common to all three cancer types would not be identified by this analysis.

Breast Cancer: Breast cancer is well established as being highly subject to endocrine and epigenetic regulation [9, 78]. Gene ontology terms associated with breast cancer in this study generally consisted of terms involving regulation of transcription and nucleotide processing. This analysis identified one pathway unique to breast cancer; the androgen receptor pathway [79]. The breast cancer susceptibility gene 1 has been documented to be a co-activator of the androgen receptor pathway and sex-hormones specifically are known to influence the progression and development of breast cancer [9].

Colon Cancer: Biological process terms associated with colon cancer in this analysis were terms with general cell machinery functions such as mRNA transport, lipid metabolism, nuclear import and negative regulation of transcription. RNA transport mechanisms were most common. However, a

number of significant pathways were identified. According to pathway analysis (Table 2.1) the α-6, β-4 integrin pathways were significant. The role of integrin's in cancer has been extensively investigated as they function across a broad range of critical biological functions including cell adhesion, motility, proliferation, differentiation and apoptosis [80]. The role of π-6, β-4 integrin in invasion by certain solid tumours is well established and it affects cell differentiation in colorectal cancers [81]. Epithelial growth factor receptor (EGFR1) pathway was identified in this analysis and its expression in colon cancer cells has previously been demonstrated and therapies which block this pathway have been modestly successful [82]. The hedgehog pathway (Table 2.1) has previously been documented to be involved in tumour development. It is most known for its involvement in basal cell carcinoma [83]. This analysis identified the TNF-/NFkB pathway. Both TNF- and NF-kB can play a role in cancer; TNF through inflammatory processes while NF-kB can influence the transcription of proto-oncogenes and can stimulate uncontrolled cell proliferation [84, 85].

Lung Cancer: Lung cancer yielded a number of highly specific GO terms; including porphyrin catabolism (GO:0006787) and biphenyl metabolism (GO:0018879). Others included antigen processing and presentation and response to glucocorticoid stimulation. Differential pathways in lung cancer included two cytokine pathways (IL5, IL6), T-cell receptor (TCR) and the B-cell antigen receptor (BCR). This has identified immune pathways in lung cancer compared to colon and breast cancer. BCR is a well-documented [86] proto-oncogene but has not been associated with lung cancer.

2.5 Conclusion

In microarray data analysis differentially expressed gene identification is crucial, both for classifier feature extraction, as well as for significant biological pathway identification. An approach to differential expression analysis, based on a Separability Index, was developed: after ranking the probe sets using a multivariate one-way ANOVA, the optimal number of top ranking probe sets was determined based on maximum class separability (as opposed to arbitrarily assigning a p-value cut-off).

The approach was implemented on a training dataset comprising 400 samples from three types of cancers: colon, breast and lung cancer. The top 4222 probe sets resulted in a maximum separability of 91%. These probe sets were then used to classify a testing dataset comprising 250 samples, using a K-NN classifier, achieving an accuracy of 92%. A second K-NN classifier was also trained using features selected based on $p < 1.8311 \times 10^{7}$, which achieved an accuracy of 89.6%.

Hierarchical clustering was used to identify clusters of genes, from the 4222, with similar expression patterns for each of the three cancers. These clusters were then examined for functional enrichment and significant biological pathways. Significant biological pathways and biological processes, previously described in cancer biology, were identified for all three cancers.

With regards to the specific data set tested, it is thus evident that a non-arbitrary feature selection scheme, based on SI, is preferable to the standard approach since greater classification accuracies can be attained while still identifying a functionally enriched gene set.

The performance is attributed to the non-arbitrary nature of the maximum SI selection criterion, which is an inherent property of the data, as opposed to the arbitrary assignment of a p-value cut-off.

3 FUZZY GENE FILTER

The previous chapter deals with the selection criteria for selecting ranked genes. This chapter deals with the ranking of genes. The effect of different ranking algorithms for gene selection on classification accuracy has been extensively discussed [2]. As mentioned in Section 1.2, most gene ranking algorithms implement either parametric or non-parametric hypothesis tests. The approach described in this chapter combines both parametric and non-parametric data-features with an aim to develop a more holistic gene ranking approach. This is done by implementing a Fuzzy Inference System (FIS) [87, 88] and the ranking system is named the Fuzzy Gene Filter (FGF) [63, 89].

Furthermore, current gold standard feature ranking techniques are not optimised for the specific data-set under consideration [36]: the order in which genes are ranked is independent of the degree of class separability exhibited by the specific data set. Hence, a Genetic Algorithm is incorporated into the FGF, allowing it to adapt to each specific data-set.

This chapter describes the design and implementation of the FGF. The FGF is tested using two publicly available data-sets and the results are compared to those of classical feature ranking techniques, as well as to results previously obtained using the same data-sets. The feature ranking algorithms are compared using four supervised classifiers :Artificial Neural Network (ANN), Support Vector Machine (SVM), Naïve Bayesian Classifier (NBC) and K-Nearest Neighbour (KNN) classifier.

This chapter is based on a paper which was presented at the 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2010, in Cordoba, Spain in June 2010 and is published in Springer-Verlag Lecture Notes in Artificial Intelligence series [63].

3.1 Fuzzy Gene Filter Design

The FGF [63, 89] is a rule based gene ranking technique based on a Fuzzy Inference System. A Fuzzy inference System [90] is a robust decisive tool which mimics the way human beings make decisions based on imprecise data. At the core Fuzzy Inference is fuzzy set theory. Fuzzy set theory, as opposed to classic set theory, assigns each variable a degree of membership [87]: whereas Boolean logic only deals with binary membership, fuzzy logic can assign a single point to multiple groups with varying degrees of membership. For more information on Fuzzy systems, see Appendix B.

Fuzzy inference has been used in many science and engineering applications [91] ranging from flight control [92] to biological signal classification [93]. One of the most successful applications of Fuzzy inference is Fuzzy control [94]: fuzzy controllers have shown to outperform classical control paradigms for multiple input multiple output (MIMO) [95] and non-linear [96] control applications. Fuzzy logic has also been implemented in washing machines [97].

Other applications of Fuzzy inference include the modelling of biological systems. Fuzzy modelling has also been used to model the human retina [98] demonstrating how Fuzzy inference can be used to model biological systems, which are intrinsically highly variable (within two samples of the same system) and complex.

The motivation behind using fuzzy logic for gene ranking lies in its ability to tolerate imprecise data [88]. Fuzzy logic is suitable for microarray data analysis due to its inherent imprecision - expression variation of biological replicates is inevitable [3]. Also, due to the FGF's heuristic nature, diverse biological and statistical expert knowledge can be incorporated when ranking genes. A schematic overview of the FGF is presented in Figure 3.1.

The FGF is based on a Mamdani fuzzy inference architecture [99] (due to its intuitive implementation) and consists of five components: Input layer, input fuzzy membership functions, rule block, output fuzzy membership functions and output layer.

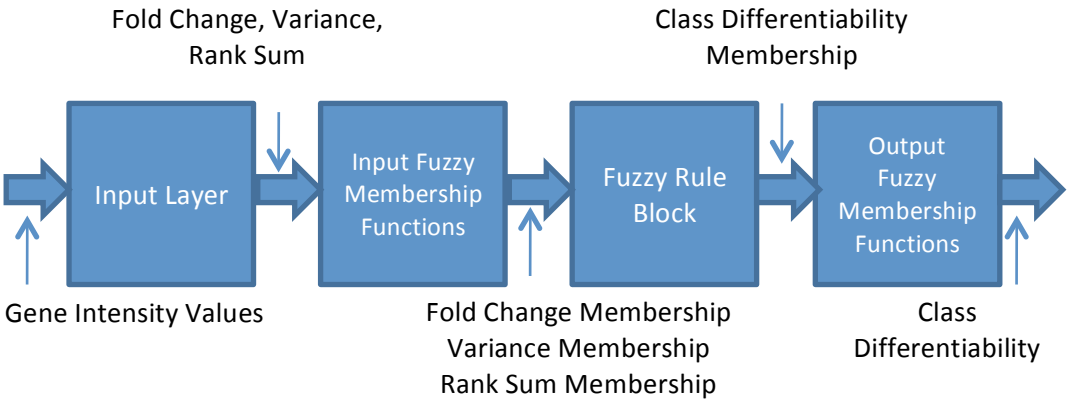


Figure 3.1: Overview of the Fuzzy Gene Filter [63].

3.1.1 Input Layer

The purpose of the input layer is to extract the relevant features which are used for gene ranking. Whereas classical approaches are either parametric or non-parametric, the FGF employs both elements when ranking genes. For each gene, three statistical features are extracted from the data: fold change, intra-class variability (parametric) and the sum of ranks (non-parametric).

The fold change, for each gene, is simply the absolute value of the log2 ratio of the mean intensity values for the two classes [3]. The absolute value is considered since a 2 fold change (a log2 ratio fold change value of 1) is the same as a -2 fold change (a log2 ratio fold change value of -1), the only difference being whether the gene is over or under expressed. This simplifies the FGF since only two fold change membership functions need to be considered.

Intra-class variability is calculated using the denominator of the two sample unpaired t-test [100]. The sum of ranks is calculated as described in Section 2.1. Since both low and high rank sums are indicative of differential expression, the mean rank sum is subtracted from each rank sum value and the absolute value is taken. This simplifies the FGF since only two rank sum membership functions are considered (large and small), as opposed to three (large, medium and small).

3.1.2 Input Fuzzy Membership Functions

The fuzzification of microarray data variables arises naturally from the subjective nature of the assignment of biologically relevant cut-off values. For example, the assignment of a biologically significant fold change cut-off is dependent on the

biological question being asked and could vary between experiments (the use of the universal 2 fold change cut-off has been criticised [20]). Fuzzy set theory allows one to take this subjectivity into account by eliminating the need to assign a crisp cut-off value. Instead, a region which allows for fold change values to be considered as being both small and large, with varying degrees, is introduced.

The input fuzzy membership functions depict the various fuzzy sets to which each input can belong. For example, a gene can have a high or low fold change between samples from two different conditions. Hence two input membership functions are allocated to the fold change input variable, namely high and low, as depicted in Figure 3.2 There are three regions depicted in Figure 3.2:

- The region between 0 and α, where a fold change value is defined as 100% low.
- The region from β upwards defines fold change values which are 100% high.
- The region between α and β where fold change values can belong to both low and high fuzzy sets with various degrees of membership - the fuzzy region.

3.1.3 Fuzzy Parameter Optimisation

Identifying optimal values, is crucial when ranking genes. Just as the assignment of a fold change cut-off value differs from data-set to data-set, so too does the fuzzy fold change region. Hence the optimal α and β values, for each specific data-set, need to be determined.

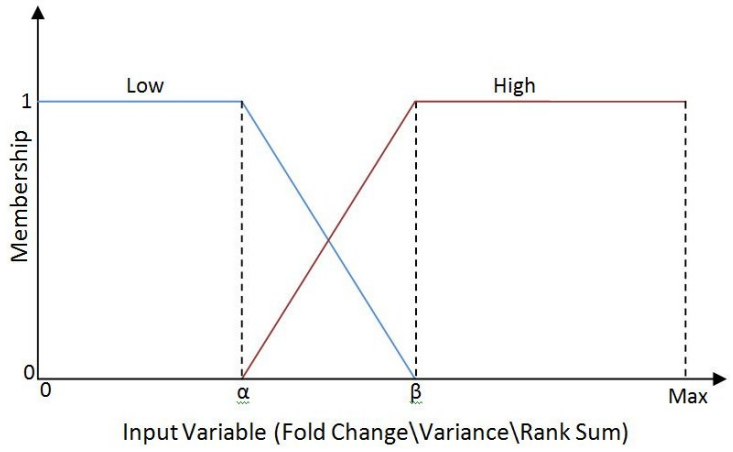


Figure 3.2: Input fuzzy membership functions (fold change, variance and Rank Sum).

A genetic algorithm (GA) is employed to identify these values. GA is a population based stochastic optimisation technique, inspired by biological evolution [101]. A population of individuals is initialised. An individual is defined as a potential solution the function being optimised. Each individual consists of a combination of genes.

GA has been applied to many real-world optimisation problems [102] ranging from optimal antenna design [103] to financial portfolio optimisation [104]. Early forms of evolutionary algorithms was originally developed in the 1960's to solve various engineering problems [105], yet it wasn't until John Holland, in his book *Adaptation in Natural and Artificial Systems* [106], formalised GA into the algorithm which is used today. Nevertheless, it was only in the 1990's, with the advent of sufficient computational power, that GA was applied to more substantial problems [101], such as the travelling salesman problem [107] and job-shop allocation [108]. For a full overview of the Genetic Algorithm, see Appendix C.

One of the most common applications of GA is feature selection [109]. Even within the context of differentially expressed gene selection, GA has been extensively applied [110]. In most of these applications, GA is used to identify the optimal combination of genes to be used for classification. Chapter 4 [64] explores this approach extensively, where GA is compared to the Population Based Incremental Learning algorithm, with regards to microarray data feature selection.

GA has also been extensively applied to fuzzy control system optimisation [100] [111], whereby fuzzy parameters are optimised for a specific task. In this context, they are optimised to identify the gene-set which results in the maximum inter-class separability.

In Chapter 4, a *gene* (in the context of genetic algorithm) is defined as a particular feature to be used for classification and an *individual* is defined as a combination of features since the GA searches for the optimal gene set. In this chapter, where GA is used to optimise a Fuzzy Inference System, a gene is defined as an α or β value. Since there are three inputs, each having two membership functions, each individual consists of six genes. α and β values are bounded decimal numbers which have the following constraints:

- $\alpha < \beta$
- $\beta < Max$
- $\alpha > 0$

Where Max is the largest fold change value present in the data-set.

After initialisation, the population undergoes iterations, or generations, of mating and mutation. Mating entails 'crossing over' or sharing the genes of two individuals to produce offspring, resulting in a new generation of potential solutions. Mutation entails modifying the genes of a randomly selected individual, preventing the algorithm from premature convergence (converging to a local optimum). The population is maintained at a fixed size, where an elite count is used to determine how many of the t-test individuals survive to the next generation.

The GA is guided by a fitness function. The fitness function indicates the proximity of an individual to the optimum or to grade individuals. It is used when selecting individuals for mating and mutation. As with biological evolution, selection for mating favours fitter individuals.

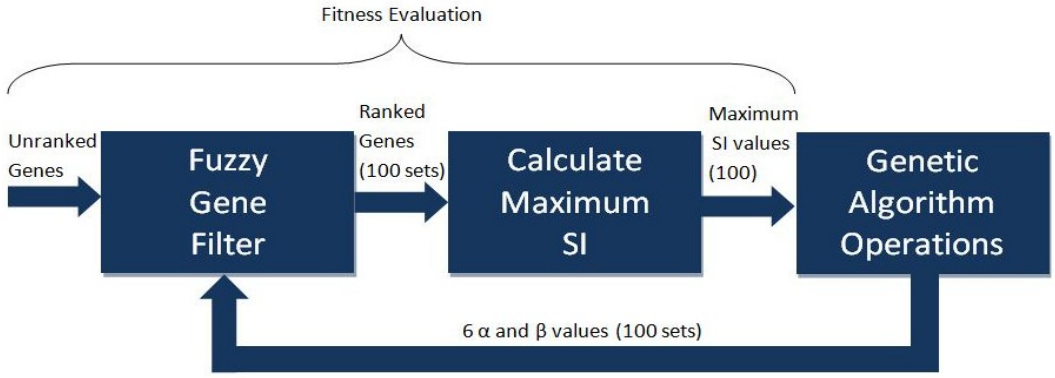


Figure 3.3: FGF Parameter optimisation overview. A population of 100 potential fuzzy parameter-sets are iteratively guided towards identifying the optimal set.

The fitness function used here is the Separability Index [69] (see Chapter 2). Genes are ranked by the FGF using specific α and β values. The fitness value is simply the maximum SI value attained when examining the SI values associated with each top ranking gene-set, as discussed in Section 2.2. An overview of the fuzzy parameter optimisation scheme is depicted in Figure 3.3.

3.1.4 Fuzzy Rule Block and Output Fuzzy Membership Functions

The rule block relates the input fuzzy variables to the output fuzzy variables, using a set of expert knowledge based linguistic expressions. In this application, expert knowledge is extracted from the underlying statistics (both parametric and non-parametric) as described in Section 2.1. For example, if a gene has low intra-class variance, a high fold change and a high rank sum then the gene is deemed to display good class differentiability and is hence assigned to the very high output fuzzy membership function. On the other hand, if the gene has high intra-class standard deviations, a low fold change and a low rank sum then the gene displays poor class differentiability and is assigned to the very low output fuzzy membership function. If two of the three criteria for good class differentiability are met the gene is assigned to the high output fuzzy membership function. If only one criterion is met, then it is assigned to the low output membership function.

Input fuzzy membership functions (antecedents) are combined using a min/max fuzzy operator: a fuzzy OR operation selects the maximum membership of the three fuzzy inputs while a fuzzy AND operation selects the minimum.

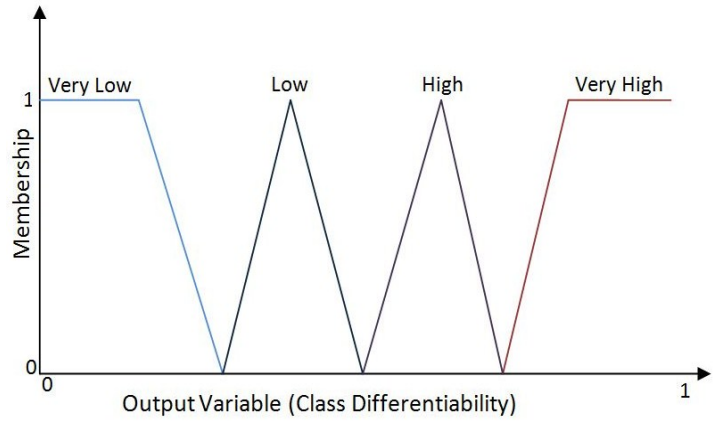


Figure 3.4: Output fuzzy membership functions.

The output fuzzy membership functions (Figure 3.4) depict the various degrees of class differentiability exhibited by the gene, based on the input features: very low, low, high and very high class differentiability. These membership functions are chosen due to the fact that there are three inputs, each having two membership functions. The fuzzy outputs are clipped and aggregated by applying the fuzzy OR operation. A crisp output is attained via centroid de-fuzzification (see Appendix B), producing the degree of class differentiability exhibited by the gene. Class differentiability is expressed as a number from 0 to 1, 0 being the worst class differentiability, 1 being the best class differentiability. The FGF is used to rank all the genes present in the data. The genes are then ranked in order of class differentiability.

3.2 Experimental Design

The purpose of the experiment described in this chapter is to examine how well the FGF performs in ranking features for various classification architectures (KNN, SVM, ANN, NBC), as compared to standard feature ranking approaches (t-test, Wilcoxon test, ROC curve analysis).

3.2.1 Cross-validation

In order to assess the performance of various classifiers on features ranked by each of the ranking approaches, a cross-validation scheme is implemented in order to identify the optimal number of top ranking features to be used for classification. A classifier is iteratively re-trained and tested, incrementing the number of top

ranking genes used until the gene-set which results in highest classification accuracy is identified. This gene-set is then selected as the classifier input space.

It is also necessary to identify the optimal classifier parameters, for each gene-set being tested. Hence, a nested stratified Leave-one-out Cross-validation (LOOCV) scheme is implemented [57]. The scheme consists of an inner loop and an outer loop. The inner loop identifies the optimal parameter values for the classifier (using a 10 fold cross-validation scheme). The outer loop calculates the LOOCV accuracy for the gene-set being tested.

LOOCV consists of training a classifier using all samples except for one. The classifier is then tested using the left-out sample. This process is repeated until each sample has been used to test the classifier. The LOOCV accuracy is then determined by calculating the percentage of correctly classified left-out samples.

LOOCV is commonly used for classification problems where there are a limited number of samples [57]. Typically, one would allocate three sub datasets: A training set (used to train the classifier), a validation set (used to identify optimal classifier parameters and features) and a testing dataset (used to quantify the performance of the classifier on 'unseen' data). If there is a limited number of samples (relative to the number of features) then it is necessary to use the training dataset as the validation set as well and implement a cross-validation scheme, such as the one described here.

Due to the expense of generating microarray data, a typical microarray experiment consists of few samples, compared to the number of features generated per sample. Hence, LOOCV is common in microarray literature [57]. Furthermore, the purpose of this experiment is to compare feature ranking algorithms, as opposed to classifiers, hence the LOOCV accuracy is sufficient to compare feature sets.

This approach is also used since it is similar to the one used by the original authors of the test data-sets, where a KNN classifier was used to diagnose prostate cancer and differentiate between Diffuse Large B-cell Lymphoma and Follicular

Lymphoma [54]. It is also similar to approach taken by Statnikov et.al. in a paper which compares various classifier architectures on microarray data [57].

In the inner loop of the LOOCV, the optimal parameters of the classifiers are identified. For each of the four classifiers tested, the following classifier parameters are optimised:

- For the SVM, the upper-bound constant C is optimised, while using a linear kernel as suggested by Statnikov et. al. [57].
- For the KNN classifier the optimal neighbourhood radius k is identified.
- For the NBC the bandwidth of the initial Gaussian kernel is optimised.
- For the ANN (MLP), the optimal number of hidden nodes is identified (within the range of one hidden node to twice the number of input nodes) while using regularisation to prevent over-fitting. A logistic activation function is used since the MLP is being used as a classifier.

Once LOOCV has been implemented using each of the classification algorithms, on features ranked by each of the feature ranking techniques, classification accuracies are compared. An ANOVA is then implemented to examine the significance of the different accuracies across the various gene ranking algorithms.

3.2.2 Data Sets

The techniques were compared using two publicly available data-sets, both made publicly available by Statnikov [65]. The first consists of 50 healthy and 52 cancerous prostate samples [10, 57]. The second consists of 58 Diffuse Large B-Cell Lymphoma samples and 19 Follicular Lymphoma samples [54, 57]. The prostate data-set was generated using the Affymetrix HG-U95 Gene Chip and consists of 10509 gene expression values per sample [10]. The lymphoma data-set was generated using the HU6800 oligonucleotide array and consists of 5469 gene expression values per sample [54].

Background correction was done using the Affymetrix MAS 5.0 algorithm. In addition, quantile normalisation with a median polish was also implemented.

Table 3.1: Prostate data set classification accuracies and number of top ranking genes (in parenthesis).

	FGF	t-test	Wilcoxon test	ROC
KNN	**96.1% (9)**	93.1% (3)	94.1% (15)	93.1% (6)
SVM	**95.0% (3)**	94.1% (14)	94.1% (19)	95.0% (8)
NBC	**94.1% (3)**	93.1% (22)	93.1% (15)	**94.1% (3)**
ANN	**95.0% (5)**	93.1% (7)	94.1% (14)	94.1% (6)

3.3 Results and Discussion

Table 3.1 and Table 3.2 depict the highest LOOCV accuracies attained by each classifier for each feature ranking algorithm, as well as the optimal number of top ranking genes used to obtain the accuracy (the value in parenthesis).

3.3.1 Prostate Data Set Results

The LOOCV accuracies attained for the various classifiers tested on the prostate data set are summarised on Table 3.1. Figure 3.5 depicts the butterfly diagrams of the various classifiers, depicting the accuracy median and 25th\75th percentiles of the four classifiers.

Classifiers trained with features ranked by the FGF resulted in the highest accuracy, for each of the classifiers tested, compared to the other gene ranking techniques ($p < 0.0231$).

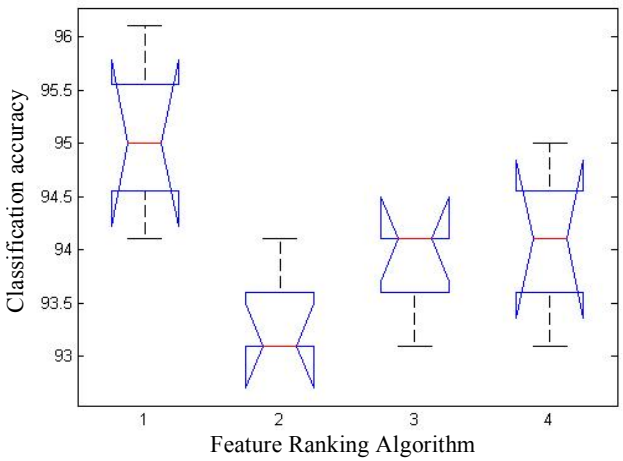


Figure 3.5: Butterfly diagram for the prostate data set, displaying the distribution of classification accuracies for each feature ranking technique (sample 1 is the FGF, 2 the t-test, 3 the Wilcoxon test and 4 ROC curve analysis).

Table 3.2: Lymphoma data set classification accuracies and number of top ranking genes (in parenthesis).

	FGF	t-test	Wilcoxon test	ROC
KNN	**100% (13)**	97.4% (6)	94.8% (4)	98.7% (2)
SVM	**100% (12)**	98.7% (5)	98.7% (39)	98.7% (28)
NBC	**97.4% (5)**	**97.4% (3)**	**97.4% (5)**	**97.4% (3)**
ANN	**98.7% (14)**	94.8% (8)	97.4% (6)	97.4% (4)

The classifier with the highest accuracy is the KNN classifier, attaining an accuracy of 96.1%, when trained using the top 9 ranking genes, as ranked by the FGF. The classifier with the highest accuracy is the KNN classifier, attaining an accuracy of 96.1%, when trained using the top 9 ranking genes, as ranked by the FGF.

The prostate data-set was originally used by Singh et. al. [10] to develop a classifier for prostate cancer diagnosis. The maximum cross-validation accuracy reported in the original paper was 86% using a 16 gene model (genes were ranked using a signal to noise ranking scheme). Statnikov et. al. [57] reported an accuracy of 92% on the same data-set. All the gene ranking techniques presented here outperformed both studies with the FGF.

3.3.2 Lymphoma Data Set Results

The LOOCV accuracies attained for the various classifiers tested on the lymphoma data set are summarised in Table 3.2. Figure 3.6 depicts the butterfly diagrams of the various classifiers, depicting the accuracy median and 25th\75th percentiles of the four classifiers.

Classifiers trained with features ranked by the FGF resulted in the highest accuracy, for each of the classifiers tested ($p < 0.1888$), albeit with less confidence that than with the prostate data set. Both the KNN and SVM classifiers attained the highest accuracy (100%).

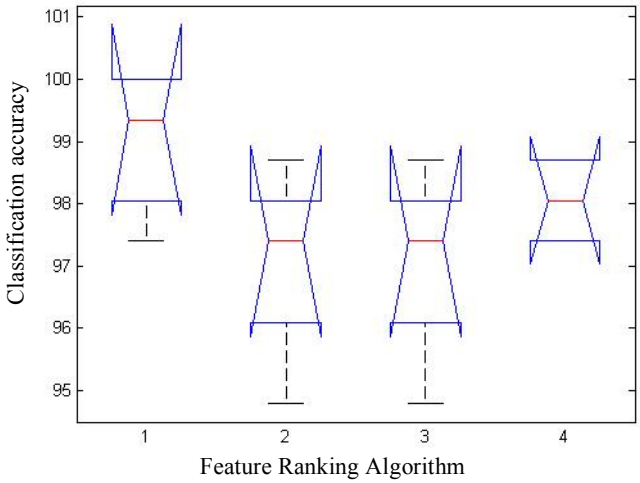


Figure 3.6: Butterfly diagram for the prostate data set, displaying the distribution of classification accuracies for each feature ranking technique (sample 1 is the FGF, 2 the t-test, 3 the Wilcoxon test and 4 ROC curve analysis).

Nevertheless, the SVM is deemed the better classifier since it was able achieve the maximum accuracy with fewer features (the top 12 ranking genes as opposed to the top 13 with the KNN classifier). Similarly, even though the features ranked by ROC curve analysis also resulted in 100% accuracy on the SVM, it did so with the top 28 features. The features ranked with the FGF achieved the same accuracy with only 12 top ranking features.

The lymphoma data-set was originally used by Shipp et. al. [54]. The accuracy reported in the original paper was 77% using weighted voting classification technique. Statnikov et. al. [57] reported an accuracy of 97.5% on the same data-set. The FGF outperformed both studies, attaining an accuracy of 100% using the SVM classifier.

3.3.3 Discussion

The performance of the FGF is attributed to the fact that it is optimised to rank genes in such a way that results in maximum class separability, as well as its incorporation of multiple features of the data when ranking genes. Furthermore, the FGF parameters are optimised to the specific data-set being analysed: the optimised fuzzy parameters for the prostate data-set are different to those of the lymphoma data-set. For example, the FGF α and β values for the fold change

membership functions, optimised for the prostate data-set, are 0.0862 and 0.7787. In contrast, the FGF α and β values, optimised for the lymphoma data-set, are 0.1098 and 0.5378.

The fold change fuzzy region for the lymphoma data-set is smaller than the prostate's. A small fuzzy region indicates less ambiguity in defining a gene as having a high or low fold change. Reduced ambiguity is a result of a clear distinction between genes which have a high fold change and genes which do not.

A data-set which contains genes with high fold change values indicates that the data is highly class-separable. Thus, the lymphoma data is more class separable than the prostate data. This is due to the fact that the two types of samples being compared in the lymphoma data-set originate from different cell lines (B-cell vs. follicular) whereas the prostate samples all have the same cell lineage (the only difference being whether a sample is cancerous or healthy). Hence, differential expression between samples from the prostate data-set is less pronounced than those from the lymphoma data-set.

This is also seen in the fact that the maximum SI obtained from the prostate data-set (0.96) is less than the SI obtained from the lymphoma data-set (1).

In terms of common features selected by the various algorithms, for the prostate data-set, all of the features identified by the t-test were also identified by the FGF. The FGF had six common features with Wilcoxon test, and yet had only two common features with the ROC technique. Algorithmically, this makes sense since the FGF incorporates elements of the t-test and the Wilcoxon test but not the ROC technique.

Similar results are obtained when comparing the features identified by the various algorithms on the Lymphoma data-set: The FGF identified five of the six of the features also identified by the t-test; two of the four features identified by the Wilcoxon test and none of the features identified by the ROC technique.

3.4 Conclusion

The development of a novel approach to expression array data feature ranking, the FGF, has been presented. The FGF considers both parametric and non-parametric data features when ranking genes. The FGF also incorporates a GA for fuzzy parameter optimisation.

After a thorough comparison of the FGF with standard gene ranking algorithms (the t-test, Wilcoxon test and ROC curve analysis), on various classifier architectures (KNN, SVM, NBC and ANN), the FGF was still able to attain the highest LOOCV accuracy on both data-sets ($p < 0.0231$ for the prostate data set and $p < 0.1888$ for the lymphoma data set).

For the prostate data set, a LOOCV accuracy of 96.1%, using the top 9 ranking genes, was attained the KNN classifier. For the lymphoma data set, a LOOCV accuracy of 100%, using the top 12 ranking genes, was attained on the SVM classifier. It is thus evident that (at least for the data sets tested) ranking genes using the FGF results in the section of a better feature set than when ranked with standard approaches, no matter which classifier is used for classification. The FGF's success is ascribed to its ability to incorporate both parametric as well as non-parametric data features when ranking genes as well as its ability to adapt to the specific data-set being analysed.

4 STOCHASTIC SEARCH GENE SELECTION

The feature selection approaches presented in previous chapters are based on a 'rank select' paradigm. The fundamental problem with this paradigm is that the features are organised in a particular order and once a cut-off has been determined, features that do not meet the cut-off are excluded. It is possible that the inclusion of 'non-differentially' expressed genes could result in a better feature space due to a possible non-linear relationship between the data and class distinction [112].

Presented in this chapter is an approach to gene selection based on stochastic search algorithms (SSA). SSAs have been applied extensively to the field of feature selection [112, 113]. A stochastic or 'guided' random search algorithm explores feature space for the best combination of features to be used in a classifier.

This chapter is based on a paper presented at the IEEE 26th Convention of Electrical and Electronics Engineers in Israel and published in the conference proceedings [64].

4.1 Stochastic Search Algorithms for Feature Selection

SSAs can be divided up into individual based search algorithms, such as simulated annealing, and population based search algorithms, such as Genetic Algorithm (for an overview of GA see Appendix C) [114]. In the context of feature selection, population based search algorithms have been shown to be highly successful in identifying optimal feature sets, and hence is the variant considered for gene selection [114] [101].

A typical SSA consists of three components [114] [101]: a generation/search procedure, evaluation or fitness function and stopping criteria. Generation/search involves randomly generating candidate solutions to the particular problem, which in the context of feature selection is a combination of features. A fitness function

evaluates the suitability of a candidate as a potential solution to the problem. The stopping criteria can be a predetermined number of algorithm iteration or an acceptable fitness level.

In the context of feature selection, there are two types of fitness functions: wrapper methods and filter methods [115, 116]. Filter methods are independent of the classification algorithm and are based on statistical evaluation criteria. Wrapper methods are dependent on the classifier and evaluate a feature set based on classification accuracy attained when the particular features are used to train and test a classifier.

Filter methods are computationally more efficient than wrapper methods since they do not require the training and testing of a classifier each time the algorithm is iterated [116]. On the other hand, since they are independent of classification accuracy, they suffer the risk of selecting features which are not suited for the particular classification accuracy being used.

Wrapper methods on the other hand find more suitable features since they optimise towards maximum classification accuracy [116]. The trade-off is that they are computationally expensive.

As a compromise between wrapper and filter methods, Separability Index [69] (see Chapter 2) can be used as the fitness function. On the one hand, it is similar to a wrapper method which uses a nearest neighbour classifier as its fitness function, and hence is also optimised towards maximum classification accuracy. On the other hand it is computationally inexpensive since no classifier is being trained and tested. Hence, SI is used as the fitness function for the approaches described in this chapter.

There are many different types of SSAs [117]. In this chapter, a probabilistic variant termed population based incremental learning, is considered. The results are compared to standard GA as well as to an ANOVA based ‘rank select’ approach.

4.2 Population-Based Incremental Learning (PBIL)

Population-Based Incremental Learning, akin to Genetic Algorithm (GA), was first described by Baluja [118] and attempts to integrate Evolutionary Optimisation with Competitive Learning. PBIL has outperformed GA in a number of applications, both with regards to accuracy (it does a more extensive exploration of every region of the search space) and speed to convergence (it lacks some of the complex mechanisms implemented by GA) [118].

PBIL maintains the fundamental aspects of GA [118]: a population of individuals (potential solutions) undergo iterations (generations) of 'genetic' rearrangements (mutation or recombination) in order to identify the optimal solution to a particular problem. The problem is captured by the fitness function, which assigns a fitness value to each individual, depending on its proximity to the optimum. Each individual constitutes a set of genes. In the context of feature extraction, each feature is defined as a gene. An individual can either have a gene or not (binary encoding), translating to the selection or exclusion of a particular feature.

As mentioned in Section 4.1, in the context of feature extraction, fitness is defined as being the extent of class separability exhibited by the selected features, quantified by the SI. Another possible fitness indication is classification accuracy, as advanced by Topon et.al. [119]. SI is preferred over classification accuracy since it is computationally less expensive (no classifier training is required). The primary difference between PBIL and GA is that in PBIL, the genome undergoes evolution as opposed to the individuals. This is implemented by the use of a probability vector. Each entry in the probability vector indicates the probability that a particular gene is selected for representation amongst individuals of the population. An overview of the PBIL algorithm is presented in Figure 4.1.

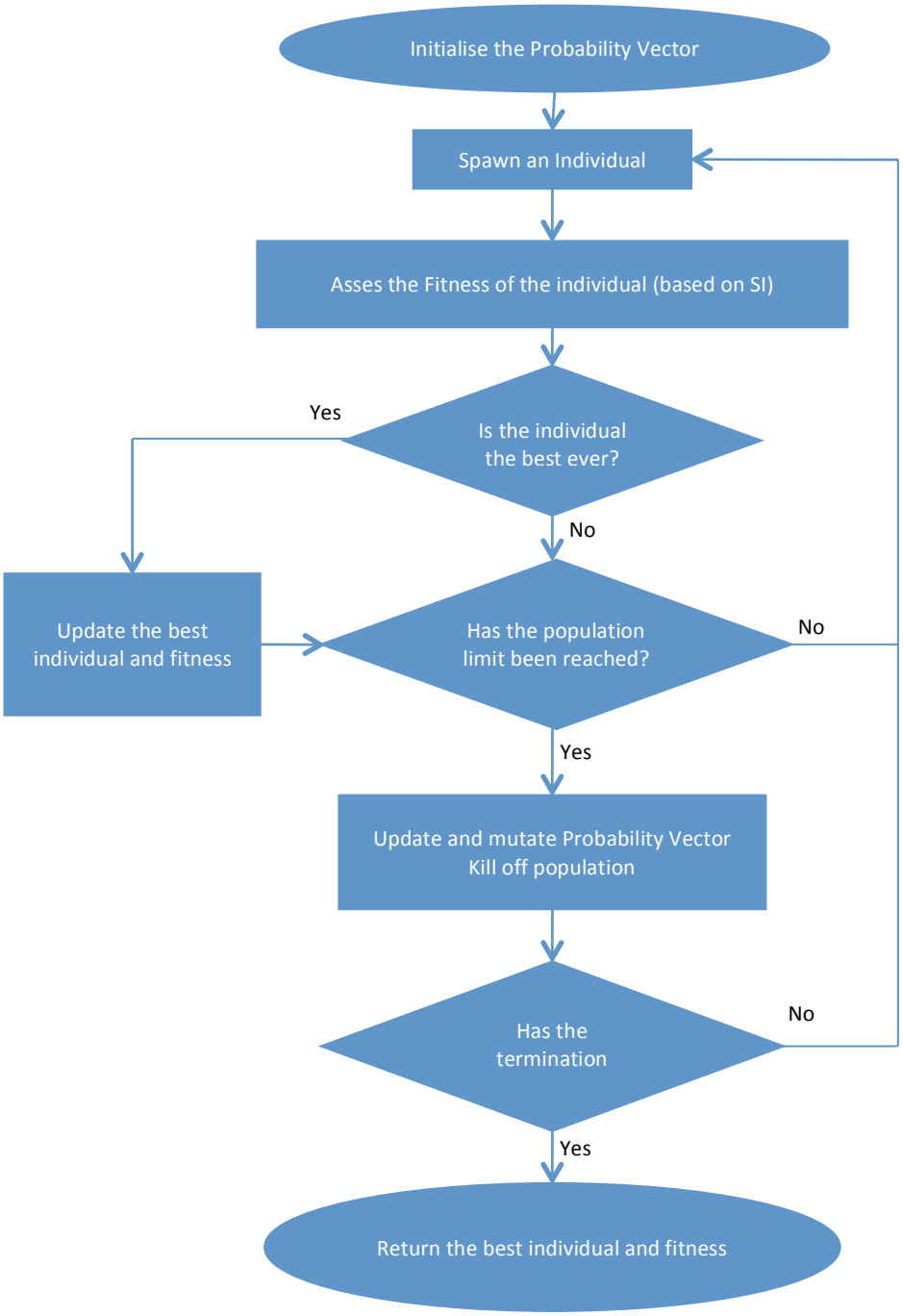


Figure 4.1: Flow Diagram of the PBIL algorithm.

An individual is assigned genes by randomly generating a vector of normally distributed numbers, with a mean of 0.5 and a standard deviation of 0.1. Each entry in the vector corresponds to a particular gene. If the number is less than the corresponding entry in the probability vector then the gene is selected to be present in the individual. At the end of each generation, the probability vector is updated. This is done by modifying each entry in the probability vector,

depending on whether its corresponding gene is present in the fittest individual ever found, in accordance with equation 4.1 [118]:

$$P(i) = P(i) \times (1 - LR) + FI(i) \times LR \quad (4.1)$$

Where:
P(i) is the i^{th} entry in the probability vector P.
LR is the learning rate.
FI(i) is the i^{th} gene of the fittest individual ever found.

The learning rate controls the influence that the fittest individual has in altering the assignment of genes amongst individuals, when they are spawned. A low learning rate means that, if a gene is present in the best individual then the corresponding entry in the probability vector will only increase slightly. The learning rate thus has an effect on how fast the algorithm converges to a solution.

In certain versions of PBIL, a negative learning rate is also implemented, in accordance with equation 4.2 [118]. This serves to steer the search away from the weakest individual [118].

$$P(i) = P(i) \times (1 + NLR) - WI(i) \times NLR \quad (4.2)$$

Where:
NLR is the negative learning rate.
WI(i) is the *i*th gene of the weakest individual ever found.

Once updated, the probability vector undergoes mutation. Mutation in the context of PBIL involves randomly reducing or increasing the probability of each entry in the probability vector, thus diversifying the genetic composition of the population [118].

Mutation, both in PBIL and in classical GA, prevents the algorithm from converging on a local optimum. Mutation is controlled by mutation probability and the mutation shift. The mutation probability specifies the probability of a gene being selected for mutation. Mutation shift indicates the extent of genetic mutation. A random number between 0 and 1 is generated. If the number is less

than the mutation probability, then the entry in the probability vector is altered as follows [118]:

$$P(i) = P(i) \times (1 - MS) + MD \times MS \quad (4.3)$$

Where:
MS is the mutation shift.
MD is the mutation direction, and can either be 0 (reduce the probability) or 1 (increase the probability).

It has been demonstrated that mutation in PBIL is not as crucial as it is in GA [118]. This is due to the probabilistic nature of individual gene allocation. Diversity is still maintained by allowing for the possibility of excluding high-probability genes and including low-probability genes, when spawning individuals. Mutation thus serves the purpose of preventing a gene probability from converging to an extreme value (0 or 1) too quickly, allowing for a more thorough search in each sector of the search space.

4.3 Implementation

All algorithms were implemented using MATLAB 7.6.0 (R2008a) on an Intel Core 2 Quad 2.4GHz PC with 3.23GB RAM. PBIL is compared to both GA and the most common approach to differential expressed gene analysis, namely ANOVA.

The algorithms were tested on a publicly available data-set [65], comprising three types of leukaemia: T-Cell Acute Lymphoblastic Leukaemia (9 samples), B-cell Acute Lymphoblastic Leukaemia (38 samples) and Acute Myelogenic Leukaemia (25 samples). Each sample consists of 5329 genes and was generated using the Affymetrix HG-U95 Human Genome Chip [57].

The PBIL and GA parameters used have been empirically determined for other binary encoded [118] problems and hence are also used in this application. PBIL was implemented using a learning rate of 0.1, a negative learning rate of 0.1, a mutation probability of 0.02 and a mutation shift of 0.05. It was empirically observed that the algorithm converged within 100 generations which was set as

the termination criterion. It was also empirically determined that a population size of 1000 is sufficient to detect individuals with maximum separability: a population of 100 could not explore enough of the search space to find individuals with as high a separability as could a population of 1000, while a population of 10000 did not find better individuals.

A binary-encoded GA was implemented using the MATLAB Genetic Algorithm and Direct Search Toolbox [120]. A stochastic uniform selection with a uniform mutation (mutation rate = 0.01) and randomly scattered crossover was used (with a crossover percentage of 80%).

Uniform mutation comprises two steps: first a fraction of the vector entries of an individual is selected for mutation, based on the mutation probability rate. The next step involves replacing each selected entry by a random number selected uniformly from the range for that entry. Uniform mutation was used due to its computational efficiency [120].

Random scatter crossover creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child. This approach allows for better diversity amongst new populations, when compared to other crossover schemes such as point crossover, and hence is preferred [101].

A population size of 10000 was used and the algorithm also terminated after 100 generations (the algorithm was tested for different population sizes and number of generations and these values were smallest population and minimum number of generations necessary to achieve maximum fitness).

Due to the stochastic nature of both PBIL and GA, and since the termination criterion for both algorithms is based on a fixed number of generations, it is possible that a different fittest individual, with a different fitness, be identified upon re-running the algorithms. Hence, average, best and worst separability values are calculated using the best individual found after 30 repeats of each algorithm.

Table 4.1: PBIL\GA\ANOVA Results Summary

Technique	Average Separability	Best Separability	Worst Separability	Av. Feature Size
PBIL	97.04%	98.61%	95.83%	326
GA	96.39%	97.22%	94.44%	2652
ANOVA (n=362)	97.22%	–	–	326
ANOVA (n=2652)	91.62%	–	–	2652
ANOVA (q<0.05)	94.44%	–	–	909
ANOVA (q<0.01)	93.06%	–	–	897
ANOVA (q<0.005)	95.83%	–	–	708
ANOVA (q<0.001)	97.22%	–	–	416

A multiple one way ANOVA (see Section 1.2.2) was implemented using the MATLAB Statistics Toolbox, while applying a False Discovery Rate correction, producing a q-value for each gene. q-value cut-offs of q < 0:05, q < 0:01 q < 0:005 and q < 0:001 are considered (due to their frequent use in microarray literature). In addition, the top NGA and NPBIL genes are also considered where NGA and NPBIL correspond to the gene set size identified by the GA and PBIL algorithms respectively.

4.4 Results and Discussion

The results for each algorithm are summarized in Table 4.1. PBIL, on average, found fitter individuals than the GA ($p<0.015$, after running both algorithms 100 times). It also managed to consistently find better feature-spaces than the ANOVA for n = 2652, q < 0,05, q < 0:01 and q < 0,005. The best PBIL run resulted in only a single instance with a nearest neighbour not belonging to its own class while the GA's best run resulted in two instances. The best PBIL run even outperformed ANOVA for q < 0:001 and n = 326.

While still allowing for diversity due to mutation, PBIL is more efficient in excluding redundant features than standard GA, as is evident by the smaller feature sizes identified. This is preferable [61] since redundant features are excluded from being used for classification. It is suspected that the PBIL's rejection of redundant features is based on the implementation of negative learning when updating the probability vector. If a feature is present in the best individual but not in the worst, as is indicative of a differentially expressed gene,

then the probability of selecting that feature increases every generation. If a feature is present in the worst individual but not in the best one, indicating that the feature has an adverse effect on class separability and is hence redundant, then the probability of selecting that feature decreases. If, however, a feature is present in both the best and the worst individual then the probability remains unaltered (since the learning rate equals the negative learning rate). GA has no explicit mechanism of excluding redundant features, whereas PBIL explicitly excludes redundant features.

Figure 4.2 depicts how the fitness of the best individual varies across the generations, for the best run of the PBIL (solid line) and Genetic Algorithms (dashed line) respectively. The best Individual found by the PBIL was identified after 87 generations. The best individual found by the GA was found after 13 generations.

The PBIL's slower convergence indicates its ability to implement a more diverse search of the problem space than the GA. GA has a higher tendency to converge prematurely, causing it to converge to suboptimal solutions. This could be remedied by increasing the mutation rate. On the other hand, due to its probabilistic nature in spawning individuals, PBIL maintains diversity irrespective of the mutation rate [118], allowing for better exploration of the search space.

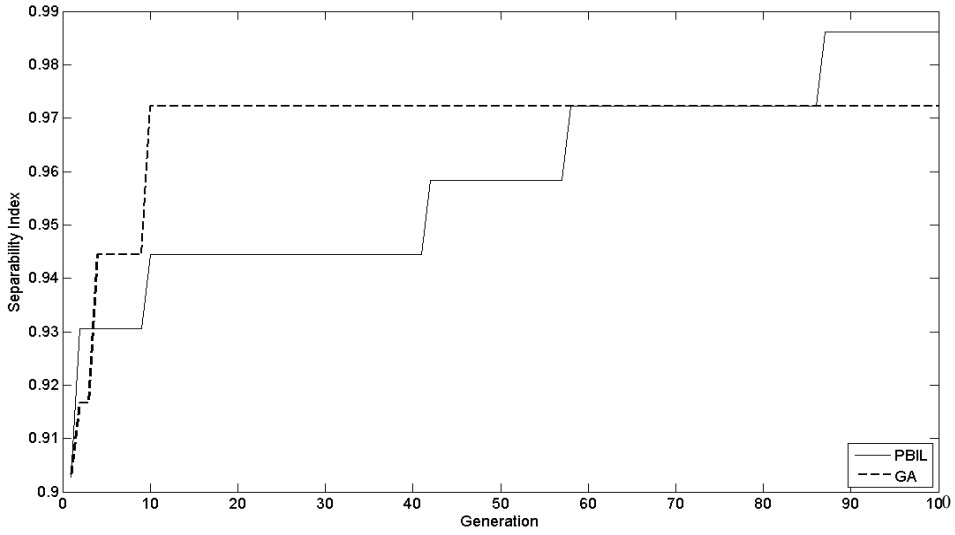


Figure 4.2: Fitness variation of the fittest individual across all generations for the PBIL (solid line) and GA (dashed line). The algorithms converged after 87 generations and 10 generations for the PBIL and GA respectively.

4.5 Conclusion

The identification of a differentially expressed gene set is central to microarray data analysis, both with regards to pathway identification and the formation of a suitable classifier feature-space for cancer classification. The effectiveness of the PBIL algorithm, in identifying an optimal classification feature-space, was tested and compared to that of regular GA and ANOVA gene selection techniques. PBIL involves iteratively probabilistically evolving the genome of a search population. A Separability Index was used to guide the algorithms through the search-space, comprising various combinations of features.

The PBIL algorithm outperformed regular GA by identifying a feature-space which yielded, on average, a higher class-separability (97.04% for PBIL and 96.39% for the GA, with $p<0.015$ after re-running each algorithm 100 times) and a fewer number of genes (PBIL - 326 genes, GA - 2652). It also, on average, outperformed the ANOVA approach for $n = 2652$ (91.62%), $q < 0,05$ (94.44%), $q < 0,01$ (93.06%) and $q < 0,005$ (95.83%). The best PBIL run (98.61%) was even able to find a better feature set than the ANOVA for $n = 326$ and $q < 0,001$ (both 97.22%).

The performance of the PBIL is ascribed to its ability to steer the search away from the worst individuals, allowing for the exclusion of redundant features.

5 CONCLUSION AND RECOMMENDATIONS

Three novel approaches to microarray data feature selection and differential expression analysis have been presented in this thesis:

- Gene selection based on Separability Index
- Gene Ranking using the Fuzzy Gene Filter
- Gene selection using Population Based Incremental Learning

These approaches address the three research questions outlined in Section 1.3, dealing with the arbitrary nature of standard techniques by provide a more holistic approach to microarray feature selection and differential expression analysis.

By using SI, a more data intrinsic approach to differentially expressed gene selection was implemented, attaining better results than standard differential expression analysis, while still maintaining functional enrichment. The approach achieved a K-NN classifier accuracy of 92% on a test data set, comprising breast, colon and lung cancer microarray data.

The FGF provides a holistic approach to gene ranking incorporating both parametric and non-parametric features. It is also optimised for the particular data set under scrutiny. Genes ranked by the FGF attained significantly higher accuracies for all of the classifiers tested, on both data sets ($p < 0.0231$ for the prostate data set and $p < 0.1888$ for the lymphoma data set). When using the prostate data set, the FGF performed best on the KNN classifier, achieving an accuracy of 96.1% with the top 9 ranking genes. When using the lymphoma data set, the FGF performed best on the SVM classifier, achieving an accuracy of 100% with the top 12 ranking genes.

The performance of the FGF is attributed to the fact that it is optimised to rank genes in such a way that results in maximum class separability, as well as its incorporation of multiple features of the data when ranking genes.

As a possible future improvement, due to its flexibility, the FGF can incorporate biological knowledge associated with the particular gene being ranked. *Apriori* associations of a particular gene to the disease under discussion can also be taken into account when ranking. This could be incorporated as an additional input variable and the fuzzy rules could accommodate the associations.

PBIL has been demonstrated to find better features (average SI = 97.04%) than GA (average SI = 96.39%), as well as ANOVA (SI = 94.44%), identifying features which collectively have a higher SI.

The PBIL algorithm can be also improved by incorporating a functional enrichment estimate of the selected gene set. This could possibly be incorporated in the fitness function.

Thus, the research questions presented in Section 1.3 have been addressed: less arbitrary, more holistic approaches to microarray gene selection have been developed, tested and presented while maintaining and even surpassing the performance of gold standard approaches.

REFERENCES

[1] R. K. Curtisa, M. Oresicb, and A. Vidal-Puiga, "Pathways to the analysis of microarray data," *Trends in Biotechnology,* vol. 23, pp. 429-435, 2005.

[2] X. Cui and G. A. Churchill, "Statistical Tests for Differential Expression in cDNA Microarray Experiments," *Genome Biology,* vol. 4, p. 210, 2003.

[3] W. B. Coleman and G. J. Tsongalis, *Molecular Diagnostics For the Clinical Laboratorian*. New Jersey, USA: Humana Press, 2006.

[4] S. Kobayashi, F. Kimura, T. Ikeda, Y. Osawa, H. Torikai, A. Kobayashi, K. Sato, and K. Motoyoshi, "BCR-ABL promotes neutrophil differentiation in the chronic phase of chronic myeloid leukemia by downregulating c-Jun expression," *Leukemia,* vol. 23, pp. 1622-1627, 2009.

[5] L. Dongguang, "DNA Microarray Expression Analysis and Data Mining for Blood Cancer," in *Future BioMedical Information Engineering, 2008. FBIE '08. International Seminar on*, 2008, pp. 377-381.

[6] D. Grimwade and T. Haferlach, "Gene-Expression Profiling in Acute Myeloid Leukemia," *New England Journal of Medicine,* vol. 350, pp. 1676-1678, 2004.

[7] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science,* vol. 286, pp. 531-537, 1999.

[8] M. E. Ross, X. Zhou, G. Song, S. A. Shurtleff, K. Girtman, W. K. Williams, H.-C. Liu, R. Mahfouz, S. C. Raimondi, N. Lenny, A. Patel, and J. R. Downing, "Classification of pediatric acute lymphoblastic leukemia by gene expression profiling," *Blood,* vol. 102, pp. 2951-2959, 2003.

[9] L. C. Dorssers, T. van Agthoven, A. Brinkman, J. Veldscholte, and M. S. K. J. Dechering, "Breast cancer oestrogen independence mediated by BCAR1 or BCAR3 genes is transmitted through mechanisms distinct from the oestrogen receptor signalling pathway or the epidermal growth factor receptor signalling pathway.," *Breast Cancer Research,* vol. 7, pp. 82-92, 2005.

[10] D. Singh, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C, Tamayo P, Renshaw AA, D'Amico AV, Richie JP, Lander ES, Loda M, Kantoff PW, Golub TR, S. WR., and e. al., "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell,* pp. 203-209, 2002.

[11] N. Servant, E. Gravier, P. Gestraud, C. Laurent, C. Paccard, A. Biton, I. Brito, J. Mandel, B. Asselain, E. Barillot, and P. Hupe, "EMA - A R package for Easy Microarray data analysis," *BMC Research Notes,* vol. 3, p. 277, 2010.

[12] D. M. Rocke, T. Ideker, O. Troyanskaya, J. Quackenbush, and J. Dopazo, "Papers on normalization, variable selection, classification or clustering of microarray data," *Bioinformatics,* vol. 25, pp. 701-702, March 15, 2009.
[13] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data," *Bioinformatics,* vol. 26, pp. 139-140, January 1, 2010.
[14] E. Glaab and R. Schneider, "PathVar: analysis of gene and protein expression variance in cellular pathways using microarray data," *Bioinformatics,* November 28, 2011.
[15] "GeneSpring GX 10.0 ", 10 ed: Agilent, 2008.
[16] "Statistical Algorithms Description Document," Affymetrix http://media.affymetrix.com/support/technical/whitepapers/sadd_whitepaper.pdf 2002.
[17] R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed, "Summaries of Affymetrix GeneChip probe level data," *Nucl. Acids Res.,* vol. 31, pp. e15-, February 15, 2003.
[18] Z. Wu and R. A. Irizarry, "Preprocessing of oligonucleotide array data," *Nat Biotech,* vol. 22, pp. 656-658, 2004.
[19] B. M. Bolstad, R. A. Irizarry, M. Åstrand, and T. P. Speed, "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias," *Bioinformatics,* vol. 19, pp. 185-193, January 22, 2003.
[20] T. J. Mariani, V. Budhraja, B. H. Mecham, C. C. Gu, M. A. Watson, and Y. Sadovsky, "A variable fold change threshold determines significance for expression microarrays.," *FASEB,* vol. 17, pp. 321-323, 2003.
[21] M. K. Kerr, M. Mitchell, and A. C. Gary, "Analysis of Variance for Gene Expression Microarray Data," *Journal of Computational Biology,* vol. 7, pp. 819-837, 2000.
[22] Student, "The Probable Error of a Mean," *Biometrika,* vol. 6, pp. 1-25, March 1, 1908.
[23] T. Yang, V. Kecman, L. Cao, and C. Zhang, "Combining Support Vector Machines and the <i>t</i>-statistic for Gene Selection in DNA Microarray Data Analysis Advances in Knowledge Discovery and Data Mining." vol. 6119, M. Zaki, J. Yu, B. Ravindran, and V. Pudi, Eds., ed: Springer Berlin / Heidelberg, 2010, pp. 55-62.
[24] C.-P. Lee and Y. Leu, "A novel hybrid feature selection method for microarray data analysis," *Applied Soft Computing,* vol. 11, pp. 208-213, 2011.
[25] M. Mohamad, S. Omatu, S. Deris, M. Misman, and M. Yoshioka, "Selecting informative genes from microarray data by using hybrid methods for cancer classification," *Artificial Life and Robotics,* vol. 13, pp. 414-417, 2009.
[26] R. A. Fisher, "The correlation between relatives on the supposition of mendelian inheritence," *Transactions of the Royal Society of Edinburgh,* vol. 52, pp. 399-433, 1918.
[27] O. G. Troyanskaya, M. E. Garber, P. O. Brown, D. Botstein, and R. B. Altman, "Nonparametric methods for identifying differentially expressed

genes in microarray data," *Bioinformatics,* vol. 18, pp. 1454-1461, November 1, 2002.
[28] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin,* vol. 1, pp. 80-83, 1945.
[29] J. A. Hanley and B. J. McNeil, "A method of comparing the areas under receiver operating characteristic curves derived from the same cases," *Radiology,* vol. 148, pp. 839-843, September 1, 1983.
[30] T. Fawcett, "An introduction to ROC analysis," *ROC Analysis in Pattern Recognition,* vol. 27, pp. 861-874, 2006.
[31] J. Swets, "Measuring the accuracy of diagnostic systems," *Science,* vol. 240, pp. 1285-1293, June 3, 1988.
[32] H. Mamitsuka, "Selecting features in microarray classification using ROC curves," *Pattern Recogn.,* vol. 39, pp. 2393-2404, 2006.
[33] B. Hanczar, J. Hua, C. Sima, J. Weinstein, M. Bittner, and E. R. Dougherty, "Small-sample precision of ROC-related estimates," *Bioinformatics,* vol. 26, pp. 822-830, March 15, 2010.
[34] D. Hand, "Measuring classifier performance: a coherent alternative to the area under the ROC curve," *Machine Learning,* vol. 77, pp. 103-123, 2009.
[35] D. B. Allison, X. Cui, G. P. Page, and M. Sabripour, "Microarray data analysis: from disarray to consolidation and consensus," *Nat Rev Genet,* vol. 7, pp. 55-65, 2006.
[36] W. Pan, "A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments," *Bioinformatics,* vol. 18, pp. 546-554, April 1, 2002.
[37] S. Dudoit, "Multiple Hypothesis Testing in Microarray Experiments," *UC Berkley Division of Biostatistics Working Paper Series,* 2002.
[38] A. Hackstadt and A. Hess, "Filtering for increased power for microarray data analysis," *BMC Bioinformatics,* vol. 10, p. 11, 2009.
[39] J. D. Storey and R. Tibshirani, *Statistical methods for identifying differentially expressed genes in DNA microarrays* vol. 224. Clifton, NJ, ETATS-UNIS: Humana Press, 2003.
[40] Y. Benjamini and Y. Hochberg, *Controlling the false discovery rate: a practical and powerful approach to multiple testing* vol. 57. London, ROYAUME-UNI: Royal Statistical Society, 1995.
[41] R. Feise, "Do multiple outcome measures require p-value adjustment?," *BMC Medical Research Methodology,* vol. 2, p. 8, 2002.
[42] J. K. Choi, U. Yu, O. J. Yoo, and S. Kim, "Differential coexpression analysis using microarray data and its application to human cancer," *Bioinformatics,* vol. 21, pp. 4348-4355.
[43] H. K. Lee, A. K. Hsu, J. Sajdak, J. Qin, and P. Pavlidis, "Coexpression Analysis of Human Genes Across Many Microarray Data Sets," *Genome Research,* vol. 14, pp. 1085-1094, June 1, 2004 2004.
[44] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 95, pp. 14863-14868, December 8, 1998.

[45] N. Belacel, Q. Wang, and M. Cuperlovic-Culf, "Clustering Methods for Microarray Gene Expression Data," *OMICS: A Journal of Integrative Biology,* vol. 10, pp. 507-531, 2006.

[46] F. Al-Shahrour. *FatiGO*. Available: http://bioinfo.cipf.es/babelomicswiki/tool:fatigo

[47] P. Pavlidis, J. Qin, V. Arango, J. J. Mann, and E. Sibille, "Using the gene ontology for microarray data mining: a comparison of methods and application to age effects in human prefrontal cortex.," *Neurochem Res,* vol. 29, pp. 1213-22, 2004.

[48] K. O'Neill, A. Garcia, A. Schwegmann, R. C. Jimenez, and D. J. H. Hermjakob, "OntoDas - a tool for facilitating the construction of complex queries to the Gene Ontology.," *BMC Bioinformatics,* vol. 437, p. 9, 2008.

[49] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene Ontology: tool for the unification of biology," *Nature Genetics,* vol. 25, pp. 25-29, 2000.

[50] S. Falcon and R. Gentleman, "Using GOstats to test gene lists for GO term association," *Bioinformatics,* vol. 23, pp. 257-258, January 15, 2007 2007.

[51] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 102, pp. 15545-15550, October 25, 2005.

[52] R. K. van Laar, X.-J. Ma, D. de Jong, D. Wehkamp, A. N. Floore, M. O. Warmoes, I. Simon, W. Wang, M. Erlander, L. J. van't Veer, and A. M. Glas, "Implementation of a novel microarray-based diagnostic test for cancer of unknown primary," *International Journal of Cancer,* vol. 125, pp. 1390-1397, 2009.

[53] M. J. Lodes, M. Caraballo, D. Suciu, S. Munro, A. Kumar, and B. Anderson, "Detection of Cancer with Serum miRNAs on an Oligonucleotide Microarray," *PLoS ONE,* vol. 4, p. e6229, 2009.

[54] M. A. Shipp, K. N. Ross, and T. R. Golub, "Diffuse large B-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning," *Nature Medicine,* pp. 68-74, 2002.

[55] T. Jirapech-Umpai and S. Aitken, "Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes," *BMC Bioinformatics,* vol. 6, p. 148, 2005.

[56] A. Kelemen, Z. Hong, P. Lawhead, and L. Yulan, "Naive Bayesian Classifier for Microarray Data," *Proceedings of the International Joint Conference on Neural Networks.,* vol. 3, pp. 1769-1773 2003.

[57] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "A Comprehensive Evaluation of Multi-category Classification Methods for Microarray Gene Expression Cancer Diagnosis," *Bioinformatics,* vol. 25, pp. 631-643, 2005.

[58] J. W. F. Catto, M. F. Abbod, P. J. Wild, D. A. Linkens, C. Pilarsky, I. Rehman, D. J. Rosario, S. Denzinger, M. Burger, R. Stoehr, R. Knuechel, A. Hartmann, and F. C. Hamdy, "The Application of Artificial Intelligence to Microarray Data: Identification of a Novel Gene Signature to Identify Bladder Cancer Progression," *European Urology,* vol. 57, pp. 398-406, 2010.

[59] J. Khan, J. S. Wei, M. Ringnér, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine,* vol. 7, pp. 673-679, 2001.

[60] R. M. Parry, W. Jones, T. H. Stokes, J. H. Phan, R. A. Moffitt, H. Fang, L. Shi, A. Oberthuer, M. Fischer, W. Tong, and M. D. Wang, "k-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction," *The Pharmacogenomics Journal,* vol. 10, pp. 292-309, 2010.

[61] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cambridge UK: Springer, 2006.

[62] M. Perez, J. Featherston, D. M. Rubin, T. Marwala, L. E. Scott, and W. Stevens, "Differentially Expressed Gene Identification based on Separability Index," *Proceedings of the Eighth International Conference on Machine Learning and Applications,* pp. 429-434, 2009.

[63] M. Perez, D. M. Rubin, T. Marwala, L. E. Scott, J. Featherston, and W. Stevens, "The Fuzzy Gene Filter: An Adaptive Fuzzy Inference System for Expression Array Feature Selection," in *Trends in Applied Intelligent Systems* vol. 6098, ed. Heidelberg: Springer, 2010, pp. 62-71.

[64] M. Perez, D. M. Rubin, T. Marwala, L. E. Scott, J. Featherston, and W. Stevens, "A Population-Based Incremental Learning Approach to Microarray Gene Expression Feature Selection," *Proceedings of the IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, Eilat, Israel, 2010 pp. 000010 - 000014.

[65] A. Statnikov. *GEMS: Gene Expression Model Selector*. Available: http://www.gems-system.org

[66] *The Eighth International Conference on Machine Learning and Applications* Available: http://www.icmla-conference.org/icmla09/

[67] MathWorks, "Statistics Toolbox," in *MATLAB*, ed, 2007.

[68] MathWorks, "Bioinformatics Toolbox," in *MATLAB*, ed, 2007.

[69] C. Thornton, *Truth from Trash: How Learning Makes Sense*. Cambridge UK: MIT Press, 2002.

[70] D. Zighed, S. Lallich, and F. Muhlenbach, "Separability Index in Supervised Learning," *Principles of Data Mining and Knowledge Discovery,* vol. 2431, pp. 241-267, 2002.

[71] L. Mthembu and T. Marwala, "A note on the separability index," in *Proceedings of the Nineteenth Annual Symposium of the Pattern Recognition Association of South Africa*, 2007, pp. 45-48.

[72] A. P. Wu, D. A. Largespada, A. Vortkam, S. Scherer, N. G. Copeland, N. A. Jenkins, G. Bruns, and K. Georgopoulos, "The lkaros Gene Encodes a Family of lymphocyte-Restricted Zinc Finger DNA Binding Proteins,

Highly Conserved in Human and Mouse," *The journal of Immunology,* vol. 156, pp. 585-592, 1996.

[73] A. Gidudu and A. Heinz, "Comparison of Feature Selection Techniques for SVM Classification," *Proceedings of the 10th International Symposium of Physical Measurements and Signatures in Remote Sensing,* vol. XXXVII, 2005.

[74] G. Unger and B. Chor, "Linear Separability of Gene Expression Data Sets," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 7, pp. 375 - 381, 2010.

[75] I. Costa, A. Lorena, L. Peres, and M. de Souto, "Using Supervised Complexity Measures in the Analysis of Cancer Gene Expression Data Sets," in *Advances in Bioinformatics and Computational Biology*. vol. 5676, K. Guimarães, A. Panchenko, and T. Przytycka, Eds., ed: Springer Berlin / Heidelberg, 2009, pp. 48-59.

[76] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory,* vol. 13, pp. 21-27, 1967.

[77] M. F. T. Pas, I. Hulsegge, A. A. Coster, M. H. Pool, H. H. Heuven, and L. L. Janss, "Biochemical pathways analysis of microarray results: regulation of myogenesis in pigs.," *BMC Dev Biol,* vol. 7, p. 66, 2007.

[78] P. S. Yan, C. M. Chen, H. Shi, F. Rahmatpanah, S. H. Wei, C. W. Caldwell, and T. H. Huang, "Dissecting complex epigenetic alterations in breast cancer using CpG island microarrays.," *Cancer Res,* vol. 61, pp. 8375-80, 2001.

[79] J. J. Park, R. A. I. G. Buchanan, S. S. Koh, J. M. Park, W. D. Tilley, and M. R. S. M. F. P. G. A. Coetzee, "Breast cancer susceptibility gene 1 (BRCAI) is a coactivator of the androgen receptor.," *Cancer Res,* vol. 60, pp. 5946-9, 2000.

[80] F. G. Giancotti and E. Ruoslahti, "Integrin Signaling," *Science,* vol. 285, pp. 1028-1032, 1999.

[81] A. M. Mercurio and I. Rabinovitz, "Towards a mechanistic understanding of tumor invasion-lessons from the alpha6beta 4 integrin.," *Semin Cancer Biol,* vol. 11, pp. 129-41, 2001.

[82] L. B. Saltz, N. J. Meropol, P. J. Loehrer, M. N. Needle, J. Kopit, and R. J. Mayer, "Phase II trial of cetuximab in patients with refractory colorectal cancer that expresses the epidermal growth factor receptor.," *J Clin Oncol,* vol. 22, pp. 1201-8, 2004.

[83] A. E. Bale and K. P. Yu, "The hedgehog pathway and basal cell carcinomas.," *Hum Mol Genet,* vol. 10, pp. 757-62, 2001.

[84] S. P. Tabruyn and A. W. Griffioen, "A new role for NF-kappaB in angiogenesis inhibition.," *Cell Death Differ,* vol. 14, pp. 1393-7, 2007.

[85] K. Oguma, H. Oshima, M. Aoki, R. Uchio, K. Naka, S. Nakamura, A. Hirao, H. Saya, M. M. Taketo, and M. Oshima, "Activated macrophages promote Wnt signalling through tumour necrosis factor-alpha in gastric tumour cells.," *Embo J,* vol. 27, pp. 1671-81, 2008.

[86] G. G. Mullighan, C. B. Miller, I. Radtke, L. A. Phillips, J. Dalton, J. Ma, D. White, T. P. Hughes, M. M. L. Beau, C. H. Pui, M. V. Relling, S. A. Shurtleff, and J. R. Downing, "BCR-ABL1 lymphoblastic leukaemia is characterized by the deletion of Ikaros," *Nature,* vol. 453, pp. 110-4, 2008.

[87] L. A. Zadeh, "Fuzzy Sets," *Information Control,* vol. 8, pp. 338-353, 1965.
[88] L. A. Zadeh, "Fuzzy sets as a basis for theory of possibility," *Fuzzy Sets and Systems,* vol. 1, pp. 3-28, 1978.
[89] M. Perez, D. M. Rubin, T. Marwala, L. E. Scott, and W. Stevens, "A Hybrid Fuzzy-SVM Classifier, Applied to Gene Expression Profiling for Automated Leukaemia Diagnosis " in *IEEE 25-th Convention of Electrical and Electronics Engineers in Israel*, Eilat, Israel, 2008.
[90] R. Hirakura and M. Oh, "Fuzzy Inference System," United States Patent, 1995.
[91] *Applications of Fuzzy Logic* Available: http://www.dementia.org/~julied/logic/applications.html
[92] S. Kurnaz, O. Cetin, and O. Kaynak, "Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles," *Expert Systems with Applications,* vol. 37, pp. 1229-1234, 2010.
[93] Ü. Elif Derya, "Adaptive neuro-fuzzy inference system for classification of ECG signals using Lyapunov exponents," *Computer Methods and Programs in Biomedicine,* vol. 93, pp. 313-321, 2009.
[94] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets and Systems,* vol. 45, pp. 135-156, 1992.
[95] O. Almeida, L. Reis, L. Bezerra, and S. Lima, "A MIMO Fuzzy Logic Autotuning PID Controller: Method and Application," in *Applied Soft Computing Technologies: The Challenge of Complexity*. vol. 34, A. Abraham, B. de Baets, M. Köppen, and B. Nickolay, Eds., ed: Springer Berlin / Heidelberg, 2006, pp. 569-580.
[96] J. H. Kim and S. J. Oh, "A fuzzy PID controller for nonlinear and uncertain systems," *Soft Computing - A Fusion of Foundations, Methodologies and Applications,* vol. 4, pp. 123-129, 2000.
[97] *Samsung WA10U3 10kg double storm top loader*. Available: http://www.samsung.com/za/consumer/home-appliances/washing-machine/top-loader/WA10U3WIP/XFA/index.idx?pagetype=prd_detail
[98] M. Perez, R. O. Davidson, D. M. Rubin, and T. Marwala, "Simulation of retinal function A fuzzy-linear approach," in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, 2008, pp. 1079-1084.
[99] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies,* vol. 7, pp. 1-13, 1975.
[100] W. Huber, A. von Heydebreck, A. Homaifar, and E. McCormick, "Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms," *IEEE Transactions on Fuzzy Systems,* vol. 3, pp. 299-315, 1995.
[101] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*: Addison-Wesley, 1989.
[102] *Real-world uses of Genetic Algorithm*. Available: http://brainz.org/15-real-world-applications-genetic-algorithms/
[103] F. Gao, Q. Liu, R. Shan, and H. Zhang, "Optimal design of smart antenna array," *Journal of Electronics (China),* vol. 21, pp. 342-345, 2004.

[104] J.-S. Chen and J.-L. Hou, "A Combination Genetic Algorithm with Applications on Portfolio Optimization," in *Advances in Applied Artificial Intelligence*. vol. 4031, M. Ali and R. Dapoigny, Eds., ed: Springer Berlin / Heidelberg, 2006, pp. 197-206.
[105] N. A. Barricelli, "Numerical testing of evolution theories. Part II. Preliminary tests of performance, symbiogenesis and terrestrial life," *Acta Biotheoretica* vol. 16, pp. 99–126, 1963.
[106] J. Holland, *Adaptation in Natural and Artificial Systems*: MIT Press, 1975.
[107] H. Braun, "On solving travelling salesman problems by genetic algorithms," in *Parallel Problem Solving from Nature*. vol. 496, H.-P. Schwefel and R. Männer, Eds., ed: Springer Berlin / Heidelberg, 1991, pp. 129-133.
[108] T. Yamada and R. Nakano, "Genetic Algorithms for Job-Shop Scheduling Problems," in *Modern Heuristic for Decision Support UNICOM seminar*, London, 1997, pp. 67-81.
[109] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *Intelligent Systems and their Applications, IEEE,* vol. 13, pp. 44-49, 1998.
[110] L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen, "Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method," *Bioinformatics,* vol. 17, pp. 1131-1142, December 1, 2001.
[111] F. Herrera, M. Lozano, and J. L. Verdegay, "Tuning Fuzzy Logic Controllers by Genetic Algorithms," *International Journal of Approximate Reasoning,* pp. 299-315, 1995.
[112] I. Guyon and A. Eliseef, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.
[113] H. Lui and L. Yu, "Toward Integrating Feature Selection Selection Algorithms for Classification and Clustering," *IEEE Transactions on Knowledge Data Engineering,* vol. 17, pp. 491-501, 2005.
[114] J. C. Spall, *Introduction to stochastic search and optimization: estimation, simulation, and control* vol. 64 Canada: John Wiley and Sons, 2003.
[115] R. Kohavi and G. H. John, "Wrapper for Feature Subset Selection," *Artificial Intelligence* vol. 97, pp. 273-324, 1997.
[116] Z. Zhu, S. Jia, and Z. Ji. (2010) Towards a Memetic Feature Selection Paradigm. *IEEE Computational Intelligence* 41-53.
[117] R. E. Korf, "Artificial intelligence search algorithms," in *Algorithms and theory of computation handbook*, J. A. Mikhail and B. Marina, Eds., ed: Chapman \& Hall/CRC, 2010, pp. 22-22.
[118] S. Baluja, "Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning," Carnegie Mellon University1994.
[119] P. K. Topon and I. Hitoshi, "Selection of the Most Useful Subset of Genes for Gene Expression-Based Classification," *Proceedings of the 2005 conference on Genetic and evolutionary computation,* pp. 453-460 2005.
[120] MathWorks, "Genetic Algorithm and Direct Search Toolbox," in *MATLAB*, ed, 2007.

[121] S. Piramuthu, M. J. Shaw, and J. A. Gentry, "A classification approach using multi-layered neural networks," *Decision Support Systems,* vol. 11, pp. 509-525, 1994.
[122] F. Matera, "Radial Basis Function Neural Network," *Substance Use & Misuse,* vol. 33, pp. 317-334, 1998.
[123] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning,* vol. 20, pp. 273-297, 1995.
[124] D. K. Slonim, "From patterns to pathways: gene expression data analysis comes of age," *Nature Genetics,* vol. 32, pp. 502 - 508, 2002.
[125] B. L. Vrusias. (2005). *Introduction to Fuzzy Logic*. Available: http://portal.surrey.ac.uk/pls/portal/url/item/02629fe9060d01d1e0440003ba296bde

APPENDIX A SUPERVISED CLASSIFIERS FOR MICROARRAY DATA CLASSIFICATION

Presented in this appendix is a brief overview of the most popular supervised classification algorithms used for microarray data classification [57]. Three types of classifiers are discussed: Artificial Neural Networks (ANN), both the Multi Layered Perceptron (MLP) and Radial Basis Function (RBF) configurations; Support Vector Machine (SVM) and K-Nearest Neighbour (KNN).

A.1 Multi-Layered Perceptron Neural Network

The Multi-Layered Perceptron (MLP) is the most fundamental form of neural network [121]. A MLP is a feed forward neural network that consists of various layers or sets of nodes.

A MLP consists of a complex network of neurons, as depicted in Figure A.1. Neurons form connections between nodes. Each neuron stores knowledge in the form of a connection strength known as a weight. A weight describes the affect a particular node has on the node to which it is connected. A node stores knowledge in the form of a bias (a value added to the inputs at the node).

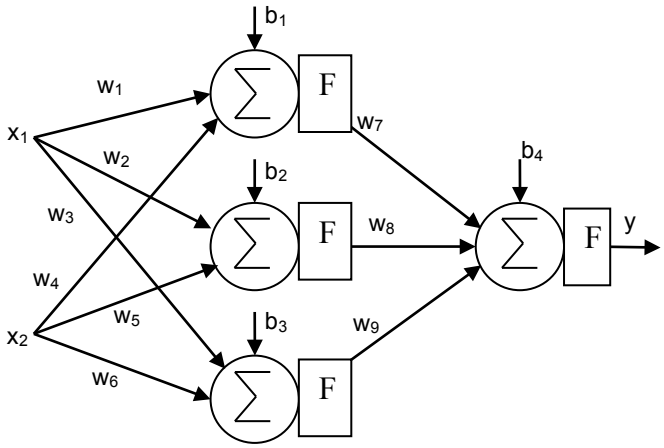


Figure A.1: A Three layered Multilayer Perceptron.

A MLP consists of various layers or sets of nodes, as depicted in Figure A.1. The first layer is the input layer, to which input data is presented. The input layer is connected to the hidden layers which are connected to the output layer, all via weighted neurons.

All the inputs to a node are summed and transformed to the output via an activation function (symbolised by the F blocks in Figure A.1). The MLP is trained via back-propagation by presenting it a portion of the input data and comparing the outputs of the network to the target outputs, iteratively adjusting the weights and biases until the MLP's outputs approximate the targeted outputs. A number of optimisation algorithms can be used to optimise the weights of the MLP, the most efficient being Scaled Conjugate Gradient (SCG).

A.2 Radial Basis Function Neural Network

The Radial Basis Function (RBF) [122] Neural Network has shown to be quicker to train than the MLP. This is due to the fact that it incorporates unsupervised clustering techniques in its training.

A RBF Neural Network is a type of neural network which has three layers of nodes: an input layer, a hidden layer, which implements non-linear RBFs, and an output layer. A RBF can take on various types of distributions. The Gaussian distribution is the most common. The mean of the RBF is referred to as its centre or centroid.

The training of a RBF neural network takes place in two stages. First, the centres of the RBFs are determined using k-means clustering: the input training data are arranged in a vector space, as depicted in Figure A.2.

Figure A.2 depicts a simple 2 dimensional vector space and each point (white oval) in the space corresponds to a data point. k centres (black ovals) are randomly assigned to the vector space, where k is simply the number of hidden nodes.

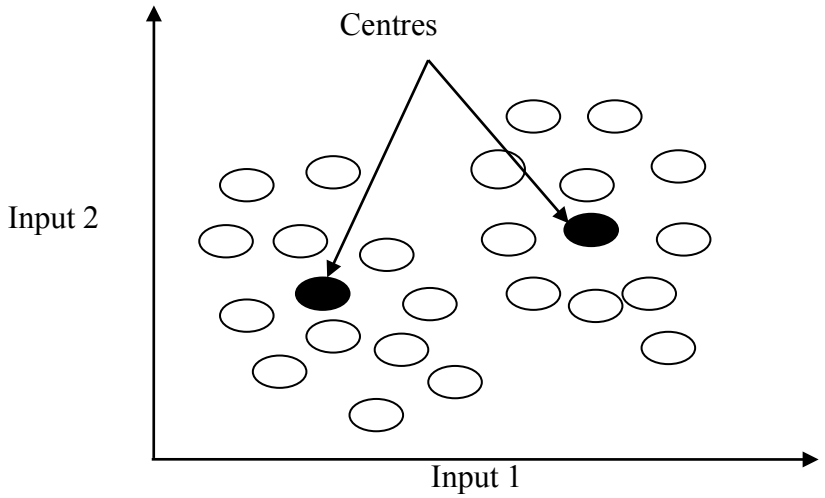


Figure A.2: Data (white ovals) distributed in a 2 dimensional vector space, showing 2 cluster centres (black ovals).

Each data point is assigned to the nearest centre, forming a set of k clusters. The centres then move to the centre of their respective clusters. The data points are then reassigned to the nearest centre and the process is repeated until the centres remain constant.

The second part of RBF neural network training involves supervised training of the weights connecting the hidden nodes to the output node. This is generally done via least square regression.

The output activation function of the RBF neural network could be linear or sigmoidal. For classification problems, a sigmoidal activation function is generally used but a linear output could also be used if the outputs are rounded to 1 or 0.

After training, the RBF is validated by determining the number of hidden nodes which produce the most accurate results and hence the optimal number of centres.

When an unclassified data point is presented to the input layer of the RBF, it is passed to the hidden layer. The output value of each hidden node corresponds to the extent that the data point belongs to that node's corresponding cluster (depending on where along the distribution the point lies). These values propagate to the output layer and the data point is assigned to a particular class.

A.3 Support Vector Machines

Support Vector Machines (SVM), originally developed by Vapnik et.al. in the mid 1990's [123], are hard, non-parametric, robust classifiers, normally trained using supervised learning. The Support Vector Machine (SVM) is considered to be one of the most significant developments in Artificial Intelligent classification in recent years. The SVM's insensitivity to a high dimensional input space makes it an ideal candidate for classification of GEP [57].

Like RBFs, SVMs operate in vector space. The classified input data is vectorised, as depicted in the simplified 2 dimensional vector space in Figure A.3.

During training, a discriminant function, or decision boundary, is generated to separate between the two classes. A margin between the discriminatory function and the nearest data points or vectors is then generated, as depicted in Figure A.3. The vectors which result in the largest margin are referred to as support vectors. Support vectors are identified through quadratic programming and Lagrange Multipliers. During cross-validation, the extent of influence of outliers is determined. Outliers can result in a decision boundary with a smaller margin, resulting in suboptimal classification accuracy. Therefore, an upper-bound constant, normally symbolised by C, is defined in order to limit the influence of outliers.

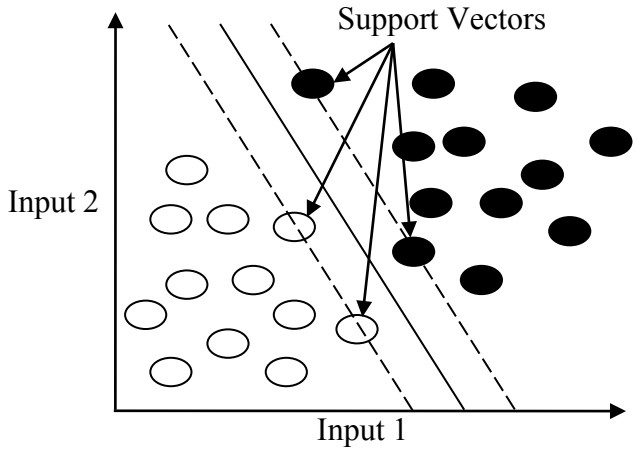


Figure A.3: Two class data (black and white ovals) distributed in a 2 dimensional vector space, showing the linear discriminant function (solid line), margin (space between the two dashed lines) and support vectors.

A discriminant function is also known as a kernel. The type of kernel depicted in Figure A.3 is a linear kernel. Other types if kernels include polynomial kernels and Gaussian kernels. Generally, for high dimensional vector spaces, linear kernels can achieve just as good accuracies as polynomial and Gaussian kernels.

If classes are not linearly separable, then the data is projected into a higher dimensional space where the classes can be separated using a linear hyperplane, as depicted in Figure A.4. Furthermore, SVMs can be designed to be fairly robust towards outliers by setting the trade-off and penalty parameters.

SVMs are also used for GEP classification since they are fairly insensitive to the Curse of Dimensionality [57]: GEPs can comprise hundreds, even thousands of expression values per sample. Often only a few samples are available for training, hence conventional feed forward Artificial Neural Networks would yield poor results, as shown by Statnikov et. al [57].

An extensive comparison between various types of classifiers is presented by Statnikov et. al.. He concludes that the multi-class SVM (MC-SVM) is the most accurate classifier for microarray data. There are a number of variations of MC-SVMs.

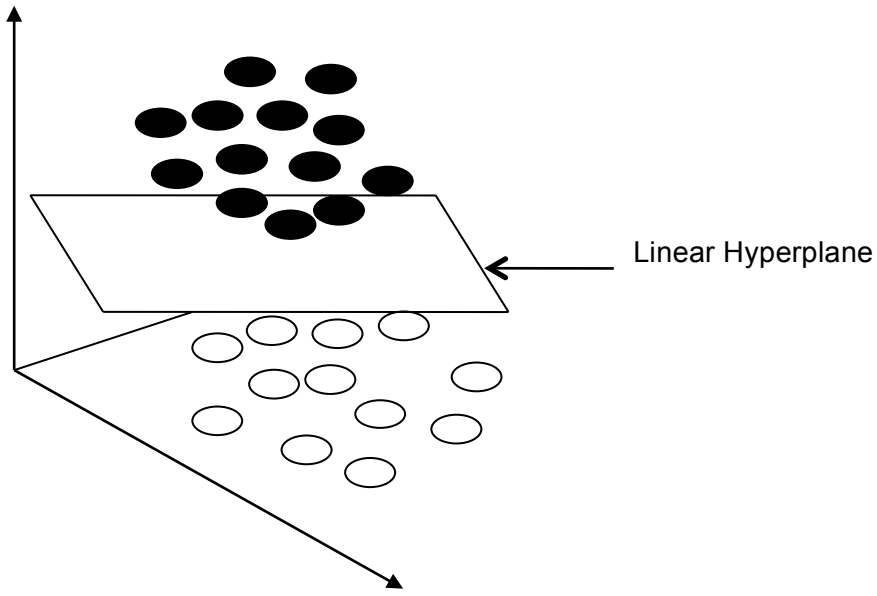


Figure A.4: The input space in Figure A.3 is projected into three dimensional space where the classes are linearly separable.

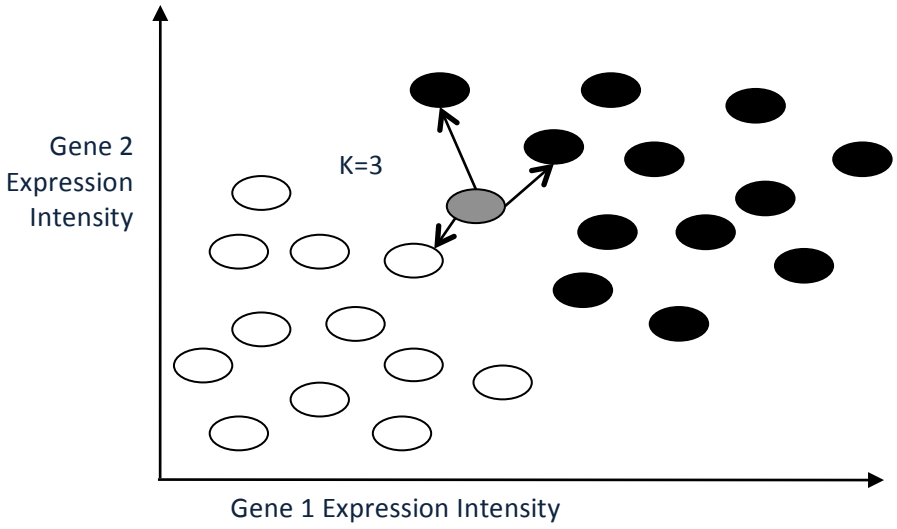


Figure A.5: K-nearest neighbour classifier. The unknown sample is assigned to the class to which the majority of its neighbours belong. In this instance, the unknown sample (grey) is assigned to the class with the majority of 3 (the neighbourhood radius is 3).

A.4 K-Nearest Neighbour

K-nearest neighbour (KNN) classification is a non-parametric classification technique first advanced by Cover et.al. [76] in the 1960's. KNN assigns an unknown sample to the class belonging to the majority of samples in it's neighbourhood.

The neighbourhood radius is specified by the number of nearest samples required to make a class assignment. The optimal neighbourhood radius is either pre-defined (for unsupervised learning) or learned during cross-validation [61]. KNN is the simplest of the algorithms described in this chapter (and hence the least computationally expensive) yet has performed well on microarray data [124].

A.6 Naive Bayesian Classifier

The Naïve Bayesian Classifier (NBC) is a probabilistic classifier based on Bayes theorem and assumes that each feature is class-independent of one another. In the context of expression profiling, NBC assumes that each gene independently contributes to the probability that a sample belonging to a particular class [56].

A NBC, like any supervised classifier, undergoes training in order to establish the optimal parameters of the probability distribution [56]. Typically, a Gaussian distribution is assumed for each feature for each class and the optimal mean and standard deviation are identified during training. When classifying an unknown sample, The NBC calculates the posterior probability of the sample belonging to each class, by comparing the distributions of the samples features to those identified during training [56]

A.5 Conclusion

The four most common algorithms for microarray data classification have been presented and discussed: MLP-ANN, RBF-ANN, SVM and KNN. Based on previous studies done, the most effective approaches to microarray data classification are SVM and KNN. Since KNN is the more computational inexpensive algorithm, it is used for most of the classifiers trained in this study.

APPENDIX B FUZZY INFERENCE

B.1 Introduction

Fuzzy logic is defined as a set of mathematical principles for knowledge representation based on degrees of membership [87]. As opposed to Boolean logic, Fuzzy logic is multi-valued. Where a Boolean number is either a one or a zero, Fuzzy logic takes on the entire spectrum from one to zero. An element can be partly true and partly false at the same time.

Fuzzy control is inspired by the human ability to make decisions based on imprecise information [94]. When deciding to cross the road, one does not need to know the roads precise width in meters in order to assess whether it is safe to cross or not. Fuzzy logic is an attempt to implement this imprecise human type of decision-making in machines, which normally deal with well defined discrete numeric values. The purpose of this appendix is to provide the reader with a background to Fuzzy modelling.

B.2 Fuzzy Set Theory

A Fuzzy set is a set which does not have clearly defined boundaries [87]. It is possible for an element to only belong partially to the set. In classic set theory, an element belongs to a set or doesn't belong to a set. It cannot belong to the set and not belong to the set simultaneously.

Fuzzy sets are particularly good at representing linguistic, subjective terms: a person can be defined as tall, average or short. Tall, average and short are Fuzzy sets since a particular height can be considered tall for one person, whereas another person can consider the same height to be average. Fuzzy set theory accounts for different degrees of tallness, as depicted in Figure B.1 [125]. This is achieved by means of membership functions. A Fuzzy membership function is a curve that defines how each input value is mapped to a degree of membership between 0 and 1.

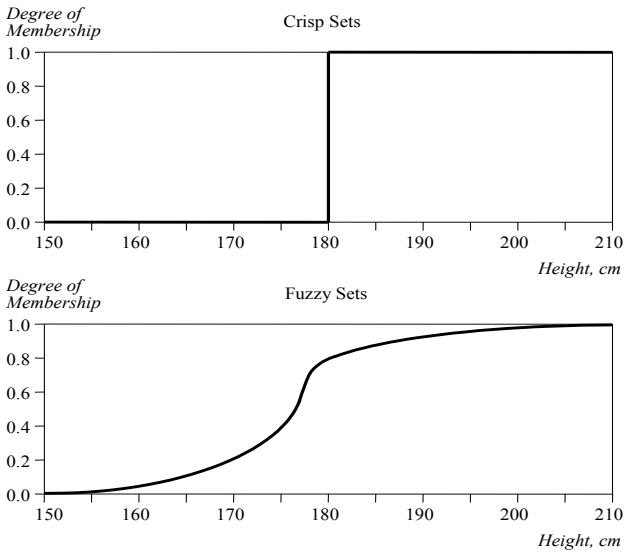


Figure B.1: A Crisp Set and a Fuzzy Set describing the height of a person [125].

B.3 Fuzzy Logic

Fuzzy logic implements imprecise human decision making via a set of Fuzzy rules [125]. Fuzzy rules implement a series of "IF…THEN" statements, which map imprecise conditions onto imprecise results. This is different to classical logic which maps precise conditions onto precise results.

- A Classic "IF…THEN" Statement: "If car-speed = 60 km/h Then Stopping Distance = 30 m".

- A Fuzzy "IF…THEN" Statement: "If car-speed is Slow Then Stopping Distance is Short".

Both Slow and Short are intangible and are both Fuzzy sets. OR and AND statements also have significance in Fuzzy logic. AND implies an intersection between two Fuzzy variables (selecting the minimum degree of membership) where OR implies the union between them (selecting the maximum degree of membership).

Fuzzy logic is used in rule evaluation by mapping input Fuzzified variables to specific output actions of a Fuzzy system.

B.4 Fuzzy Inference

Fuzzy inference is the process which maps input variables to output variables via a complex Fuzzy system [90], comprising Fuzzy membership functions and Fuzzy logic rules. This process involves four steps [125]:

1 Fuzzification of inputs.

2 Rule evaluation.

3 Aggregation of outputs.

4 De-fuzzification of outputs.

B.4.1 Fuzzification

Fuzzification involves determining the degree of membership of each crisp input value, to the various Fuzzy sets [125]. This is done by passing the input values into Fuzzy membership functions. Each crisp input could be mapped to more than one fuzzy membership curve, as illustrated in Figure B.2 [125].

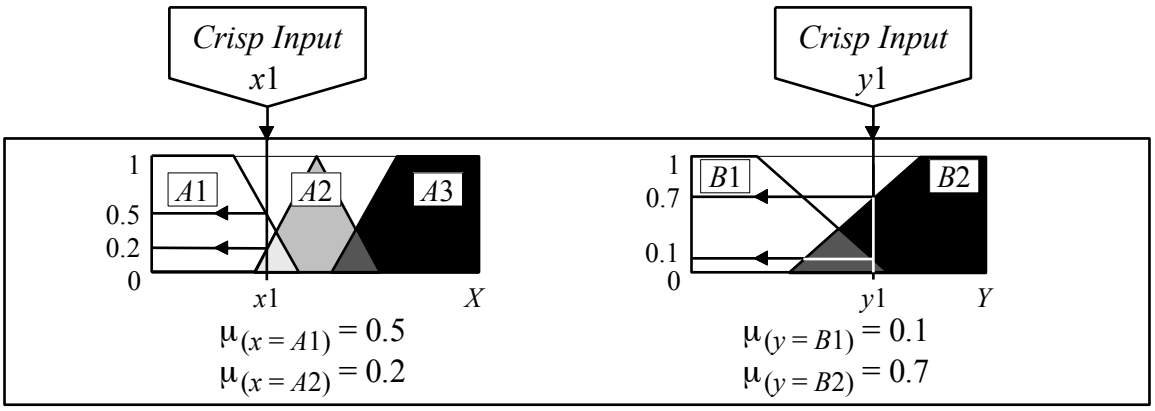


Figure B.2: Fuzzification of the Input Variables [125].

In Figure B.2, A1, A2, A3, B1, B2 are the various Fuzzy sets. μ (x) is the degree of membership of the input to a particular Fuzzy set.

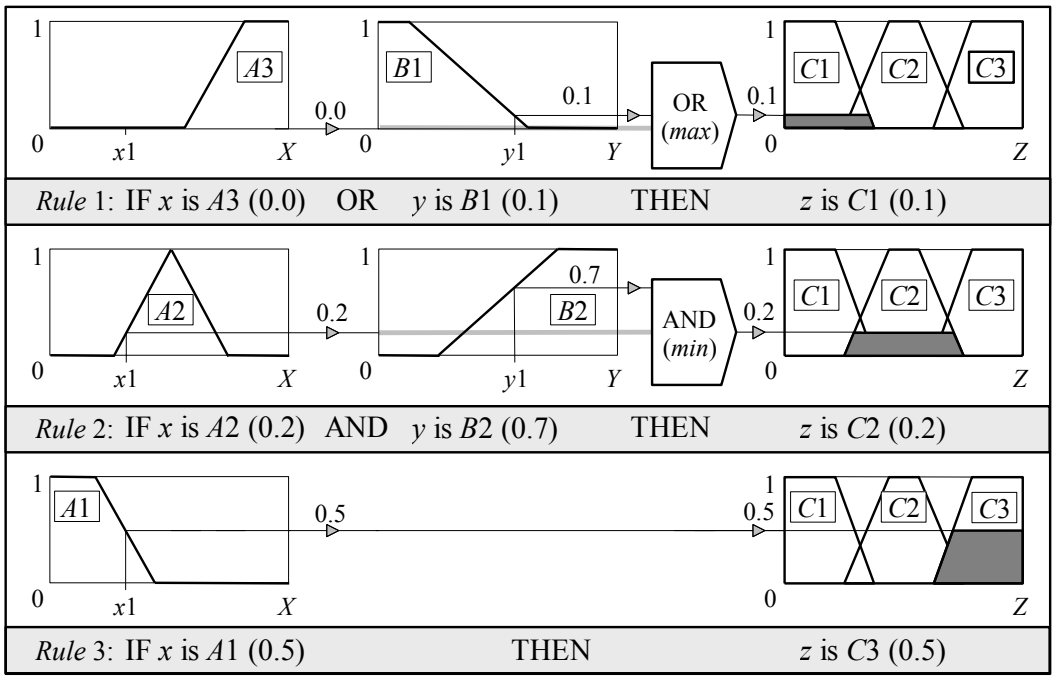


Figure B.3: The mapping of the Degree of Membership of the Antecedent to the Consequent [125].

B.4.2 Rule Evaluation

Once the degree of membership of each input variable, to each Fuzzy set, is determined it is necessary to map the input membership values to output membership values. This is done by implementing a set of Fuzzy "IF…THEN" statements, as illustrated by Figure B.3 [125]. The input degree of membership values are defined as the antecedent to the Fuzzy rules. The output degree of membership values are defined as the consequent. Once an output degree of membership value is obtained, the corresponding membership curve is clipped at that value.

Rule 2 states that IF the input variable x belongs to Fuzzy set A2 AND the input variable y belongs to the Fuzzy set B2, THEN the output variable z is mapped to the Fuzzy set C2. In this particular example, the input variable x has a membership of 0.2 to set A2 and y has a membership of 0.7 to set B2. Therefore, the output membership curve C2 is clipped at 0.2, since a Fuzzy AND is implemented, selecting the minimum degree of membership to be mapped to the output. A Fuzzy OR selects the maximum degree of membership, as illustrated in Rule 1.

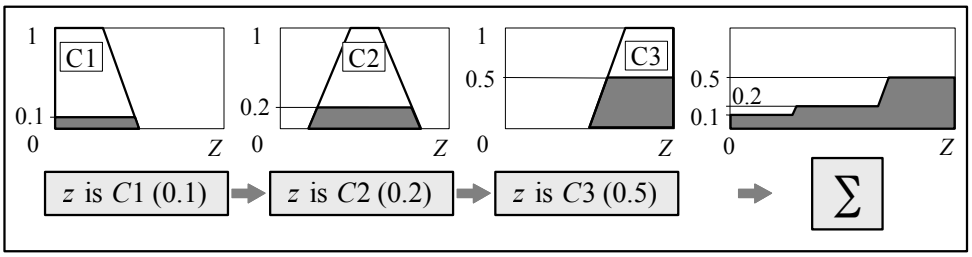


Figure B.4: Aggregation of each of the results from each of the rules to form a single Output Set [111].

B.4.3 Aggregation

Aggregation involves combining the results of the rules, producing a single output set. Figure B.4 [125] illustrates how the results of the rules, depicted in Figure B.3, are aggregated (using the fuzzy OR operation). The clipped fuzzy membership functions are combined together to form the aggregated fuzzy output.

B.4.4 De-fuzzification

A crisp input is derived from the aggregated Fuzzy set, by determining the centre of mass of the output Fuzzy set, using the following equation [99]:

$$x^* = \frac{\int \mu_i(x)\, x\, dx}{\int \mu_i\, dx} \tag{B.1}$$

Where x^* is the Defuzzified output, x is the Crisp Output and $\mu_i(x)$ is the degree of membership of a crisp output value to a particular set.

This is known as the Centroid de-fuzzification method. Another method is the Scaling de-fuzzification method, which scales the entire output set and the crisp value corresponding to the maximum degree of membership of the set is determined as the output value.

The Fuzzy inference technique described here is known as the Mamdani inference system [99]. The other commonly used technique is the Sugeno inference system, which is not discussed here. Mamdani inference is highly efficient in capturing knowledge but is computationally expensive. Sugeno inference is computationally effective and optimal for control applications [125].

B.6 Conclusion

An introduction to Fuzzy set theory, Fuzzy logic and Fuzzy inference is given. The major steps involved in Fuzzy inference are dealt with: fuzzification, rule evaluation, aggregation and de-fuzzification.

APPENDIX C GENETIC ALGORITHM

C.1 Introduction

GA is a population based optimization technique inspired by biological genetics and the Darwinian theory of evolution (survival of the fittest and natural selection) [101, 106]. GA performs a guided search through a population (set of numerical data) whereby individuals (potential solutions to a problem) undergo a 'natural selection' process in order to identify the best individual. The genetic algorithm described here is based on the one originally described by Holland et.al. [106].

An individual consists of a combination of genes, where the definition of a gene is application specific. For example, in feature selection, a gene is defined as a feature, where a gene can take on two possible states: 1 (the feature is selected) or 0 (the feature is rejected). This is also known as binary encoding (Figure C.1) since an individual is represented as a binary number. Other encoding approaches include float point encoding where an individual is represented as a decimal number. The optimal combination of genes could lie dormant amongst the population and could come from a combination of individuals. An individual with a genetic combination close to the optimal is described as being fit.

A new generation of individuals are spawned by mating two individuals from the current population. The fitness function is used to determine how close an individual is to the optimal solution. The selection function ensures that genetic information from the fittest individuals is passed down to the next generation, generating a fitter population. Eventually the population will converge on the optimal solution or get as close to it as possible. Implementation of a GA is carried out in four steps: Initialization, selection, reproduction and termination.

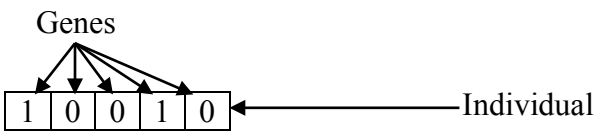


Figure C.1: Individual consisting of a number of genes using binary encoding.

C.1 Initialisation

Initialization entails encoding the chromosomes into a format suitable for natural selection. There are several types of encoding modalities, each with their advantages and disadvantages. Each individual of a population can be represented as a binary number. Since a binary number consists of ones and zeros (base 2), more digits are required to define an individual than if a decimal number was used (base 10). This lends itself to greater diversity in chromosome representation and hence greater variance in subsequence generations. The problem with binary encoding is that most populations are not naturally represented in binary form due to the length of binary numbers; they are computationally expensive.

Another form of encoding is floating point encoding. Each individual is represented as a floating point number or a combination of floating point numbers. Floating point encoding is far more efficient than binary encoding. Value encoding is similar but allows for characters and commands to represent an individual.

C.2 Selection

Selection of individuals for mating involves using a fitness function. A fitness function is used to determine how close an individual is to the optimal solution. The fitness function is the only part of the GA which has knowledge of the problem. The fitness function for the Sudoku problem is discussed in Section 3.

After defining the fitness of each individual, it is necessary to select individuals for mating. There are various methods used. Two methods are discussed here. The Roulette technique involves first summing the fitness's of all the individuals of a population and then selecting a random number between zero and the summed result. The fitness's are then summed again until the random number is reached or just exceeded. The last individual to be summed is selected.

Another selection technique is the tournament method. The tournament method involves selecting a random number of individuals from the population and the fittest individual is selected. The larger the number of individuals selected, the better the chance of selecting the fittest individual.

Selection ensures that the fittest individuals are more likely to be chosen for mating but also allows for less fit individuals to be chosen. A selection function which only mates the fittest individuals is termed elitist and may result in the algorithm converging to a local minimum.

C.3 Reproduction

Reproduction consists of two different genetic operations: crossover and mutation.

Crossover is the process by which two individuals share their genes, giving rise to a new individual. Crossover ensures that genes of fit individuals are mixed in an attempt to create a fitter new generation. There are various types of crossover depending on the encoding type, two of which are mentioned here: simple and arithmetic crossover.

Simple crossover is carried out on a binary encoded population. This involves choosing a particular point and all genes up until that point will come from the one parent while the rest comes from the other (Figure C.2). For example, one parent has the following binary configuration: 11010100. It is also possible to choose multiple points, which signify where crossover occurs.

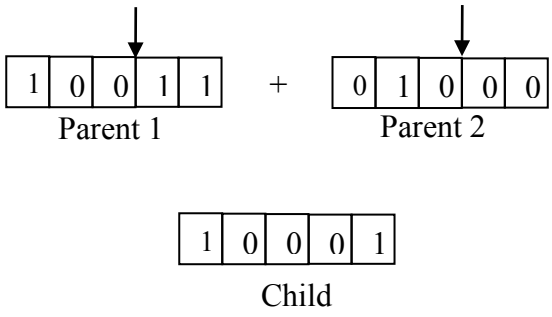


Figure C.2: Simple Crossover. Genes from the two parents are combined to form a new individual – the Child. All genes before the crossover point (indicated by the arrow) are derived from the Parent 1 while the rest of the genes are derived from Parent 2.

In arithmetic crossover a new generation is created from adding a percentage of one individual to another. For example an individual has the value 9.3 and another 10.7. If we select 30% from the one and 70% from the other the child will be 10.2.

Over the course of reproduction, a child's chromosome will go through mutation. Mutation is when the gene sequence of a chromosome is altered slightly, either by changing a gene or by changing the sequence. This is done to ensure that the population converge to a global minimum as opposed to a local minimum.

C.4 Termination

Termination: determines the criteria for the algorithm to stop. This can be once the optimal solution is reached but could be computationally expensive. Otherwise the GA can terminate once a certain number of generations has been reached, if the optimal solution has not been reached or once no better solution can be achieved.

C.5 Conclusion

Genetic algorithm consists of four operations which vary depending on the encoding scheme implemented. For binary encoding, point crossover and mutation are generally implemented while for float point encoding arithmetic crossover is generally used. For selection, a trade-off must be made between selecting fit individuals for mating and ensuring diversity in th epopulation to prevent convergence to local minima.