Cluster Analysis of Gene Expression Data on Cancerous Tissue Samples

Steven Conrad Dinger

A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in fulfillment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, 2011

DECLARATION

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

.....
(Signature of Candidate)

.....day of.....year.....

ABSTRACT

The cluster analysis of gene expression data is an important unsupervised learning method that is commonly used to discover the inherent structure in the large amounts of data generated by microarray measurements. The focus of this research is to develop a novel clustering algorithm that adheres to the definition of unsupervised learning whilst minimising any sources of bias. The developed diffractive clustering algorithm is based on the fundamental diffraction properties of light, which presents a novel view and framework for clustering data. The algorithm is tested on multiple cancerous tissue data sets that are well established in the literature. The overall result is a clustering algorithm that outperforms the conventional clustering algorithms, such as k-means and fuzzy c-means, by 10% in terms of accuracy and more than 30% in terms of cluster validity. The diffraction-based clustering algorithm is also independent of any parameters and is able to automatically determine the correct number of clusters in the data.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Michael Anton Van Wyk, for providing support, guidance and opportunities during my study. I am grateful for all the comments and help he has afforded throughout the research. I also appreciate the opportunity to have been a part of the control research group at the University of Witwatersrand.

I would also like to thank Prof. David Rubin and Dr. Sergio Carmona for their advice and exposure to the medical aspect of the project. I also thank Mr. Jonathan Featherstone for his help on the microarray terminology and equipment demonstrations at the molecular medicine and haematology department in the University of Witwatersrand.

Finally, I would like to thank my family for their continued support and patience throughout my studies.

Contents

DECLARATION 1

ABSTRACT 2

ACKNOWLEDGEMENTS 3

LIST OF FIGURES 8

LIST OF TABLES 9

1 INTRODUCTION 10

2 BACKGROUND 13

2.1 Description of Relevant Biology 13
2.2 The DNA Microarray 15
2.3 Statistical Analysis and Feature Selection 19
2.3.1 Statistical testing 19
2.3.2 Feature selection 25
2.4 Microarray Data Visualisation 26
2.5 Supervised Learning 30
2.5.1 Artificial neural networks 30
2.5.2 Support vector machines 31
2.5.3 Naïve Bayes networks 32
2.6 Unsupervised Learning 33
2.7 Distance Metrics 34
2.7.1 Euclidean distance 34
2.7.2 Manhattan distance 35
2.7.3 Maximum distance 35
2.7.4 Minkowski distance 35
2.7.5 Mahalanobis distance 35
2.8 Gene Expression Data Clustering 36
2.8.1 Gene-based clustering 37
2.8.2 Sample-based clustering 37
2.9 Clustering Algorithms 39
2.9.1 Hierarchical clustering 39
2.9.2 k-means 40
2.9.3 Self-organising map 41
2.9.4 Fuzzy c-means 44

2.10 Summary 45

3 DATA STANDARDISATION AND TRANSFORMATION 46

3.1 Data Standardisation 46
3.2 Data Transformation 48
3.2.1 Principal component analysis 48
3.2.2 Singular value decomposition 49
3.2.3 ISOMAP 50
3.3 Summary 54

4 CLUSTER VALIDATION 55

4.1 Internal Criteria 55
4.2 External Criteria 56
4.3 Relative Criteria 57
4.4 Cluster Validity Indices 58
4.4.1 Davies-Bouldin index 58
4.4.2 Dunn's index 59
4.4.3 Calinski Harabasz index 59
4.4.4 I index 60
4.5 Summary 60

5 DIFFRACTIVE CLUSTERING 62

5.1 Fundamental Theory of Diffraction 62
5.1.1 Fraunhofer diffraction 63
5.1.2 The Fourier transform and diffraction 64
5.2 Formalisation of Clustering Algorithm 66
5.2.1 Hierarchical diffractive clustering 68
5.2.2 Numerical clustering solution 69
5.2.3 Algorithm implementation 69
5.2.4 Cluster number selection 70
5.2.5 Classification 71
5.2.6 Algorithm illustration and properties 72
5.3 Summary 75

6 CLUSTERING RESULTS AND ANALYSIS 77

6.1 Golub Data Set 77
6.2 MILEs Data Set 82

6.3 Khan Data Set 85
6.4 Shipp Data Set 88
6.5 Pomeroy Data Set 91
6.6 Summary 94

7 CONCLUSION AND RECOMMENDATIONS 96
7.1 Discussion 96
7.2 Future Work 97

REFERENCES 102

LIST OF FIGURES

1.1 Various steps involved in analysing microarray data. 12
2.1 Relevant features pertaining to the eukaryotic cell. 14
2.2 Transcription and translation of mRNA into protein. 15
2.3 Illustration of the DNA microarray. 16
2.4 Oligonucleotide array with the steps involved in an expression study. . . 18
2.5 Dendrogram of the Golub data set using 51 ANOVA-selected feature genes. 28
2.6 A box plot for gene expressions under two conditions. 29
2.7 Scatter plot using the first two principal components of the Golub data set. 29
2.8 A generic layered ANN. 31
2.9 The two paradigms of hierarchical clustering. 39
2.10 Illustration of the iterative steps involved in the k-means algorithm. . . . 41
2.11 A two-dimensional self-organising map. 42
2.12 Rearrangement of the SOM units during trainging phase. 43
3.1 Logarithmic spiral illustrating properties of ISOMAP. 53
3.2 Residual variance of the ISOMAP algorithm as dimensionality increases. 54
5.1 Geometrical setup for the Fresnel-Kirchoff formula. 63
5.2 Physical setup for observing Fraunhofer diffraction. 64
5.3 Two parallel light rays originating from points O and Q in the xy plane. 65
5.4 Artificial data set used to illustrate the properties of diffractive clustering. 72
5.5 Logarithmic scale plot of the cluster number $\pi(\sigma)$. 73
5.6 Aperture function for the artificial data set for $\sigma = 8.9 \times 10^{-3}$. 74
5.7 Evolutionary tree diagram illustrating the convergence of the cluster centres. 74
5.8 Sigma evolution of the aperture function. 75
6.1 Residual variance of the ISOMAP algorithm for the Golub data set. . . . 78
6.2 Scatter plot of the *a priori* classification for the Golub data set. 79
6.3 Cluster lifetime plot for the two dimensional Golub data set. 80
6.4 Residual variance of the ISOMAP algorithm for the MILEs data set. . . 83
6.5 Scatter plot of the *a priori* classification for the MILEs data set. 84
6.6 Cluster lifetime plot for the three dimensional MILEs data set. 85
6.7 Residual variance of the ISOMAP algorithm for the Khan data set. . . . 86
6.8 Scatter plot of the *a priori* classification for the Khan data set. 87
6.9 Cluster lifetime plot for the three dimensional Khan data set. 88
6.10 Residual variance of the ISOMAP algorithm for the Shipp data set. . . . 89
6.11 Scatter plot of the *a priori* classification for the Shipp data set. 90
6.12 Cluster lifetime plot for the two dimensional Shipp data set. 91

6.13 Residual variance of the ISOMAP algorithm for the Pomeroy data set. . 92
6.14 Scatter plot of the *a priori* classification for the Pomeroy data set. 93
6.15 Cluster lifetime plot for the two dimensional Pomeroy data set. 94
6.16 Overall performance results of the clustering algorithms. 95

LIST OF TABLES

2.1 Post hoc tests used to determine which conditions are significant. 22
2.2 Visualisation techniques for microarray data. 27
3.1 Data normalisation methods. 47
4.1 Common indices that measure the similarity between partitions **C** and **P**. 57
6.1 Comparison of the clustering results for the Golub data set. 81
6.2 The six major subtypes of ALL. 82
6.3 Comparison of the clustering results for the MILEs data set. 84
6.4 Comparison of the clustering results for the Khan data set. 87
6.5 Comparison of the clustering results for the Shipp data set. 90
6.6 Comparison of the clustering results for the Pomeroy data set. 93

1 INTRODUCTION

The DNA microarray is a recently developed technology which enables biologists to measure gene expression profiles for thousands of genes simultaneously. The manufacturing of a DNA microarray commonly involves placing spots onto a glass film, where each spot represents a gene. A quantitative measurement is obtained from the number of complementary DNA (cDNA) hybridisations to the microarray. The number of hybridisations can be estimated using fluorescent markers attached to the cDNA samples together with laser imaging techniques [1].

The two main types of commercially available microarrays are cDNA chips and oligonucleotide chips, with the difference being in the manufacturing technique. The initial experiments performed using these chips have suggested that genes of similar function have similar expression profiles [2]. The data obtained from microarray experiments however is still accumulating and the biological importance being assessed.

Microarray measurements, which are currently being developed for cancer patients, allow for faster and more accurate diagnosis than previous clinical methods [3]. The genome however is large and as a result the amount of collected data is large. This, together with only a small number of samples presents a data analysis problem that is well suited for clustering [3–6]. The initial step therefore is to find patterns, assuming they exist in the genome, and then build classifiers from which more accurate and faster diagnoses can be achieved.

A problem with cluster analysis is determining the correct number of clusters in a high-dimensional data such as those obtained from microarrays. The purpose of this work is to develop a clustering algorithm that can automatically determine the correct number of clusters whilst successfully clustering the data. The developed unsupervised algorithm is also expected to cluster arbitrarily shaped clusters, which is commonly absent for most other clustering algorithms. The high-dimensional space also remains a challenge since the distance metrics, such as the Euclidean metric, are not as effective when the dimension increases [7].

A common solution to the dimensionality problem involves reducing the dimensions of the data prior to cluster analysis using an appropriate mapping technique. The most recognised and used technique is principal component analysis (PCA). It has however been shown that PCA performs inadequately for clustering gene expression data [8]. The idea of using non-linear reduction techniques on expression data, such as isometric mapping (ISOMAP), has been tested with surprising results that outperform linear techniques

like PCA [9].

The large amount of information embedded in the genome is hard to analyse statistically and is often compounded with noise from external factors like laboratory equipment. The supervised learning techniques, along with selecting feature genes to aid analysis, have been shown to perform exceedingly well in the literature [10–12]. The problem is that supervised learning techniques, like feature gene selection, make use of *a priori* information and in a sense are biased towards what information is used and how it is used. The benefit of unsupervised techniques is that the bias is minimised as much as possible, since the techniques do not require any predetermined information pertaining to the data.

The clustering analysis is focused on microarray data particularly those obtained from cancerous tissue samples. The data sets analysed are the Golub, Microarray Innovations in LEukaemia (MILE), Khan, Shipp and the Pomeroy data set [10,13–16]. The aforementioned data sets cover acute myeloid leukaemias, acute lymphoblastic leukaemias, small round blue-cell tumours, diffuse large B-cell lymphomas and central nervous system cancers respectively. The data sets therefore cover a number of cancers which allows for the robustness of the developed clustering algorithm to be examined.

A flow diagram, shown in figure 1.1, illustrates the various steps involved in analysing microarray data. The data representing the gene expression levels is often normalised to correct for background effects and random noise due to different microarray chips. After the data is normalised gene selection or clustering is performed to filter down and select the most important features of the data. The selected genes can then be used to train a classifier for making predictions and diagnosing a patient. The other possible route, which is used in this dissertation, is to automatically discover subtypes in the data.

It is also common to validate the functions of the selected genes and test their biological significance using an ontology. The gene ontology (GO) is a collaborative effort which has standarised the vocabularies describing gene products in terms of biological processes, cellular components and molecular functions [17]. The use of the GO for profiling groups of genes, and testing the statistical significance of the results produced, is currently an active area of research [18].

The research goal is to accurately predict subtypes in cancerous tissue data in an unbiased and unsupervised autonomous fashion. The various techniques that are capable of preforming in this manner are to be tested using well known data sets on cancerous tissue samples. The results obtained are also to be validated by using appropriate methods and comparison criteria.

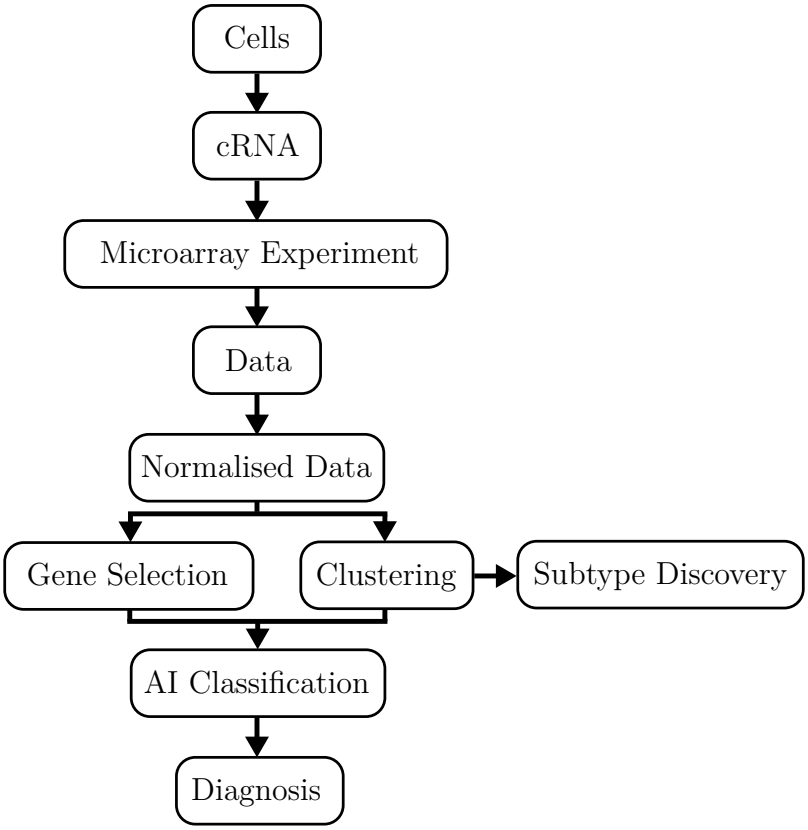


Figure 1.1: Various steps involved in analysing microarray data.

A novel technique was discovered which was derived using a physical and mathematical framework based on light diffraction. The author derived the technique independently of any sources and only later discovered its surprising similarity to scale-space clustering, which is derived using a probabilistic framework [19]. To the best knowledge of the author, this dissertation represents the first use of the scale-space algorithm (and obviously by virtue of its novelty, the diffraction clustering algorithm) on gene expression data.

The dissertation is organised in the following manner: Chapter 2 provides the necessary background on the biology and technology for measuring gene expression levels, an explanation on supervised learning classifiers with a description on unsupervised clustering algorithms that are used for comparison. Chapter 3 provides the tools used for standardising and transforming the data to a lower dimensional space. Chapter 4 outlines the various performance measurements and indices for validating the clustering results.

In Chapter 5 the derivation of the novel diffraction based clustering algorithm, which represents a contribution of this dissertation, is presented. Chapter 6 presents the results obtained from the various algorithms and data sets. Chapter 7 summarises the important findings and suggestions for future development of the algorithm and work.

2 BACKGROUND

In order to understand the problem of clustering gene expression data an overview of the relevant biology, employed technology and commonly used statistical methods are covered. The visualisation tools used on the results of microarray experiments are also presented in order to illustrate the various representations of the data. A brief description on some of the main supervised learning techniques and networks is also presented. The focus of the research however is cluster analysis and as such a comprehensive description explaining the terminology, distance metrics and different clustering algorithms is provided.

2.1 Description of Relevant Biology

In order to understand the volume and nature of the data pertaining to gene expressions, the biology of cells and their mechanism to replicate and code information should be understood. A cell can be divided into two classes prokaryotic and eukaryotic, with the latter containing a "true" nucleus i.e. has a nuclear membrane. A simplified diagram of a eukaryotic cell is shown in figure 2.1.

The cell is enclosed and protected by a phospholipid bilayer with the nucleus embedded in the cell's cytoplasm. The nucleus has its own nuclear envelope with nuclear pores sited around it to allow for the DNA (deoxyribonucleic acid) to interact with the rest of the machinery in the cytoplasm. The DNA molecule is a double helix of complementary strands with nitrogenous bases: adenine, cytosine, guanine and thymine. The complementary bases adenine and thymine are joined by hydrogen bonds, similarly for the complementary bases cytosine and guanine.

The cell uses DNA to transmit its hereditary information to the next generation via segments of the DNA called a gene [20]. The information transmitted by the DNA pertains to the construction of proteins, which are the functional units of life. All proteins consist of about 20 different amino acids that are covalently bonded in a sequence. The DNA molecule therefore acts as a linear translation platform in which triplet sequences of nucleotides code for every amino acid.

The *genome* is all the genetic material and collection of genes that is required by an organism to produce its proteins [20]. The human genome has about 30 000 to 40 000 genes whereas a simple yeast cell has about 6 000 genes [20]. The remarkable fact of life is that every multicellular organism has its entire genome contained in every cell [3]. The

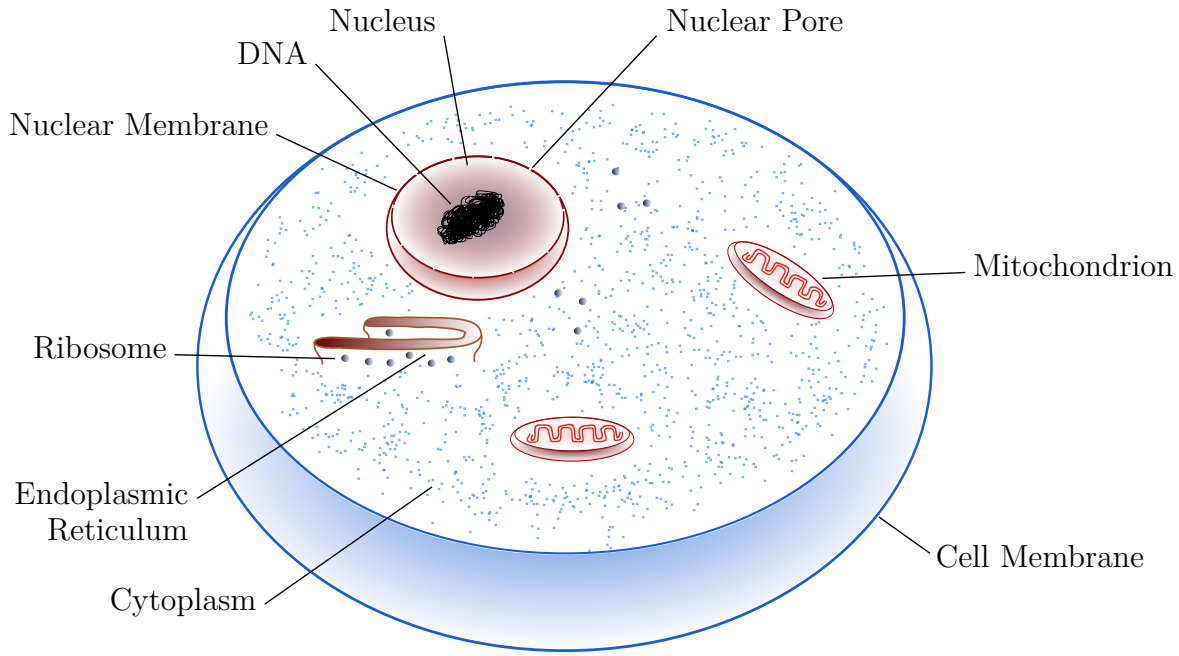


Figure 2.1: Relevant features pertaining to the eukaryotic cell.

cells of different tissues however can differ in terms of the amount and type of proteins produced in the cells.

A gene is said to be *expressed* if the protein which it codes for is produced or synthesized [3]. In an average human there are expression levels for about 10 000 different genes, which is collectively referred to as the *expression profile* of the cell [3]. A large number of genes located in all the cells of an organism share common functions, metabolism being such an example. The various internal and external factors however can adjust the amount of some gene expressions in different cells and even in the same cell.

The ribosomes are the protein synthesizing factories for the cell. As illustrated in figure 2.1 the ribosomes are situated outside the nucleus in the cytoplasm whereas the DNA is protected inside the nuclear envelope. The direct interaction is therefore broken between the ribosomes and genes. The communication occurs via a linear molecule called messenger ribonucleic acid (mRNA), which is an exact copy of the gene that is being expressed.

The mRNA is *transcribed* inside the nucleus and transported out to the ribosomes where it is *translated* into amino acids and subsequently into protein. A single gene is able to produce numerous identical protein molecules by manufacturing multiple copies of the corresponding mRNA molecule, as illustrated in figure 2.2.

The amount of protein produced can differ for a cell in which case the expression level for

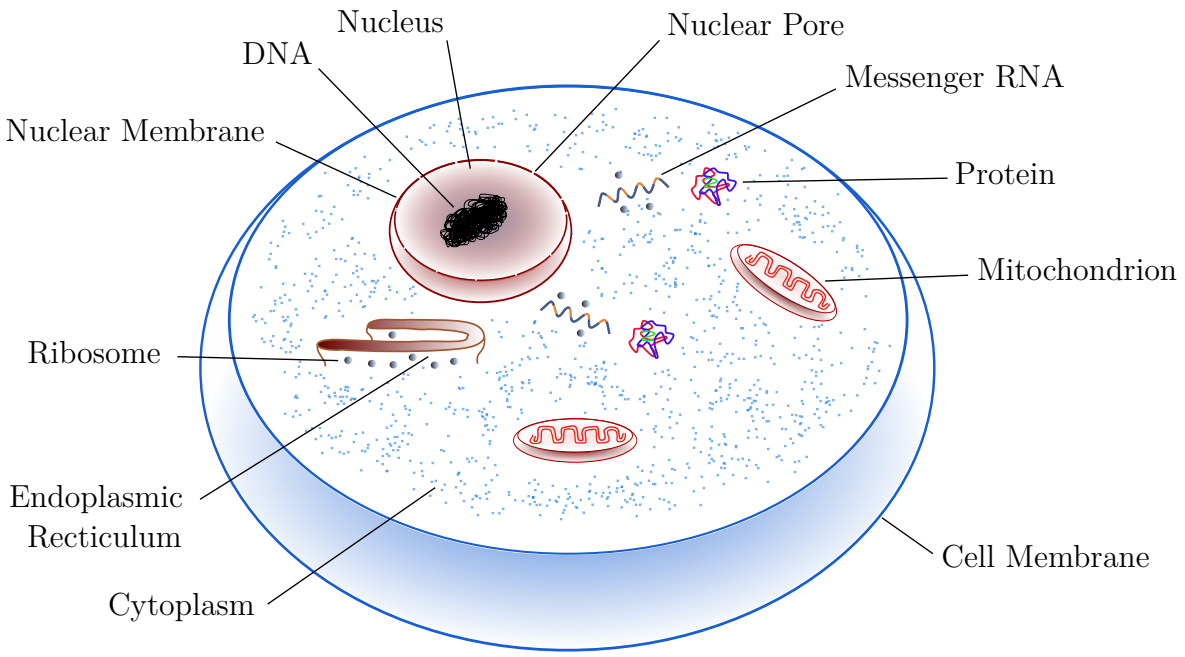


Figure 2.2: Transcription and translation of mRNA into protein.

each gene is regulated by what is known in current research as *regulatory networks*. The transcription process of the gene into mRNA is regulated by factors known as transcription factors [3]. The transcription factors bind to upstream promoter elements (UPEs) or an enhancer which increases the accuracy and rate of mRNA synthesis respectively [20]. The transcription factors can also be used to repress the expression of a certain gene. The gene expression profile therefore provides information about the *biological state* of the cell, and is measurable through the concentration of the respective mRNA molecules produced by a cell.

The direct proportionality between mRNA molecules and the amount of proteins expressed is an assumption. The relationship is in fact more complicated as there are several post-translational steps involved which affect the ratio of mRNA to protein concentration. The assumption however is used and validated by the numerous measurements performed thus far on the human genome. The following section deals with the technology and devices that are used to capture and measure the gene expression profile for a cell.

2.2 The DNA Microarray

The essentials of the microarray device, and the respective microarray experiments, are discussed in order to understand the capabilities of this technology. A brief discussion

on the fabrication process is also provided and includes the two most common types of microarrays.

The importance of this technology lies in the fact that microarrays can measure the expression levels for thousands of genes simultaneously during essential biological processes across collections of related samples [1]. Specifically the microarray measures the amount of mRNA in a cell, which is quantitatively related to the amount of protein synthesised [1]. The amount of mRNA for various genes is assumed to be directly proportional to the gene expression levels [4].

The various applications for microarray include: the comparison of different tissues, effects of drugs on a cell and understanding aging or fetal growth development [4]. A list of some of the fields that also benefit from this technology include: drug development, comparative genomics, diagnosis and functional genomics [4].

The basic structure of the DNA microarray, shown in figure 2.3, consists of a substrate (silicon, glass or plastic) onto which single stranded DNA molecules, each with different sequences, are deposited [4]. The single stranded DNA molecules are referred to as *probes*, and are arranged in a regular grid-like pattern on the substrate [21]. The types of probes deposited on the substrate depends on the purpose of the array. An array for example can be deposited with an arbitrary set of probes to uncover a general set of queries [4].

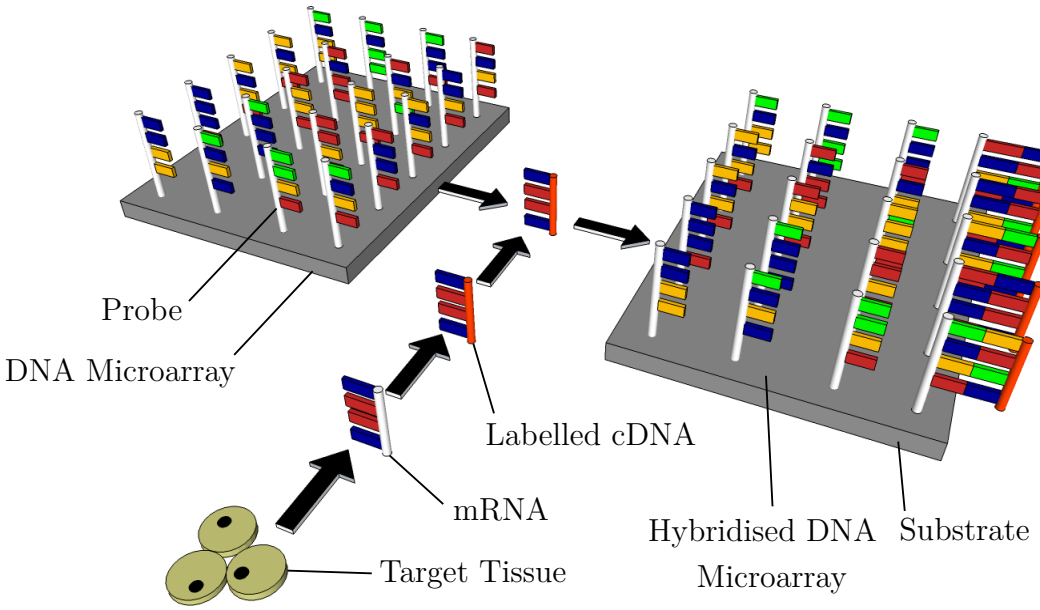


Figure 2.3: Illustration of the DNA microarray.

The two main types of microarray fabrication include *in situ* synthesis and the deposition of pre-synthesized single-sided DNA fragments [22]. The latter fabrication process involves either the deposition of polymerase chain reaction amplified cDNA probes or the printing of pre-synthesized oligonucleotides [22]. The former technique employed for *in situ* synthesis of probes is usually a photochemical process known as photolithography [1].

The advantage of *in situ* synthesis is that the probe sequences are known exactly in contrast to the pre-synthesized technique. The disadvantage however of *in situ* synthesis is that the size of the probes are usually limited, which results in a target gene having to be represented by about 20 probes [4].

The Affymetrix technology, which can measure up to 20 000 genes, follows the *in situ* type fabrication process, together with a match/mismatch probe strategy [4]. The millions of deposited perfect match probes (25 nucleotides in length) are made identical to the target sequence or gene, whereas the mismatch probes also 25 nucleotides in length, have a single nucleotide changed at the centre position.

The mismatch probes are used to estimate the lack of hybridization or background intensity of the captured image. The average difference is commonly used by the Affymetrix software, and is defined for N probes per gene as

$$\text{Gene Expression Value} = \frac{\sum_{i=1}^{N} (P_i - M_i)}{N}, \tag{1}$$

where P_i and M_i denote the i^{th} perfect match and mismatch probe respectively. The average difference in equation 1 provides a quantitative measure for the gene expressions across the 20 probe pairs for each gene [4].

The procedure for measuring gene expressions firstly involves extracting the mRNA molecules of a biological sample and then reverse transcribing them into complementary DNA (cDNA) sequences. The sample containing these cDNA molecules is often referred to as the *target* [4]. The target sample is then transcribed back to cRNA that is labelled with biotin. The solution is then placed onto the array where it diffuses and hybridises to the corresponding probes. The mixture is then washed, stained and finally exposed to an appropriate light source with the correct wavelength for excitation of the dye. The image captured contains multiple features, or hybridised spots, with the intensity of each feature related to the amount of mRNA [1]. The various steps of a microarray expression study are shown in figure 2.4.

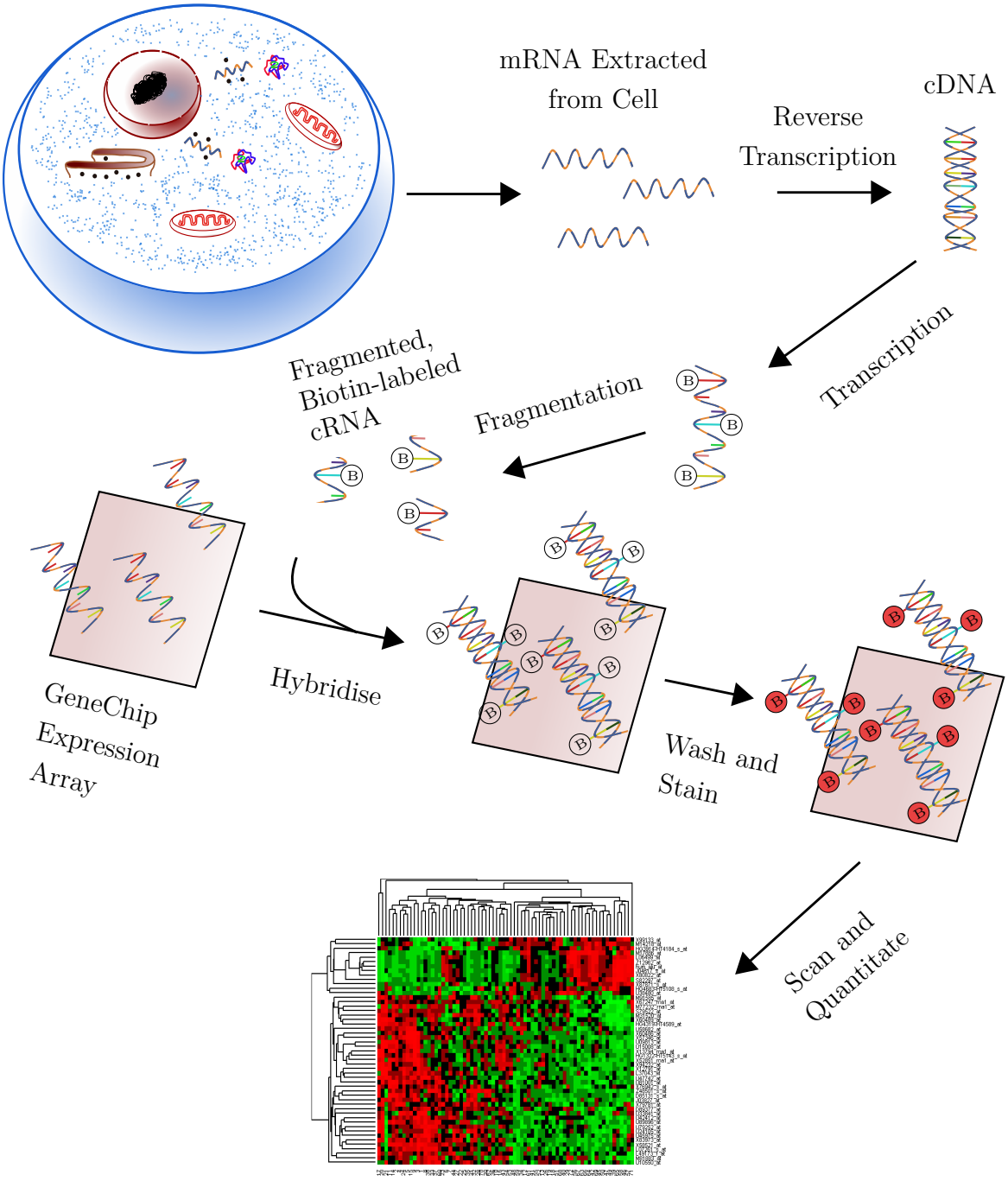


Figure 2.4: Oligonucleotide array with the steps involved in an expression study.

The two different types of microarray expression studies include single-channel and multi-channel experiments. In a single-channel type experiment only one sample target is labeled and measured. In a two-channel experiment, for instance, there are two target samples each with a unique colour dye for labelling. The advantage of a two-channel experiment is that it only requires a single microarray for the comparison of two different

samples and is therefore more cost effective [4].

The format of the data collected from an experiment is usually expressed as a matrix $N_g \times N_s$, where N represents the number of genes or samples [3]. The entries of the matrix, E_{gs}, represent the expression level for gene g and sample s. The size of an expression matrix is considerably large especially when one considers that a typical experiment involves 7 000 genes with a 100 samples.

The DNA microarray is therefore a useful and capable tool for measuring the large amounts of data embedded in the human genome. The most important aims when analysing a gene expression experiment, as mentioned by Domany [3], can be summarised as follows:

1. Identifying the genes, using their expression profiles, that are associated with cancers and other important processes.
2. Partition tumors into classes based on their expression profiles and in familiar clinical classification. Expression profile classification can be used as a diagnostic or therapeutic tool.
3. Use the data analysis to obtain information relating to the unknown functions of certain genes.

2.3 Statistical Analysis and Feature Selection

The DNA microarray, unfortunately, is not immune from the variety of noise sources that compound to the variability of the gene expression measurements. The variation caused by the laboratory procedure and protocols for example makes it difficult to distinguish the inherent variation due to differentially expressed genes. The sources of noise due to the laboratory procedure include: mRNA preparation, transcription, labelling, hybridization parameters (temperature, time etc.) and contaminants that effect the image analysis.

2.3.1 Statistical testing

The classical approach to analysing microarrays involves defining the type of problem based on two different criteria [4]. The criteria are given as:

1. Number of samples
2. *A priori* knowledge of the distribution

The number of samples can range from one to many samples. The second criterion includes parametric testing and non-parametric testing. A test is *parametric* if the data are assumed to have a specific distribution in contrast to a *non-parametric* test where there is no *a priori* knowledge of the distribution [4].

The statistics commonly used on the microarray data include the F-statistic, the chi-square statistic and the Student's t-statistic [4]. The F-statistic can be used to determine if the variance of the control subjects is different from the variance of the patients [4]. The chi-square is also another variance testing statistic that can be used to evaluate the performance of new microarray technology [4].

t-Statistic

The t-statistic is used on two different samples in order to evaluate whether the samples have a distribution with the same mean, and is defined as

$$t = \frac{\left(\bar{X}_1 - \bar{X}_2\right) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, \tag{2}$$

where, for the i^{th} sample of size n_i, the population mean is given as μ_i, the sample mean as $\bar{X}_i$ and s_i as the sample standard deviation.

The t-statistic in the context of microarray analysis is often used to determine whether a gene expression is stochastic or regulated between cancer and healthy patients for example. The F-distribution is also used prior to calculating the t-statistic in order to determine if the samples have the same variance [4].

The t-statistic, however, is limited when there are multiple parallel comparisons to be made, which is a typical procedure performed in a microarray experiment. The problem occurs because of the chosen significance level and applying the t-test multiple times to each gene separately. A microarray, for example, can have up to 10 000 genes, which implies that for a significance level of 5% that 500 genes will appear to be regulated, when in fact they could have changed randomly [4].

ANOVA

The testing paradigm called analysis of variance (ANOVA), another classical statistical tool, is able to avoid the increase in error introduced from multiple group comparisons [4]. The basic operations and steps involved in ANOVA are described as follows:

1. ANOVA calculates the mean for each group (e.g. cancer type).
2. The overall mean is calculated across all groups.
3. The within-group variation is calculated, which is the difference between each point and the group mean.
4. The between-group variation is calculated, which is the deviation between the group mean and the overall mean.
5. The final calculation is the ratio of the between-group variation to the within-group variation, which is known as the F-statistic.

The ANOVA test has two types of models with each model either being a one-way or two-way model. The type I model specifically compares each condition for a difference in expression, whereas a type II model treats the conditions as random [4]. The one-way model only considers a single factor to contribute to the variability of the results, namely the genes [4].

In a two-way ANOVA model more than one factor is considered to affect the variability of the data, the genes and the microarray platforms [4]. The difference between model types I and II for the ANOVA tests is illustrated by the partitioned sum-of-squares

$$S_t = S_c + S_e, \tag{3}$$
$$S_t = S_c + S_a + S_e, \tag{4}$$

where S_t is the total sum, S_c is the condition sum, S_e is the error sum and S_a the array sum [4]. The aforementioned sum-of-square terms can be defined, for sample X_{ij} with k number of genes and n number of conditions, as follows

$$S_t = \sum_{i=1}^{k}\sum_{j=1}^{n}\left(X_{ij} - X_{..}\right)^2, \tag{5}$$
$$S_c = \sum_{i=1}^{k}\sum_{j=1}^{n}\left(\bar{X}_{i.} - \bar{X}_{..}\right)^2, \tag{6}$$
$$S_e = \sum_{i=1}^{k}\sum_{j=1}^{n}\left(\bar{X}_{ij} - \bar{X}_{i.}\right)^2, \tag{7}$$
$$S_a = \sum_{i=1}^{k}\sum_{j=1}^{n}\left(\bar{X}_{.j} - \bar{X}_{..}\right)^2, \tag{8}$$
$$X_{..} = \sum_{i=1}^{k}\sum_{j=1}^{n} X_{ij}. \tag{9}$$

The ANOVA procedure determines whether a gene is statistically significant and therefore differentially expressed in any of the multiple conditions tested, however it does not indicate which specific condition(s). The post hoc tests, as described in table 2.1, can be used with ANOVA to determine which conditions are statistically significantly different from one another.

Table 2.1: Post hoc tests used to determine which conditions are significant.

Test	**Description**
Tukey	Means for each condition are ranked in order of magnitude, lowest mean gets a value of 1. The pairwise differences between means are tabulated between each group pair and divided by the standard error. The resulting q value is compared to a studentised range critical value, and if the q value is larger then it is deemed statistically significant.
Student-Newman-Keuls	Similar to the Tukey test with exception of the critical value. All the q values are compared to a different critical value which makes the test less conservative compared to the Tukey test.

Multiple testing corrections

Multiple testing correction procedures are able to mitigate the problem encountered by the statistical tests, which is the increase in false positives as the number of comparisons increases. The main problem in statistical tests is selecting a suitable confidence level known as the p-value. A typical p-value is 5% which is the probability of a false positive occuring (1 in 20).

The choice of the p-value however ultimately depends on the stringency of the experiment and expected quality for the results. The following information on the methods for multiple testing correction is summarised from the manual describing Agilent technologies for gene expression analysis [23].

The four main types of multiple testing correction techniques are listed as follows:

1. Bonferroni
2. Bonferroni-Holm step-down
3. Westfall and Young Permutation
4. Benjamini and Hochberg false-discovery rate

The list is ordered in terms of the technique's stringency level; for example the Bonferroni method is the most stringent compared to the Benjamini and Hochberg correction method. The stringency of the method determines the false-discovery rate and therefore how many false positives are encountered. The obvious trade-off is that as the stringency increases, a lower selected p-value, the false negative rate increases i.e. more significant genes fail hypothesis test.

Bonferroni The Bonferroni method multiplies the p-value for each gene by the total number of genes N_g. The resulting number is compared to the error rate (assumed to be 0.05), as shown in the relation

$$\text{Corrected } p\text{-value} = p\text{-value} \times N_g < 0.05. \tag{10}$$

If the result is smaller than 0.05 the gene is selected as it is deemed to be significant. An example is that if there are 10 000 genes then the corrected p-value will have to be below 5×10^{-6} for it to be considered significant.

Bonferroni-Holm step-down The Bonferroni-Holm step-down approach is similar to the Bonferroni method, however the stringency is reduced by altering the correction factor as listed in the following steps:

1. Genes are ordered from smallest to largest according to their p-values.
2. The first gene is tested using equation 10.
3. The second gene is tested using the relation

$$\text{Corrected } p\text{-value} = p\text{-value} \times (N_g - 1) < 0.05. \tag{11}$$

4. The rest of the genes are tested using similar formula with the N_g term altered by the gene's position in the ranking.

The method continues along the list of genes until there are no more genes found significant or until a certain user-selected number of genes is reached.

Westfall and Young permutation The Bonferroni and Holm methods are called single-step procedures as each p-value is corrected independently. The Westfall and Young method uses the fact of dependence between genes and permutes all the genes at once. The Westfall and Young procedure is similar to that of the Holm step-down method

except that it also permutes the genes to find the distribution, as listed in the following steps:

1. The p-values of the original data set are found and ranked according to their size (base test).
2. An artificial data set is created by permuting the genes into either a control or treatment set.
3. The p-values are recalculated for the pseudo-data set and ranked.
4. The minimal p-values are retained and compared to the original p-values.
5. The process is performed multiple times with the final adjusted p-values consisting of those which were lower than the base test by some proportion.

The Westfall and Young method however is the slowest technique especially for large permutations of the data set.

Benjamini and Hochberg false discovery rate The Benjamini and Hochberg false discovery rate is the least stringent method, and as a result tolerates more false positives. The method however still minimises the number of false negative genes to a respectable amount. The procedure is outlined as follows:

1. The p-values are ordered from smallest to largest.
2. The largest p-value remains the same.
3. The second largest p-vale is multiplied by N_g, divided by its rank and then compared to the threshold using the relation

$$\text{Corrected } p\text{-value} = p\text{-value} \times \frac{N_g}{N_g - 1} < 0.05. \tag{12}$$

4. The remaining genes are tested using similar formula with the correction factor altered by the gene's position in the ranking.

The Benjamini and Hochberg method is a good alternative to the other family-wise error rate techniques. The difference being that the false discovery rate method determines the percentage of genes selected as significant that occurred by chance i.e. false positives. The Benjamini and Hochberg method is the recommended technique as it is the least stringent and offers a good balance between statistically significant genes and false positives.

2.3.2 Feature selection

A significant and crucial problem to solve in microarray data analysis is selecting genes that are highly correlated with a specific phenotype or class. The data obtained from microarrays also contains a large number of irrelevant and redundant genes that can be filtered out in order to reduce the dimensionality of the gene feature space. The filtering of the genes should also be performed before feature selection, as it reduces the number of false positives.

To clarify the previous statement an analogy of say splitting 50 people into two random groups is often used [24]. The characteristics of the people (height, weight, age etc) are endlessly measured such that many differences arise between the two groups. The differences may reflect real biological changes, in terms of the microarray experiment, but many will be a result of chance. The requirement therefore is to filter out genes before analysis is carried out in order to reduce the number of false positives [24].

It is shown in literature that not more than 10% of the genes commnoly used in acute lymphoblastic leukemia studies are actually related to cancer classification [10]. It is therefore necessary to search for those genes which distinguish the biological process under investigation. The main task of feature selection is therefore to reduce the dimenensionality of the data as much as possible whilst retaining the most relevant genes of the different conditions.

The reason for reducing the dimensionality using feature selection is that it can improve classification predictors and the performance of clustering algorithms. The other benefits of a low-dimensional space include improvement in visualisation of the data and significant improvement of the signal-to-noise ratio.

The two common procedures for achieving dimension reduction are called *feature selection* and *feature extraction* [11]. In feature selection a test is performed whereby the features (genes) that contribute the most to the class separability are chosen. The test can either be a univariate approach whereby the features are ranked, or multivariate where a criterion function is optimised.

The other approach, feature extraction, deals with the linear or non-linear mapping of the data set from the original high-dimensional feature space to a lower-dimensional feature space. The transformation operation can also be supervised or unsupervised, where in the former case some criterion of separability or predictability is maximised, as shown by Bair et al [25].

The feature selection problem is more formally defined as the subset of features p that,

given m set of features and n labeled samples, maximises and contributes the most to class discrimination. The number of possible subsets representing the feature space is the following choose function $\binom{m}{p}$. The total number of subsets is therefore large even for reasonable values of m and p.

To overcome a large feature space, heuristic techniques such as sequential forward or backward selection have been developed [11]. The techniques use an objective function which is maximised each time a feature is either added or removed respectively. The feature selection techniques can be categorised into three main groups depending on how they are integrated with the classifier: *filter methods*, *wrapper methods* and *embedded methods* [11].

The filter method utilises a statistic or an informative criterion, such as the Pearson correlation coefficient defined as

$$R_{XY} = \frac{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{(n-1)\sigma_X \sigma_Y}, \tag{13}$$

from which a rank for the features or genes can be determined.

The top p ranking genes are selected and used to train or test the classifier. In wrapper methods a search strategy is implemented, such as sequential forward or backward selection, for the best subset of features. The performance of a classifier is used to evaluate the feature selection and as such is wrapped around the algorithm of the classifier [11]. Embedded methods however are those that are built into the classifier architecture, with the features selected as part of the particular classifier [11].

2.4 Microarray Data Visualisation

The visualisation of microarray data can be accomplished using a variety of techniques. The form of the microarray data however is important and determines which technique is used for visualisation. The three major forms of gene expression data include: time series, identical parameters and non-identical parameters [26]. The time series data are a collection of gene expression profiles that are taken over specific time intervals. The identical parameter data are obtained when the samples from different patients are arranged in a two-dimensional grid and compared. The non-identical parameter data are obtained when there are parameters for many different sets of observations or conditions [26].

The various visualisation tools and techniques are summarised, as shown in table 2.2, from the findings of Prasad and Ahson [26]. The abbreviated applications in table 2.2

stand for the following: Self-Organising Map (SOM), Support Vector Machine (SVM), Principal Component Analysis (PCA) and Hierarchical Clustering (HC).

Table 2.2: Visualisation techniques for microarray data.

No.	**Visualisation Tool**	**Description**	**Application**
1	Clustering view (temporal)	Waves illustrate the genes across the time sequence.	SOM, SVM, k-means, HC and PCA
2	Heat map	Colour display for variation in expression intensity	SOM, k-means, SVM and HC
3	Dendrogram	Gene tree, array tree and colour coded bands of gene expression.	HC
4	Scatter plot	Plots genes after PCA. Data points are plotted on axes.	PCA
5	Box plot	Box representing interquartile range and median	Raw and preprocessed data

The clustergram or dendrogram is often used to visualise the hierarchical clustering results of gene expression data. The Golub data set was filtered using a one-way ANOVA test with a p-value cutoff of 3%, the results of which are shown in figure 2.5. It is clear from the vertical colour separation in figure 2.5 that there are two distinct classes of cancer, which for this data set are the acute myelod leukaemia (AML) and acute lymphoblastic leukaemia (ALL) classes [10].

The red coloured blocks in the dendrogram illustrate genes that are expressed above the mean, which is coloured black, and green for genes that are below the mean for the data set. The tree diagram or hierarchical structure of the genes and samples are shown on the vertical and horizontal axis respectively. The genes can also be annotated as is illustrated by the gene accession numbers on the vertical axis in figure 2.5. The genes and samples are arranged such that similar expression values are situated next to each another, which allows for a clear representation of the distinct classes.

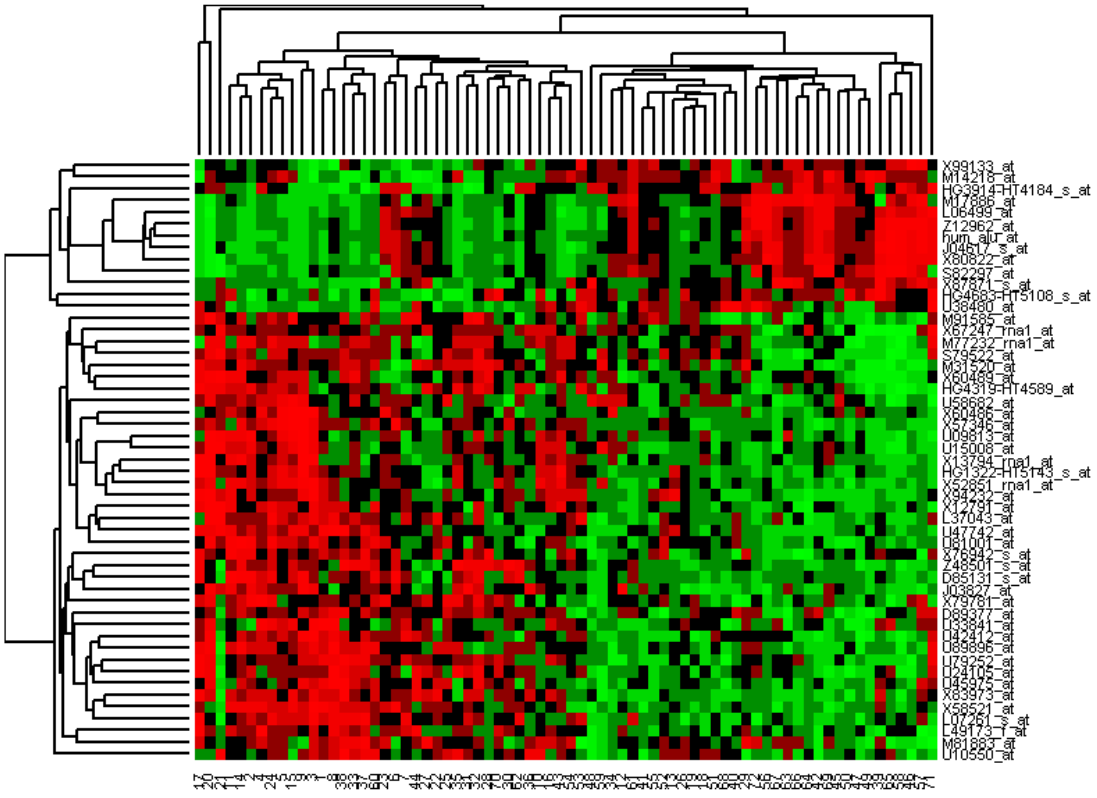


Figure 2.5: Dendrogram of the Golub data set using 51 ANOVA-selected feature genes.

The most common representation of raw and preprocessed expression data is a diagram called a *box plot*. The box plot for a generic example is shown in figure 2.6, where the line in the middle of the box represents the median. The superior edge of the box is the upper quartile (75%) and the inferior edge is the lower quartile (25%). The box has two tails defined as one and a half times the interquartile range, with the data points outside the tails referred to as outliers [4].

The measured data can also be visualized using a scatter plot, whereby the gene expressions for the first experiment or condition are represented by the horizontal axis and the expressions for the second experiment or condition are represented by the vertical axis [4]. The scatter plot is also useful for displaying the principal components of the data set. An example of a scatter plot for the first two principal components of the Golub data set is shown in figure 2.7. The principal component analysis is unable to recover the distinct classes in two dimensions, as shown in figure 2.7. An enhanced method such as isometric mapping (ISOMAP) is better suited for recovering embedded structure in a high-dimensional space, as discussed in section 3.2.3.

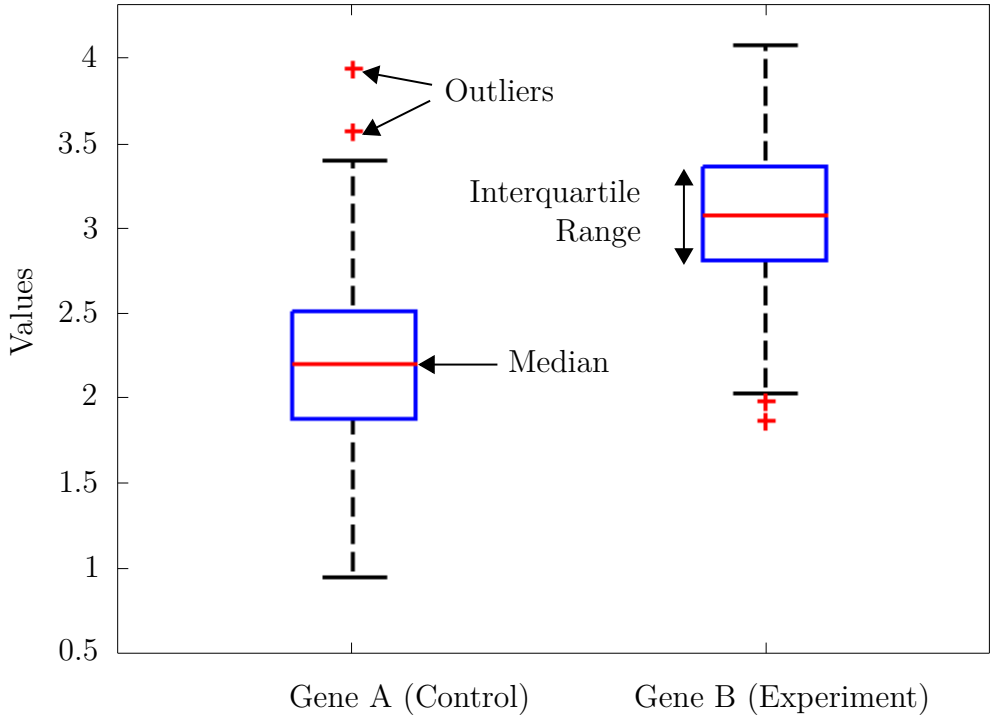


Figure 2.6: A box plot for gene expressions under two conditions.

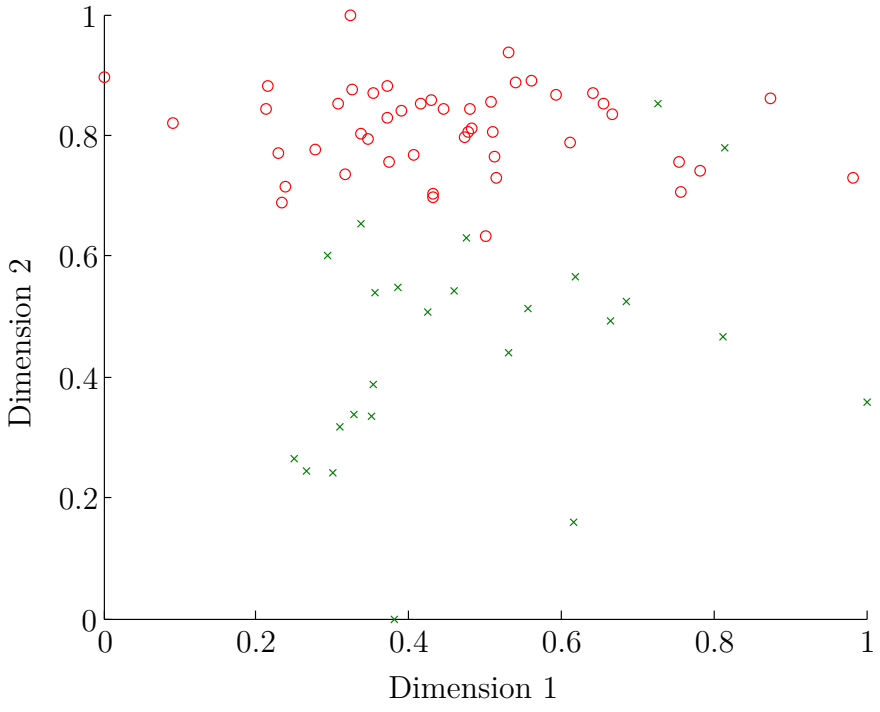


Figure 2.7: Scatter plot using the first two principal components of the Golub data set.

2.5 Supervised Learning

In supervised learning the class labels are supplied as a training data set which is used to build up a model. A signal therefore exists that is used to train and test the classifier. The most common supervised learning techniques are artificial neural networks, support vector machines and naïve Bayes [27].

The statistical analysis of class prediction is an important problem involving gene expressions and has received considerable attention in recent times [27]. The problem is essentially to predict the diagnostic category of the patient using their gene expression profile. The challenges of supervised learning in the context of microarray studies are:

1. The gene or feature space is large.
2. The number of samples or training set is relatively small.
3. Finding feature genes, without bias, that contribute significantly to the different classes.

A brief discussion on the three main types of classifiers is given in order to illustrate the similarities and processes involved in supervised learning methods.

2.5.1 Artificial neural networks

A data set can be used to train a classifier using algorithms such as gradient descent or particle swarm optimization. The most common and best performing architecture is usually an adaptation of the artificial neural network (ANN) [28]. An illustration of a generic layered artificial neural network is shown in figure 2.8.

The number of hidden layers add complexity to the network, with the overall size of the network related to the dimensions in the sample space [29]. The artificial neural network, and its numerous variations, have been proven to be a valuable network for the classification of gene expressions [30].

The use of stochastic networks is also an attractive field, and includes the Hopfield model and the Boltzmann machine [29]. The computing units are treated with stochastic behaviour such that each unit computes at different times, with the result of computation provided at a variable amount of time [29]. The minimization of an energy function can also be used to express the solution of the problem domain. The introduction then of thermal noise into these types of networks can prevent the network from reaching local minima.

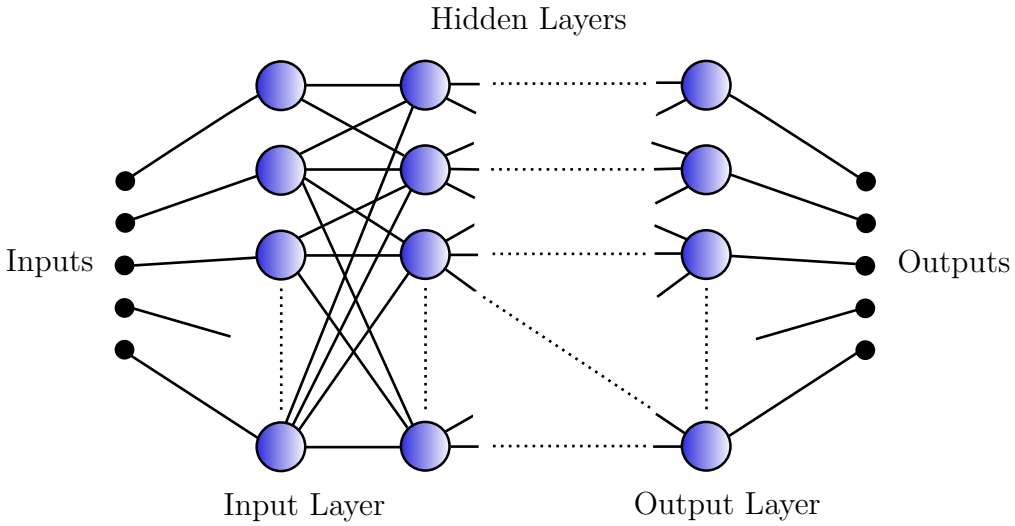


Figure 2.8: A generic layered ANN, adapted from Rojas [29].

2.5.2 Support vector machines

Considering a gene expression experiment in which there are n genes and m experiments, a m-element vector $\mathbf{x}_i$ can be constructed for the ith gene, where $1 \leq i \leq n$. A binary classifier simply constructs a hyperplane which separates class members and non-class members. The problem is that in most circumstances a hyperplane, that separates the two classes, does not exist [31]. A solution to this inseparability problem is to map the data to higher-dimensions and construct a plane in that space [31].

The drawbacks associated with mapping the data to a higher-dimensional (feature) space include the problem becoming computationally expensive and trivial solutions overfitting the data. Support vector machines however overcome these problems by choosing a maximum margin separating plane that prevents overfitting. Also by using a decision function that only uses vector dot products the requirement of defining the vectors explicitly in the high-dimensional space is removed [31].

The support vector machine consists of a kernel function, which acts as the dot product in the feature space, and a stringency parameter for the margin of the separating plane [31]. The parameters are usually dependent on the data set. The simplest kernel function that can be defined for genes $\mathbf{x}$ and $\mathbf{y}$ [31], is given by

$$k(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}, \tag{14}$$

$$= \sum_{l=1}^{m} x_l y_l. \tag{15}$$

The kernel function defined in equation 14 is adjusted to $k(\mathbf{x}, \mathbf{y}) + 1$, for technical rea-

sons [31]. The kernel can also be raised to a general power d to obtain a polynomial separating surface, and is given as

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d . \tag{16}$$

The radial kernel, using the Gaussian function, is also commonly used and is defined as

$$k(\mathbf{x}, \mathbf{y}) = e^{\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{2\alpha^2}}, \tag{17}$$

where α is the width of the Gaussian. A kernel matrix can be defined $\mathbf{K}$ which is usually modified to alleviate erroneous classifications produced from noise in the data [31]. The noise can be reduced by shifting the diagonal of the matrix by some constant, as shown in the following

$$\mathbf{K}_{ij} := \mathbf{K}_{ij} + \mathbf{I}\lambda \left(\frac{n^+}{N} \right), \tag{18}$$

where n^+ is the number of positive training examples, N is the total number of training examples and λ is a scale factor [31]. The support vector machine has been shown to be successful at classifying gene expressions, however, in the case of Brown (et al) the classes that were selected prior to analysis formed clusters in the input space [31]. The process is therefore open to various sources of bias which should be illustrated and examined.

2.5.3 Naïve Bayes networks

The naïve Bayes is a probabilistic classifier as opposed to neural networks and support vector machines which are both deterministic. The naïve Bayes classifier is based on Bayes' theorem with naïve independence assumptions [32], as shown by the following

$$P(C|F_1, ..., F_n) = \frac{1}{Z} P(C) \prod_{i=1}^{n} P(F_i|C), \tag{19}$$

where Z is a normalisation factor that depends on the features F_i, $1 \leq i \leq n$, and C being the dependent class variable. The naïve Bayes classifier maximises the conditional probability defined in equation 19. The classifier is able to avoid the high-dimensional feature space as it only uses the products of the assumed independent features.

2.6 Unsupervised Learning

The different types of machine learning generally include supervised learning, reinforcement learning and unsupervised learning [33]. In supervised learning the correct output is supplied along with the input data to the network, with the error between the sets used for adjusting the weights of the network. In reinforcement learning the machine is rewarded or punished based on the actions that affect the state of its environment. In unsupervised learning there are neither target outputs nor rewards from the environment.

The goal of a network that undergoes unsupervised learning is to build representations of the input space [33]. The representations or patterns of the input space can be used for predicting future inputs and decision making. An example of unsupervised learning includes dimensionality reduction, which is usually implemented using neural networks such as the multilayer perceptron [33].

The most frequently used multivariate technique to analyse gene expression data is the field of unsupervised learning called *clustering* [4]. Clustering problems consist of finding data points that are similar to one another in their cluster and dissimilar to points in other clusters [5]. The main advantage of clustering is that no *a priori* knowledge is needed about the data.

The classic types of clustering algorithms are categorised into hierarchical and non-hierarchical clustering. The hierarchical clustering algorithms group the objects through an agglomerative or divisive process and provide a natural graphical representation of the data, called a dendrogram [4]. The non-hierarchical algorithms partition the objects into k clusters, such that objects in the same cluster are similar compared to objects in other clusters [5].

The complex type of clustering algorithms include those that do not define a hard or crisp membership for the objects belonging to a cluster. An example of a complex clustering algorithm is fuzzy c-means which relates each object to a given cluster centroid with different degrees of membership [5]. The other form of complex clustering is called probabilistic or model-based clustering in which the data are assumed to be a mixture of underlying probability distributions, such as Gaussian mixture models [5]. The limitation however of probabilistic clustering is that is relies on the assumption that the dataset follows a specific distribution, which may not always be true [34].

In the context of gene expression data, clusters can be formed using the genes or the samples that exhibit similar behaviours or patterns. In gene-based clustering the genes are treated as objects and the samples as features in contrast to sample-based clustering.

Gene-clustering offers further insight and understanding of gene function, gene regulation and cellular processes of an organism [5]. Sample-based clustering can be used to distinguish between samples that are possibly indistinguishable using classical morphological approaches [5].

The underlying principal for any unsupervised or clustering technique is that only the data is used to measure the similarity between points. A similarity measure is therefore usually defined which provides a quantitative relationship between data points. The stronger this relationship the more likely the data points belong in the same cluster. The similarity measures can usually operate in the high-dimensional spaces associated with microarray experiments but have limitations regarding outliers and noise.

2.7 Distance Metrics

In order to measure the similarity between groups of genes or samples, a measure of similarity or distance needs to be defined. The various measures can be defined for several types of data such as: numerical, categorical, binary and mixed-typed data. The data type obtained from microarray experiments is numerical and as such the focus is on the numerical distance measure.

A metric follows an axiomatic definition, such that a distance $d(\cdot, \cdot)$ between two vectors, say $\mathbf{x}$ and $\mathbf{y} \in \mathbb{R}^n$, has the following properties:

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$ $d(\mathbf{x}, \mathbf{y}) = 0$ i.f.f $\mathbf{x} = \mathbf{y}$,
2. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$,
3. $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$.

2.7.1 Euclidean distance

The Euclidean distance is the most common and widely used distance measure for numerical data [6]. The Euclidean distance between vectors $\mathbf{x}$ and $\mathbf{y}$ elements of a n-dimensional real space $\mathbb{R}^n$ is defined as

$$d_{euc}(\mathbf{x}, \mathbf{y}) = \left[\sum_{i=1}^{n} (x_i - y_i)^2 \right]^{\frac{1}{2}}. \tag{20}$$

2.7.2 Manhattan distance

The Manhattan distance, also known as the city block distance, considers the distance between the two vectors, $\mathbf{x}$ and $\mathbf{y} \in \mathbb{R}^n$, to be the sum of all the vector attributes or components [6]. The Manhattan distance is defined to be

$$d_{man}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i| . \tag{21}$$

2.7.3 Maximum distance

The maximum distance, or supremum distance, is defined to be the maximum of the distances between elements $\mathbf{x}$ and $\mathbf{y} \in \mathbb{R}^n$ [6], as shown by

$$d_{max}(\mathbf{x}, \mathbf{y}) = \underbrace{\max}_{1 \leq i \leq n} |x_i - y_i| . \tag{22}$$

2.7.4 Minkowski distance

The distance measure can be generalised to a to the Minkowski distance [6], such that for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ [6] the Minkowski distance is defined as

$$d_{min}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^r \right)^{1/r}, \tag{23}$$

where $r \geq 1$ and is called the order of the Minkowski distance. When $r = 2,\ 1$ and ∞ the distance is simply the Euclidean, Manhattan and maximum distance respectively. The distance has been found work suitably for clusters that are isolated and compact, however in other situations the performance is often degraded by large scale attributes [6].

2.7.5 Mahalanobis distance

The Mahalanobis distance is often used to reduce the distortion created by the linear combinations of attributes [6]. The distance is defined as

$$d_{mah}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y}) \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \mathbf{y})^T}, \tag{24}$$

where $\mathbf{\Sigma}$ is the covariance matrix of the data set, and is given by, for d objects in data set V with n variables $\nu_1, \nu_2, ..., \nu_n$, as

$$\mathbf{\Sigma} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix} \tag{25}$$

where for variables j and k the covariance is given by

$$c_{jk} = \frac{1}{d} \sum_{i=1}^{d} \left(\nu_{ij} - \bar{\nu}_j\right) \left(\nu_{ik} - \bar{\nu}_k\right), \tag{26}$$

with the mean of the pth variable given by

$$\bar{\nu}_p = \frac{1}{n} \sum_{i=1}^{d} \nu_{ip}. \tag{27}$$

The Mahalanobis distance therefore uses a weighted scheme applied to the data, which reduces the distortion in most instances [6]. The Mahalanobis distance however requires high computation as the covariance matrix is computed using all of the data points [6].

2.8 Gene Expression Data Clustering

The methods for selecting feature genes are generally biased supervised techniques since they require the class descriptors *a priori*. As such the feature selection methods produce results that are favourable only when the information about each sample is known. A problem exists however when it is required to determine the features or new partitions of a large data set when no *a priori* information is provided. The unbiased and unsupervised approach is commonly known as clustering which involves discovering the inherent structure in the data.

The purpose of cluster analysis is to discover and investigate relationships between objects in order to decide if the data can be represented by a small number of clusters of similar objects [21]. The objective of clustering is to assign the n objects in the data set such that objects in the same group are as similar as possible, while being dissimilar to objects of different groups [21].

The problem of clustering can be stated more formally as: given n data points $\mathbf{x}_i$, where $i = 1, 2, ..., n$, in a d-dimensional space, identify the underlying structure of the data [3].

The result of clustering is to partition the n data points into m clusters such that points in a cluster are more *similar* compared to points in other clusters. The problem in clustering is that the similarity measure is selected arbitrarily and the results depend on the resolution at which the data set is viewed.

Cluster analysis techniques can be used to identify biologically relevant groups of genes and samples, and also provide insight into the function and regulation of genes [21]. The two main types of clustering in the context of microrarray data are called *gene-based clustering* and *sample-based clustering*. In gene-based clustering the genes are treated as objects and the samples as features, whereas in sample-based clustering the opposite occurs [34].

2.8.1 Gene-based clustering

The purpose of gene-based clustering is to cluster co-expressed genes which indicate co-function and co-regulation [34]. The requirements for gene-based clustering are specified as:

1. Cluster analysis is the first step in data mining and knowledge discovery. Prior knowledge is therefore usually limited and a clustering algorithm that estimates the correct number of clusters without requiring the number of predetermined clusters is favoured.
2. The noise in microarray experiments is large and an algorithm should be capable of extracting as much useful information from a high level of background noise as possible.
3. Provide some graphical representation of the cluster structure.
4. Extract interconnected gene expression profiles.

In some sense the large number of genes used in a microarray experiment makes the clustering task easier as apposed to clustering only a few samples. The problem however is that a large proportion of the gene sets used are randomly distributed across the samples, which makes the clustering task challenging.

2.8.2 Sample-based clustering

The gene-expression profiles obtained from microarray experiments are usually associated with several particular macroscopic phenotypes [34]. The purpose of sample-based clus-

tering is to find the phenotype structures or substructures of the microrarray samples [34]. The large number of genes and inherent noise requires a reduction in dimensionality or choosing a subset of genes to discriminate the classes for good quality and reliable results. The genes used for this purpose are called *informative genes* [34]. The two main categories for selecting informative genes fall into *supervised analysis* and *unsupervised analysis* [34].

The supervised approach is similar to the feature selection methods covered in section 2.3.2. In supervised gene selection the samples are labelled according to the phenotype attached on the sample. The major steps involved in training a classifier of this nature include [34]:

1. Selecting a training sample set, which is normally the same size as the original sample set.
2. Informative gene selection, such that the selected genes with their expression profiles can distinguish different phenotypes. Methods include neighbourhood analysis, support vector learning and ranking-based methods.
3. Sample clustering of all the samples using the selected informative genes as features.

In the case of unsupervised sample-based clustering the information connecting the phenotype and the sample does not exist. Unsupervised clustering offers a premise of automatically discovering the phenotypes of samples and the discovery of unknown substructures in the sample space [34]. The following two challenges, as described by Jiang et al, make it difficult to detect phenotypes of samples and select informative genes [34]:

1. The number of samples is relatively low compared to the number of features (genes), which makes the high-dimensional space very sparse. Detecting distinct class structures becomes difficult especially for density-based approaches.
2. A large proportion of the genes measured are irrelevant and contribute significantly to the noise in the data. About 10% of the genes measured are informative, but are covered in the large noise component of the data set. The selection of informative genes in such data is therefore difficult and remains a challenging task.

An interesting solution to the challenges stated is to realise that the processes in feature selection and sample clustering are in fact *interrelated* [34]. The first step in *interrelated clustering* is to partition the data set using a clustering algorithm, and then use a ranking method, such as the t-statistic, to score each gene according to their relevance to the partition [34]. The top ranking genes are then selected and the process repeated, with each iteration converging to the true sample structure and the final subset of genes being

the best group of informative features.

2.9 Clustering Algorithms

The most common algorithms used to cluster gene expression data are presented in order to understand their strengths and drawbacks. The algorithms can be used to cluster both the genes or samples depending on what information is required from the experiment. The algorithms covered constitute a variety of clustering types such as partitional and hierarchical algorithms. The algorithms also can be defined in terms of how the data points belong to each cluster i.e. crisp and soft clustering. The three conventional hard or crisp algorithms are hierarchical clustering, k-means and the self-organising map (SOM). A soft or complex type of clustering algorithm includes the fuzzy c-means in which each data point belongs to a cluster with a certain degree of membership.

2.9.1 Hierarchical clustering

The main principle behind hierarchical clustering is to group the data into a tree structure through either a divisive or agglomerative process [5]. In agglomerative clustering, called a bottom-up approach, each data point is initially its own cluster and subsequently each datum is merged based on the pairwise distance until there is a single remaining cluster. In divisive clustering all the data points start in the same cluster and divided until each datum forms its own cluster. An illustration of the two types of hierarchical clustering is shown in figure 2.9.

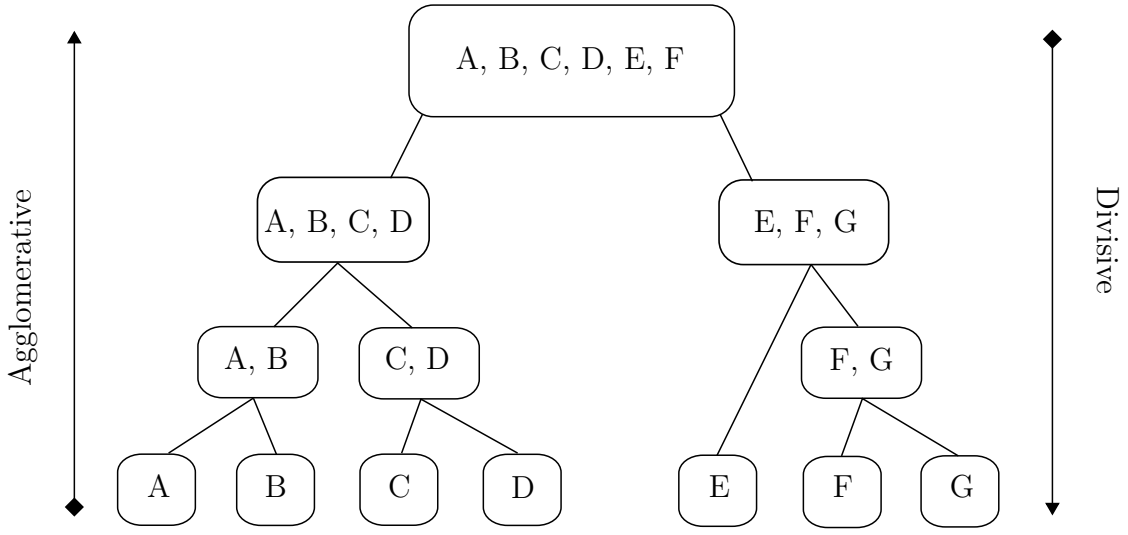


Figure 2.9: The two paradigms of hierarchical clustering, adopted from [5].

The agglomerative approach is based on a linkage metric in which there are three types: single, complete and average linkage. The single linkage metric determines the distance between two clusters as the distance between their closest elements [5]. The complete linkage metric uses the largest distance between the members of two clusters. In average linkage the cluster centroids are used to determine the distance between clusters. The divisive approach generates $(2N - 2)$ possible two-subset divisions for a single cluster with N objects, and is therefore too computationally expensive in most applications [5].

The hierarchical representation of the clustering results allows for the recognition and visualisation of any global patterns in the expression profiles [5]. The hierarchical clustering approach however has several issues such as robustness to noise, high-dimensionality and sensitivity to outliers [5]. The algorithm is also vulnerable and computationally expensive for large data sets, which are common in the analysis of gene expressions. The two approaches in hierarchical clustering, agglomerative and divisive, also follow a greedy strategy which does not allow for the refinement of clusters once they have been discovered [5].

2.9.2 *k*-means

The k-means is a commonly used partitioning algorithm, and has been used extensively in clustering gene expression data. The steps involved in the algorithm are outlined as follows:

Algorithm: k-means

1. k data points are randomly chosen as the cluster centroids.
2. The remaining data points are assigned to the closest centroid.
3. The new cluster mean or centroid is calculated and the algorithm reassigns the data points.
4. The algorithm then terminates when there is no significant change in the cluster boundaries.

The k-means algorithm therefore minimises the distance between the data points and the selected number of k centroids, as illustrated in figure 2.10. The objective function that the k-means algorithm minimises is the within-cluster sum-of-squares W, and is defined

as

$$W = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \bar{\mathbf{x}}_i\|^2, \tag{28}$$

where

$$k = \text{Number of clusters},$$
$$\bar{\mathbf{x}}_i = \text{Mean of cluster } C_i.$$

The advantages of k-means is that it can converge to the local optimum in only a few iterations and is therefore efficient for large data sets [5].

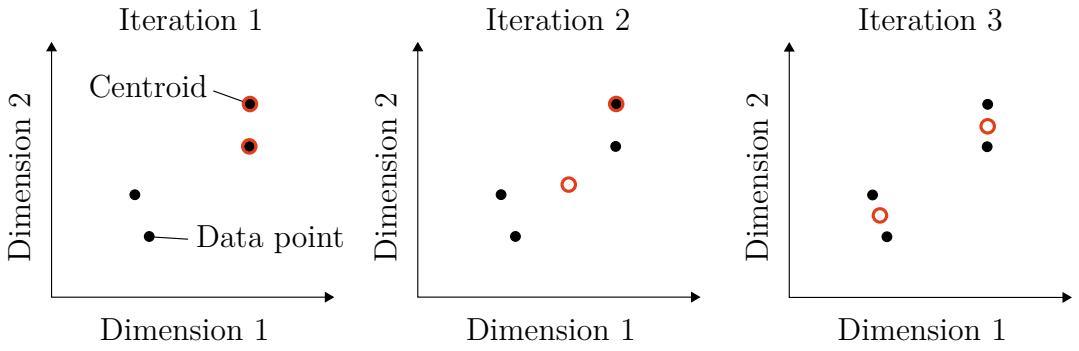


Figure 2.10: Illustration of the iterative steps involved in the k-means algorithm.

The major drawback of the k-means algorithm is that the user is required to specify the parameter k i.e. the number of expected clusters. The other disadvantage is that the intialisation process for the centroids is random which could lead to different results when the algorithm is repeated on the data set. The lack of information about gene expression data usually results in the user iteratively applying k-means to analyse the clusterings in order to determine the optimal parameter for k. The k-means algorithm is also susceptible to outliers and noise, since the centroid is calculated as the mean in the algorithm.

2.9.3 Self-organising map

The self-organising map (SOM) is the most commonly known unsupervised neural network learning algorithm, and was first developed by Kohonen in 1984 [5]. A Kohonen network or self-organising map is a type of clustering tool. The SOM divides the input patterns into similar groups without any *a priori* knowledge of the correct output and is thus identical to other clustering algorithms in this regard.

The SOM, however, unlike k-means and hierarchical clustering conveys information about the relationships and original positioning of the input patterns [4]. The architecture is unique in that it is able to reduce the dimensionality of the problem whilst maintaining the topology of the input space [4]. An illustration of a two dimensional SOM is shown in figure 2.11.

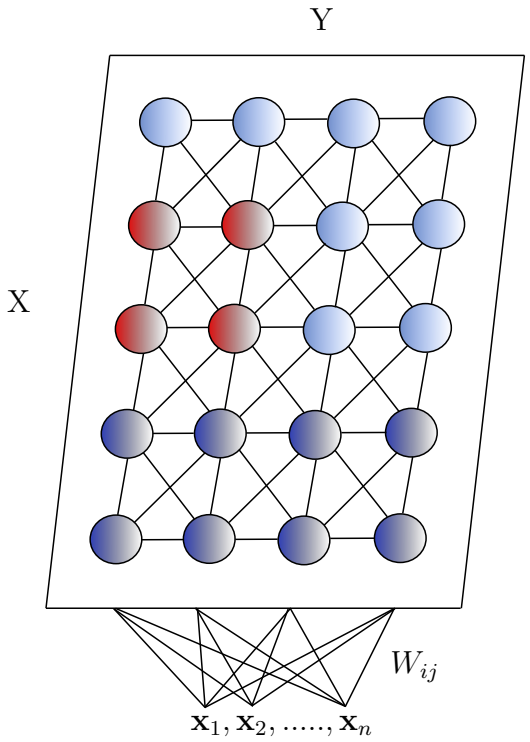


Figure 2.11: A two-dimensional self-organising map.

The SOM consists of a $n \times m$ grid of units (neurons), with each unit μ connected to all of the input vectors and represented by the prototype vector $\mathbf{p}_\mu = [p_{\mu 1}, ..., p_{\mu d}]$. The input connection has an associated weighting, which is used to determine the similarity between the unit and the input. A neuron is activated if the distance between its input and weight vector is minimal, with the Best Matching Unit (BMU), denoted $\mathbf{p}_b$, calculated from

$$\|\mathbf{x} - \mathbf{p}_b\| = \underbrace{\min}_{\mu} \|\mathbf{x} - \mathbf{p}_\mu\| \,. \tag{29}$$

The weighting of the activated neuron is adjusted and is proportional to the difference between the input and initial weighting. The neighbouring units are also adjusted, however, the distance between the neighbouring unit and the activated unit is factored into the amount of weight correction, as discussed by Vesanto and Alhoniemi [35], and given

as

$$\mathbf{p}_\mu(t+1) = \mathbf{p}_\mu(t) + \alpha(t) h_{b\mu}(t) \left[\mathbf{x} - \mathbf{p}_\mu(t)\right], \tag{30}$$

where

$$\begin{aligned} t &= \text{Time}, \\ \alpha(t) &= \text{Learning coefficient}, \\ h_{b\mu}(t) &= \text{Neighbouring kernel centred on BMU}. \end{aligned}$$

The training is completed when the adjustment of the weight vectors is diminished and the clusters are then identified by mapping the data points to the output neurons, shown in figure 2.11. The propagation of the neurons to the distributed clusters is shown in figure 2.12. The SOM units, which are nearest to a specific cluster, are attracted to the centre of that cluster until there is negligible change in the SOM structure.

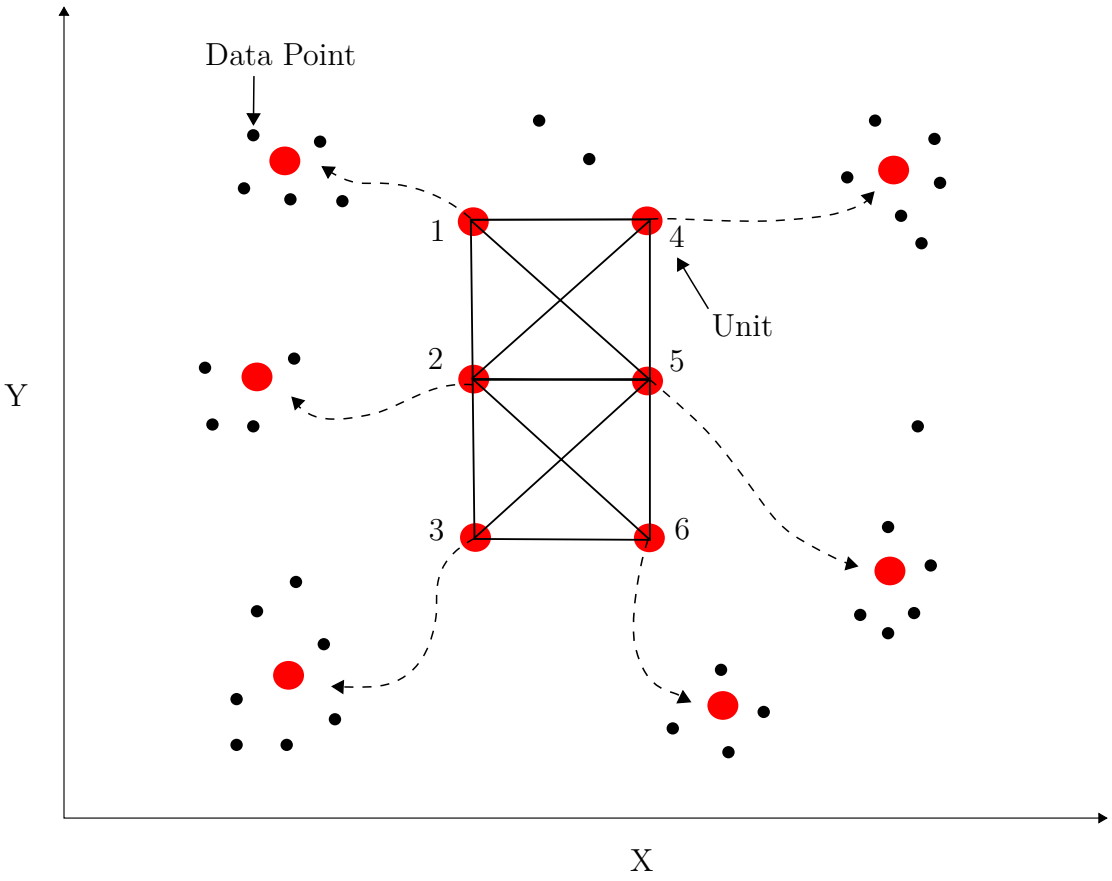


Figure 2.12: Rearrangement of the SOM units during trainging phase.

The benefit of the self-organising map, being a vector quantised approach, is that the high-dimensional data of gene expressions can easily be mapped to a lower dimension and

visualised [5]. The neural network type architecture also makes the SOM more robust to noise in the data.

The disadvantages of the SOM are similar to that of k-means as the user must specify the number of clusters as well as the topology of the neurons. The SOM can also converge to a local optimum rather than the global optimum if the initial weighting of the neurons is not correctly selected [5]. The reason for suboptimal convergence is a result of the unspecified values selected for the learning rate and topology of the neurons.

2.9.4 Fuzzy *c*-means

The fuzzy c-means algorithm is the most popular soft type clustering algorithm for gene expression analysis [5]. The fuzzy c-means algorithm uses a similar approach to the k-means algorithm in optimising the cluster centroids. In fuzzy c-means, however, each gene is considered a member of all the clusters with varying amounts of membership. The membership is closely related to the distance or similarity measure between a gene and a given cluster centroid [5]. The closer a gene is to a given cluster centre the closer the membership is to 1, otherwise the value is closer to 0.

The parameters of the fuzzy c-means algorithm include the number of predefined clusters c, fuzzification parameter $1 \leq m < \infty$ and a small positive number ϵ [5]. The algorithm iteratively updates the membership matrix and centroid positions until the change in the memberships are less than ϵ.

The fuzzy c-means algorithm is also based on the minimisation of an objective function similar to that of equation 28. The difference however of the fuzzy c-means objective function to equation 28 is a weighting or membership term for each data point and its corresponding cluster. The iterative steps involved in the fuzzy c-means algorithm are described as follows:

Algorithm: Fuzzy c-means

1. Initialise membership matrix $U = [u_{ij}]$.
2. Calculate the centroid vectors $\mathbf{c}_j$ using the following equation

$$\mathbf{c}_j = \frac{\sum_{i=1}^{N} u_{ij}^m \mathbf{x}_i}{\sum_{i=1}^{N} u_{ij}^m}. \tag{31}$$

3. Update the membership matrix U, using

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|\mathbf{x}_i - \mathbf{c}_j\|}{\|\mathbf{x}_i - \mathbf{c}_k\|} \right)^{\frac{2}{m-1}}}. \tag{32}$$

4. If $\|U(k+1) - U(k)\| < \epsilon$ then terminate, else go to step 2.

The advantage of the fuzzy c-means algorithm is that it can uncover the complex relationships between genes in regulatory pathways. The disadvantage of the fuzzy c-means algorithm is the fact that it is a local heuristic search algorithm that is likely to get trapped in local sub-optima [5].

2.10 Summary

The biology, technology and statistical methods used for understanding and analysing gene expression data have been covered. The feature gene selection methods although important for supervised learning techniques, such as the support vector machine, do not factor into unsupervised cluster analysis. The focus of this research is to test the various clustering algorithms on gene expression data obtained from oligonucleotide microarrays and compare it to the developed diffractive clustering algorithm.

The various microarray visualisation techniques have been covered in order to understand the different representations of gene expression data and the clustering results. The plots that are of importance in the study pertain to the clustering analysis of gene expression data, and plots that are independent of time, such as the scatter plot. The objective of this study is to overcome the challenges presented by unsupervised clustering analysis i.e. to overcome the high-dimensionality and develop a clustering algorithm that is not susceptible to noise.

The various clustering algorithms that are used for comparison include k-means which is a classical partitioning algorithm, hierarchical clustering, SOM which is a neural network type architecture and the fuzzy c-means algorithm which is a non-crisp clustering type algorithm. The aforementioned algorithms each have a unique quality which is integrated into the design of the diffractive clustering algorithm.

3 DATA STANDARDISATION AND TRANSFORMATION

Data standardisation and transformation are important aspects of gene expression cluster analysis. The distance metrics used in clustering algorithms are usually sensitive to and dependent on the scale, location and dimensionality of the data. The transformation process allows the high-dimensional data to be transformed to a lower more manageable space. The standardisation techniques include methods such as the z-score method, which normalises the data to unit variance and zero mean. Linear and non-linear transformation algorithms are also discussed with the ISOMAP algorithm selected and used in conjunction with the developed diffractive clustering algorithm.

3.1 Data Standardisation

Standardisation refers to the process of shifting and scaling the data to some dimensionless quantity [6]. Standardisation of the raw data is commonly performed before cluster analysis is undertaken. The metrics used to measure the dissimilarity between data points, an example being the Euclidean distance, are usually sensitive to differences in magnitude or scales of the input data [6].

The two main approaches to the standardisation of the raw data include: global standardisation and within-cluster standardisation [6]. The latter approach is more difficult to implement since the clusters are unknown prior to standardisation. An iterative solution to this problem, proposed by Klett, obtains clusters based on overall estimates, with the estimated clusters then used to determine the within-group variances for standardisation from a second cluster analysis [6].

It is noted in literature that the adopted standardisation procedure usually depends on the convention of the particular field of study [6]. The general formula to standardise the raw data set is shown in equation 33. The various standardisation methods are obtained by selecting the different definitions for the variables L_j and M_j for the standardisation equation, which is defined as

$$x_{ij} = \frac{x^*_{ij} - L_j}{M_j}, \tag{33}$$

where

$$
\begin{aligned}
i &= \text{Object } (1 \leq i \leq n), \\
j &= \text{Dimension } (1 \leq j \leq d), \\
x_{ij} &= \text{Raw data point}, \\
x^*_{ij} &= \text{Standardised data point}, \\
L_j &= \text{Location measure}, \\
M_j &= \text{Scale measure}.
\end{aligned}
$$

The most common normalisation methods are the mean, median, standard deviation, range, Huber's estimate, Tukey's biweight estimate and Andrew's wave estimate [6]. The different methods of normalisation are shown in table 3.1, where $\bar{x}^*_j$, R^*_j and σ^*_j are defined using the following equations

$$\bar{x}^*_j = \frac{1}{n}\sum_{i=1}^{n} x^*_{ij}, \tag{34}$$

$$R^*_j = \underbrace{\max}_{1\leq i\leq n} x^*_{ij} - \underbrace{\min}_{1\leq i\leq n} x^*_{ij}, \tag{35}$$

$$\sigma^*_j = \left[\frac{1}{n-1}\sum_{i=1}^{n}\left(x^*_{ij} - \bar{x}^*_j\right)^2\right]^{\frac{1}{2}}. \tag{36}$$

Table 3.1: Data normalisation methods, obtained from [6].

Method	L_j	M_j
z-score	$\bar{x}^*_j$	σ^*_j
USTD	0	σ^*_j
Maximum	0	$\underbrace{\max}_{1\leq i\leq n} x^*_{ij}$
Mean	$\bar{x}^*_j$	1
Median	$x^*_{\frac{n+1}{2}j}$ if n is odd $\frac{1}{2}\left(x^*_{\frac{n}{2}j} + x^*_{\frac{n+2}{2}j}\right)$ if n is even	1
Sum	0	$\sum_{i=1}^{n} x^*_{ij}$
Range	$\underbrace{\min}_{1\leq i\leq n} x^*_{ij}$	R^*_j

The first standardisation method in table 3.1 is the z-score, which is used to transform normal variants to the standard score form [6]. The z-score method normalises data

such that it has a mean of 0 and a variance of 1 which results in the location and scale information of the original data being lost. An important and commonly used standardisation involves the range of the data set, which is defined in equation 35. The range methods however have been found to be sensitive to outliers [6]. The other approach to standardisation involves using the rank of the data matrix as scores, which has been found to be more insensitive to outliers [6].

3.2 Data Transformation

Data transformation specifically deals with the whole data set as apposed to standardisation which concentrates on the individual data points. In microarray experiments the number of genes can exceed thousands, while the sample size is usually less than a hundred. The gene, or feature, space therefore has a large number of dimensions which makes it difficult to visualise the structure of the data set and uncover interesting patterns [21]. The solution is to capture as much of the variation in the data as possible whilst projecting the data to a lower, typically two or three, dimensional space.

3.2.1 Principal component analysis

Principal component analysis belongs to a class of dimensionality reduction techniques called *projection methods* [21]. Linear projection methods select one or more linear combinations of the original variables to maximise some measure of interest [21]. Linear discriminant analysis (LDA) is often employed as it maximises the ratio of the between-class scatter and within-class scatter [36], described in section 4. The problem with LDA is that the sample classes need to be known *a priori* which defeats using cluster analysis.

In the case of PCA the goal is to reduce the large dimensionality whilst retaining as much of the variation in the high dimensional space as possible. The principal components obtained from PCA are variables that are linearly dependent on the original variables but are uncorrelated, with the first few components capturing the most variation [6].

The derivation of the principal components follows that presented by Gan, Ma and Wu [6]. Defining $\mathbf{v} = (\upsilon_1, \upsilon_2..., \upsilon_d)^T$ to be a vector of d random variables. The initial step is to find a linear function $\mathbf{a}_1^T\mathbf{v}$ that maximises the variance, where $\mathbf{a}_1$ is a d-dimensional vector $(a_{11}, a_{12}, ..., a_{1d})^T$. The function $\mathbf{a}_1^T\mathbf{v}$ is defined as

$$\mathbf{a}_1^T\mathbf{v} = \sum_{i=1}^{d} a_{1i}\upsilon_i. \tag{37}$$

The next step is to find a linear function $\mathbf{a}_j^T\mathbf{v}$ that is uncorrelated to $\mathbf{a}_1^T\mathbf{v}, \mathbf{a}_2^T\mathbf{v}, ..., \mathbf{a}_{j-1}^T\mathbf{v}$ and has maximum variance. After d steps there will be be d linear functions that meet the imposed criteria. The j^{th} derived variable $\mathbf{a}_j^T\mathbf{v}$ is the j^{th} principal component. For $j = 1, 2, ..., d$ it can be shown that the j^{th} principal component is provided by $z_j = \mathbf{a}_j^T\mathbf{v}$, where $\mathbf{a}_j$ is an eigenvector of the covariance matrix $\mathbf{\Sigma}$ that corresponds to the j^{th} largest eigenvalue λ_j. The first procedure is to define the optimization problem formally, which is stated in the following

$$\max \left[var \left(\mathbf{a}_1^T\mathbf{v} \right) \right], \tag{38}$$

which is subject to $\mathbf{a}_1^T\mathbf{a} = 1$, and the variance of first principal component calculated from $var \left(\mathbf{a}_1^T\mathbf{v} \right) = \mathbf{a}_1^T\mathbf{\Sigma}\mathbf{a}$.

The technique of Lagrange multipliers is employed in order to solve the above optimisation problem, where λ is defined as the Lagrange multiplier. By rearranging equation 38 and multiplying the second term by λ, the result obtained is

$$\mathbf{a}_1^T\mathbf{\Sigma}\mathbf{a} - \lambda \left(\mathbf{a}_1^T\mathbf{a} - 1 \right). \tag{39}$$

Differentiating equation 39 with respect to $\mathbf{a}_1$ leads to

$$\mathbf{\Sigma}\mathbf{a}_1 - \lambda\mathbf{a}_1 = 0, \tag{40}$$

$$\left(\mathbf{\Sigma} - \lambda\mathbf{I}_d \right) \mathbf{a}_1 = 0, \tag{41}$$

where $\mathbf{I}_d$ is the $d \times d$ identity matrix.

The result obtained by solving the optimisation problem shows that λ_j is in fact the j^{th} eigenvalue of the covariance matrix and $\mathbf{a}_j^T$ is the j^{th} eigenvector of the covariance matrix $\mathbf{\Sigma}$, as shown in equation 41. The use of PCA to reduce the high dimensionality of the gene feature space is commonly employed before cluster analysis. It was shown however through an empirical study that PCA can produce misleading results and degrade the quality of the resulting clusters [8].

3.2.2 Singular value decomposition

Singular value decomposition (SVD) is another data transformation method that is widely used in data compression and is in fact identical to PCA [6]. The linear projection technique SVD is derived similarly to the method shown by Gan [6]. Defining $D = \{\mathbf{x}_1, \mathbf{x}_1, ..., \mathbf{x}_n\}$ to be a numerical data set in d-dimensional space. The data set D

can then be represented by an $n \times d$ matrix X, which is defined as

$$X = (x_{ij})_{n \times d}\,, \tag{42}$$

where x_{ij} is the j-component value of $\mathbf{x}_i$.

Defining $\bar{\mu} = (\bar{\mu}_1, \bar{\mu}_2, ..., \bar{\mu}_d)$ to be the column mean of matrix X, and for $j = 1, 2, .., d$ is given as the following

$$\bar{\mu}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij}. \tag{43}$$

The vector $\mathbf{e}_n$ is defined as a column vector of length n with all the elements equated to one. The SVD method then expresses $X - \mathbf{e}_n$ as the following

$$USV^T = X - \mathbf{e}_n \bar{\mu}, \tag{44}$$

where U is an $n \times n$ column orthonormal matrix, S is an $n \times d$ diagonal matrix containing the singular values, and V is a $d \times d$ unitary matrix. The columns of matrix V are the eigenvectors of the covariance matrix C of X, which is expressed as

$$C = \frac{1}{n} X^T X - \bar{\mu}^T \bar{\mu} = V \Lambda V^T. \tag{45}$$

The matrix C is a positive semidefinite matrix and as such has non-negative eigenvalues and orthonormal eigenvectors. The singular values are related to the eigenvalues and expressed for $j = 1, 2, ..., d$ as

$$s_j^2 = n\lambda_j. \tag{46}$$

The benefit of singular value decomposition is that it avoids using the explicit form of the covariance matrix, which becomes ill conditioned for a large number of features compared to samples [37]. The principal component scores Z can be obtained from the following expression

$$Z = US. \tag{47}$$

3.2.3 ISOMAP

The PCA and SVD methods for reducing the dimensionality of the feature space are both linear techniques i.e. they construct a lower dimensional space using a linear function of

the original higher dimensional space. A recent article written by Shi and Luo explored the use of a non-linear dimensionality technique on cancer tissue samples called isometric mapping (ISOMAP) [9]. The technique replaces the usual Euclidean distance with geodesic distance, which has the ability to capture and characterise the global geometric structure of the data [9]. The ISOMAP technique is also able to deal with non-linear relationships between data points as it is based on manifold theory [9].

The ISOMAP technique makes use of the geodesic distance between data points, which is the path of minimal curvature on the embedded manifold. The setup for the algorithm is based on a data matrix D with dimension $n \times m$, where the rows are co-ordinate vectors and approximate geodesic distances are calculated to estimate their relative relationship [9]. The three main steps of Tenenbaum's ISOMAP algorithm are:

1. Construction of the neighborhood graph $\mathbf{G}$, which is achieved by defining the neighbourhood points with k nearest neighbours, or within a radius ϵ. Connection of neighbouring points using the Euclidean distance as the edge length.

2. Computation of the geodesic distance for the pairs of points. The geodesic distance between two points $d_G(P_i, P_j)$ is the shortest path on $\mathbf{G}$. The geodesic distance for neighbouring points is the Euclidean distance, whereas for distant points it is calculated using the Dijkstra algorithm [38]. The geodesic matrix can be obtained using $D_G = [d_G^2(P_i, P_j)]_{i,j=1}^n$.

3. Construct the d-dimensional embedding by applying the multidimensional scaling (MDS) algorithm to the geodesic matrix D_G [39].

The steps of the ISOMAP algorithm are summarised next.

Algorithm: ISOMAP

Input:
Pairwise Euclidean distance, $d(P_i, P_j)$, between points in the input data matrix D. Neighbouring parameter k and embedding dimension p.

Output:
Reduced data matrix $n \times p$.

1. Assignment of k neighbours to data points.

2. Construct neighbourhood graph $\mathbf{G}$ by connecting neighbourhood points (P_i, P_j) with edge length $d(P_i, P_j)$.

3. Initialisation of geodesic distance $d_G(P_i, P_j) = d(P_i, P_j)$ for neighbouring pairs, otherwise $d_G(P_i, P_j) = \infty$.

4. Determine minimum path distance in D_G by replacing entries $d_G(P_i, P_j)$ by $\min\{d_G(P_i, P_j), d_G(P_i, P_l) + d_G(P_l, P_j)\}$, where $l = 1, 2, ..., n$.

5. Apply the MDS algorithm to D_G to obtain the p-dimensional embedding.

The ISOMAP algorithm was compared to the PCA dimensionality technique by Shi and Luo [9]. The main differences between the two methods are highlighted and summarised as follows

1. PCA uses the Euclidean distance as a similarity measure, whereas ISOMAP is a non-linear dimensionality reduction technique that uses the geodesic distance. The geodesic distance is a generalisation of the Euclidean distance on a high-dimensional manifold space. Gene expressions represent a high-dimensional and non-linear complex process, which is better represented by non-linear dimensionality reduction techniques such as ISOMAP.

2. PCA finds an orthogonal reduced space that is a linear combination of the original variables. The ISOMAP technique however preserves the global geometric structure of the gene expression data, which allows the inherent cluster structure to be maintained.

3. In terms of complexity the PCA technique has to compute and decompose the covariance matrix, and it was found by Shi and Luo that the computational time of ISOMAP was much faster than PCA [9].

4. PCA has no parameters which inhibits flexibility, whereas ISOMAP has two, the neighbourhood number and embedding dimension, that if chosen correctly allow for superior performance.

To illustrate the ability of the ISOMAP algorithm a three dimensional logarithmic spiral was constructed and shown in figure 3.1. The Euclidean distance (dashed line) between two points is different to the geodesic distance (solid line) along the manifold, of which the points are embedded, as indicated in figure 3.1.

The residual variance, defined as

$$e_p = 1 - R^2(D_G, D_\lambda), \tag{48}$$

is a measure of the error produced from dimensionality reduction. The standard linear correlation coefficient R, shown in equation 48, takes all the entries of the geodesic matrix D_G and D_λ, where the latter term represents the Euclidean distance matrix in the reduced p-dimensional space [40].

The change in the residual variance as the number of dimensions increases, shown in figure 3.2, provides information pertaining to the intrinsic dimensionality of the data. It is suggested by Tenenbaum et al, that the dimension at which the "elbow" of the residual variance curve is observed, is the correct dimension [41]. The elbow, or point of inflection, represents where the residual variance starts to decrease linearly instead of exponentially. The point of inflection also indicates where there is significant preservation of structure in the original space.

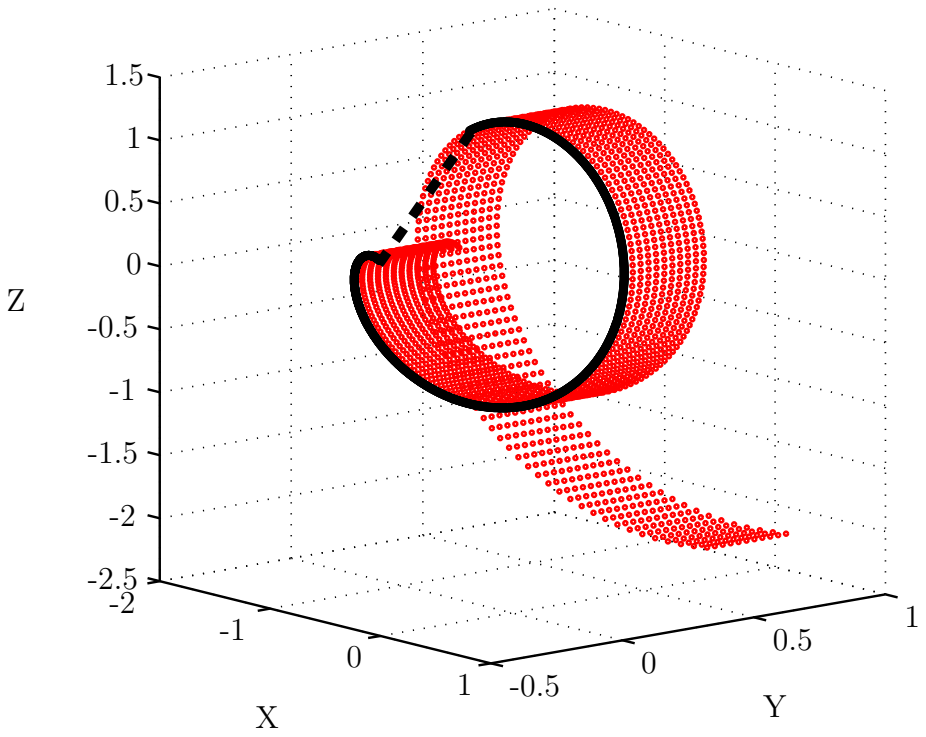


Figure 3.1: Logarithmic spiral illustrating properties of ISOMAP.

The ISOMAP algorithm can also be used to correctly determine the dimension to project the data onto from the non-linear manifold of which it is embedded. The spiral data set, shown in figure 3.1, is an example where the algorithm and the resulting change in residual variance indicate that the correct dimension to project the data is in fact two dimensions.

The ambiguity, however, still exists around the precise value to choose for the number of neighbours k or the radius ϵ. An iterative scheme could be implemented whereby the value of k, or ϵ, is changed and the resulting residual variance curve observed for distinct regions, such as the point of inflection.

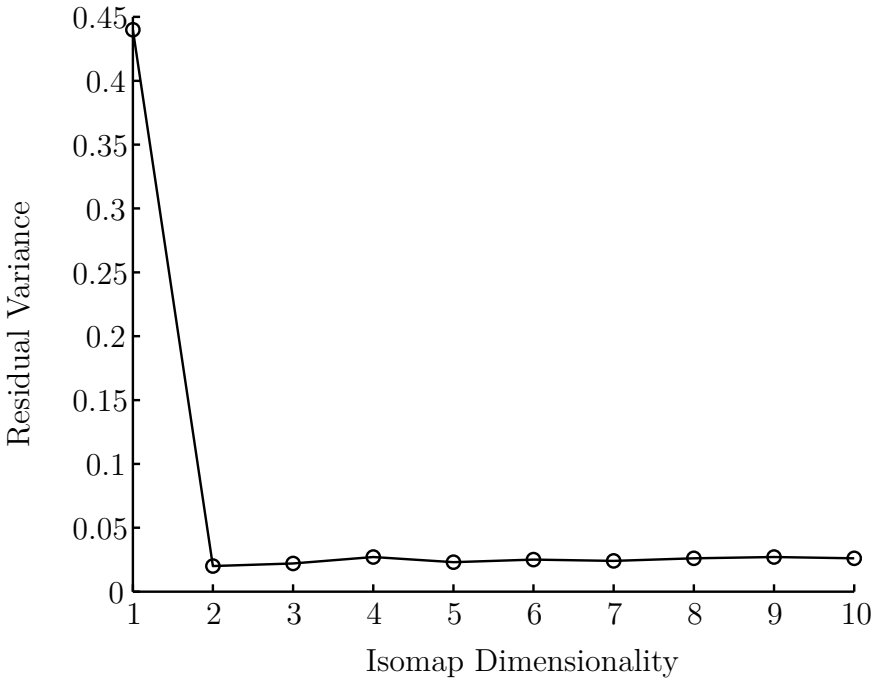


Figure 3.2: Residual variance of the ISOMAP algorithm as dimensionality increases.

3.3 Summary

The normalisation of data is important particularly in gene expression cluster analysis. The various techniques each have their advantages, such as the amount of sensitivity to outliers in the data set. The linear mapping techniques, examples being principal component analysis and singular value decomposition, play important roles in cluster analysis of gene expression data. Recent studies however have shown that a non-linear mapping technique called ISOMAP outperforms the classical linear mapping techniques.

The other benefit, although present in PCA, is that the ISOMAP residual variance curve can be used to detect the inherent dimensionality of the data set. The unsupervised clustering approach is coherent with this procedure as no *a priori* information is required for analysis. The ISOMAP algorithm is also able to conserve the non-linear geodesic distance between data points, which is suitable for analysing complex interacting genes in microarray data.

The overall benefit is that standard units are established for each dimension using the standardisation procedure, along with a solution to the curse of dimensionality by transforming the high-dimensional data to a lower-dimensional space. The reduced space complements the clustering analysis as it reduces the sensitivity and problems associated with distance metrics in high-dimensional spaces.

4 CLUSTER VALIDATION

The purpose of cluster analysis is to partition a data set into K distinct groups based on specific features. The data points within a group are selected such that they are more similar to each other than data points situated in different groups [34]. The number and type of clusters however can vary significantly for different clustering algorithms. The k-means algorithm for example depends on random initialization which can produce different types of clusters each time the algorithm is run. It is therefore important to validate the reliability and quality of the different partitions or sets of clusters produced by the various algorithms.

4.1 Internal Criteria

The purpose of using internal criteria is to evaluate the resulting clustering structure using the quantities and features inherent in the data set [6]. A common quantity that is used to measure the data set and validate the resulting cluster is the proximity matrix [42]. The cophenetic correlation coefficient can be used to measure the degree of similarity between the cophenetic matrix P_c and the proximity matrix P [42].

The cophenetic matrix is defined, using the similarity matrix, as the minimum distance at which each object merges during the clustering process. The coefficient is used to measure the reliability of the pairwise distances of a dendrogram produced by some hierarchical algorithm compared to the similarity matrix of the data. The (i, j) elements of P and P_c are given as d_{ij} and c_{ij} respectively. The range of the cophenetic correlation coefficient C is between -1 and 1, with large values indicating greater similarity between P and P_c, and is defined as

$$C = \frac{\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}c_{ij} - \mu_P\mu_C}{\sqrt{\left(\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}^2 - \mu_P^2\right)\left(\frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}^2 - \mu_C^2\right)}}, \tag{49}$$

where

$$M = \frac{n\,(n-1)}{2},$$
$$\mu_P = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij},$$
$$\mu_C = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}.$$

4.2 External Criteria

The external criteria approach evaluates the resulting cluster partition **C** by comparing it to a prespecified structure that is imposed on the data set **P** [6]. The partition **P** is usually defined from expert knowledge of the data set and the expected clustering structure. The two general approaches are

1. Compare the resulting cluster structure **C** to independent partition **P**.
2. Compare the proximity matrix **Q** to the partition **P**.

The first approach is used extensively in research when it is known what the clustering structure should be in a gene expression data set. The resulting clustering structure is defined as $\mathbf{C} = \{C_1, ..., C_m\}$, with the imposed partition defined as $\mathbf{P} = \{P_1, ..., P_s\}$. The following list of variables are used to calculate some of the commonly used indices, which are shown in table 4.1.

1. a is the number of pairs of data points which are in same cluster of **C** and in the same cluster of **P**.
2. b is the number of pairs of data points which are in same cluster of **C** but in different clusters of **P**.
3. c is the number of pairs of data points which are in different clusters of **C** but in the same cluster of **P**.
4. d is the number of pairs of data points which are in different clusters of **C** and in different clusters of **P**.
5. $M = a + b + c + d = \frac{n(n-1)}{2}$.

The range for the three indices in table 4.1 is $R, J, F \in [0, 1]$. The higher the values for these indices the greater the similarity between **C** and **P** [6].

Table 4.1: Common indices that measure the similarity between partitions **C** and **P**.

Index	**Formula**
Rand statistic	$R = \frac{a+d}{M}$
Jaccard coefficient	$J = \frac{a}{a+b+c}$
Folkes and Mallows index	$F = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$

The second approach compares the proximity matrix **Q** to the partition **P** [6]. The Γ statistic or normalised Γ statistic is computed using the information in the proximity matrix **Q** and the matrix **Y**, where **Y** is defined as

$$Y_{ij} = \begin{cases} 1 & \text{if } g(x_i) \neq g(x_j), \text{ for } i, j = 1, 2, ..., n; \\ 0 & \text{otherwise.} \end{cases}$$

The function g maps the data elements to the cluster number introduced by partition **P**. The Hubert's Γ statistic is defined to be

$$\Gamma = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij} Y_{ij}, \tag{50}$$

and is described in detail by Gan et al [6]. The elements X_{ij} represent the pairwise distances in the similarity matrix **Q**. The range for the Hubert's statistic is also $[0, 1]$, with large values indicating a strong correlation between the two matrices.

4.3 Relative Criteria

The fundamental idea of relative criteria is to select the best clustering result out of a set of defined schemes according to a predefined criterion [6]. If there are a set of parameters for a clustering algorithm, denoted P_{alg}, then the set of defined schemes is produced by using different parameters in P_{alg} [6]. The problem, as described by Gan et al [6], is divided into two parts depending on whether the number of clusters n_c is a parameter or not:

1. $n_c \notin P_{alg}$: The optimal choice for the parameters is determined by varying the parameters over a wide range, and then identifying the largest range where n_c remains constant i.e. has the longest lifetime.

2. $n_c \in P_{alg}$: The optimal choice for the number of clusters and algorithm parameters

are decided using the best value produced by a validity index. The validity index values are calculated by incrementing the cluster number and varying the algorithm parameters.

4.4 Cluster Validity Indices

The validation and measure of the quality of the clusters produced by an algorithm is usually achieved using a cluster index. The three fundamental criteria to investigate cluster validity are external criteria, internal criteria and relative criteria [6]. The first two approaches, external and internal criteria, depend on statistical testing which is computationally expensive, as apposed to using relative criteria. It was found in literature that some cluster validity indices also perform better for compact clusters, but not for arbitrary shaped clusters commonly discovered in gene expression data [6]. The four main types of indices that exist are the: Davies-Bouldin Index (I_B), Dunn's Index (I_D), Calinski Harabasz Index (I_C) and a recently developed index I [43].

4.4.1 Davies-Bouldin index

The Davies-Bouldin index measures the within scatter with respect to the between-cluster separation [43]. The scatter within a cluster S_i is defined as

$$S_i = \frac{1}{|C_i|} \sum_{x \in C_i} \|x - z_i\|_2 \,. \tag{51}$$

The distance between clusters is defined as d_i, and stated as

$$d_i = \|z_i - z_j\|_2 \,, \tag{52}$$

where z_i is the i^{th} cluster centre. The Davies-Bouldin index is the sum of the ratio between the scatter and cluster separation as shown by the following

$$I_B = \frac{1}{K} \sum_{i=1}^{K} R_{i,q}, \tag{53}$$

where $R_{i,q}$ is defined as

$$R_{i,q} = \underbrace{\max}_{j, j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{i,j}} \right\}. \tag{54}$$

The Davies-Bouldin index should be minimised as much as possible which is indicative of a proper clustering solution.

4.4.2 Dunn's index

The Dunn's index takes the ratio of cluster separation to the within cluster scatter. Let the i^{th} cluster be defined as C_i in the N-dimensional real space $\mathbb{R}^N$. The diameter Δ of a cluster is defined as

$$\Delta\left(C_i\right) = \underbrace{\max}_{x,y \in C_i} \left\{d\left(x,y\right)\right\}. \tag{55}$$

The separation distance δ between clusters is given by

$$\delta\left(C_i, C_j\right) = \underbrace{\min}_{x \in C_i, y \in C_j} \left\{d\left(x,y\right)\right\}, \tag{56}$$

where the variable $d\left(x,y\right)$ is the Euclidean distance between points x and y. The Dunn's index is stated as

$$I_D = \underbrace{\min}_{1 \leq i \leq K} \left\{ \underbrace{\min}_{1 \leq j \leq K, j \neq i} \left\{ \frac{\delta\left(C_i, C_j\right)}{\underbrace{\max}_{1 \leq k \leq K} \left\{\Delta\left(C_k\right)\right\}} \right\} \right\}. \tag{57}$$

The larger the Dunn's index the better quality the clusters, with the solution that maximises I_D taken to be the optimal number of clusters in the data set [43].

4.4.3 Calinski Harabasz index

The Calinski Harabasz index is defined similarly to the Dunn's index as the ratio of the cluster separation to the within cluster scatter. The index is calculated as

$$I_C = \frac{traceB/\left(K-1\right)}{traceW/\left(n-K\right)}, \tag{58}$$

for n data points and K clusters. The trace of the between and within cluster scatter matrices B and W are defined respectively as

$$traceB = \sum_{k=1}^{K} n_k \left\| z_k - z \right\|_2^2 , \tag{59}$$

$$traceW = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_i - z_k \right\|_2^2 . \tag{60}$$

The Calinski Harabasz index in complete form is

$$I_C = \frac{\left[\frac{\sum_{k=1}^{K} n_k \| z_k - z \|_2^2}{K-1} \right]}{\left[\frac{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \| x_i - z_k \|_2^2}{n-K} \right]}, \tag{61}$$

where the larger the Calinski Harabasz index value the better the clustering algorithm.

4.4.4 I index

The I index for K clusters is defined as follows

$$I = \left(\frac{1}{K} \times \frac{E_1}{E_K} \times D_k \right)^p , \tag{62}$$

where

$$E_K = \sum_{k=1}^{K} \sum_{j=1}^{n} u_{kj} \left\| x_j - z_k \right\|_2 , \tag{63}$$

$$D_K = \underbrace{\max}_{i,j=1}^{K} \left\| z_i - z_j \right\|_2 . \tag{64}$$

The factor p is used to control the contrast between different clusters [43]. The number K that maximises I is considered to be the correct number of clusters [43].

4.5 Summary

The main source of validation comes from external criteria, such as the rand index, which requires *a priori* classification of the data or samples that are to be clustered. The external criteria measure the similarity of the clustering solution between the actual classification and calculated results from the algorithm.

The other main type of cluster validation, which is based on relative criteria, relies on the inherent structure of the data and the clustering solution. The validity indices utilise metrics or distances of the within cluster scatter and the between-cluster separation. The validity indices each weight the distances differently, which gives rise to performance characteristics that depend on the type and shape of clusters within the data.

The number of clusters and the quality of the clustering solution can be determined using a validity index. The definition however of clustering remains ambiguous and as such each index value can differ significantly. The individual variation in results produced by the indices can be corrected by averaging the set of results.

The accuracy of the clustering results can also be calculated using the *a priori* classification of the samples. The accuracy is simply the difference between the actual clustering solution and the algorithm results. The accuracy closely corresponds to external criteria as both measure the similarity between the actual and algorithm results.

5 DIFFRACTIVE CLUSTERING

The following clustering algorithm derivation is the main contribution of the research and proposes a different view and solution for clustering sampled data. The algorithm with its derivation were formulated in the absence of any similar algorithms or contributing sources. The algorithm is based on the diffractive principals of light and the Fourier relationship found between the aperture and object image. The developed algorithm assumes that each data point is a point source or impulse of light which can be filtered as the aperture function is adjusted.

A hierarchy of filtered images results, for data that is represented in two dimensions, with the inherent structure of the data captured at a specific resolution. The algorithm has been designed to be complementary to each of the classical algorithms. The algorithm however is different in the sense that it does not require the predetermined number of clusters, which in most unsupervised applications is not specified.

5.1 Fundamental Theory of Diffraction

The underlying property of diffraction follows that if an opaque object is placed between a point light source and a white screen, the shadow cast by the object would not have perfect sharpness at the boundaries as predicted by geometrical optics [44]. The smearing effect observed at the boundaries of the shadow is closely related to the spreading of light after passing through a small aperture [44]. The overall name given to these observations and related behaviour of light is called *diffraction*. The famous *Fresnel-Kirchoff integral formula* is used to derive the relationship between the observed diffraction pattern and the aperture through which the light is passed [44].

The Fresnel-Kirchoff integral formula is obtained from Kirchoff's integral theorem, which is stated as

$$U_P = -\frac{1}{4\pi}\int\int\left(U\nabla_n\frac{e^{ikr}}{r} - \frac{e^{ikr}}{r}\nabla_n U\right)dA, \tag{65}$$

where ∇_n is the normal component of the gradient at the surface of integration. The theorem relates the value of any scalar wave function at any point P inside an arbitrary closed surface to the value of the wave function at the surface [44]. The Fresnel-Kirchoff

integral formula describes the scenario illustrated in figure 5.1, and is defined as

$$U_P = -\frac{ikU_0 e^{-i\omega t}}{4\pi} \int\int \frac{e^{ik(r+r')}}{rr'} \left[cos\left(\mathbf{n},\mathbf{r}\right) - cos\left(\mathbf{n},\mathbf{r}\right)\right] dA, \tag{66}$$

where the scalar wave function U represents spherical monochromatic waves traveling outward from the source S [44].

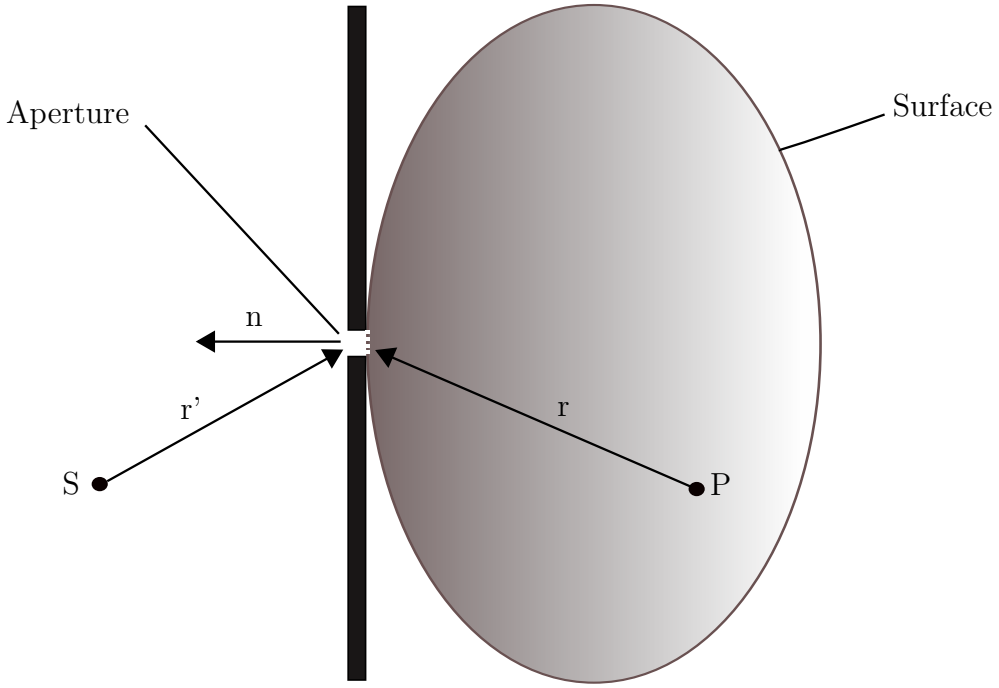


Figure 5.1: Geometrical setup for the Fresnel-Kirchoff formula.

The two main types of diffraction patterns that can occur are known as *Fraunhofer diffraction* and *Fresnel diffraction* [44]. In a qualitative sense Fraunhofer diffraction occurs when the incident and diffracted waves are approximately planar. The approximation is valid when the source and receiving point are both at large distances from the the aperture, as the curvatures of the waves can be neglected. The approximation however fails when the distances are small and the curvature of the wave is significant resulting in Fresnel diffraction. The Fraunhofer type diffraction is used for the rest of the analysis as it has a simpler mathematical representation.

5.1.1 Fraunhofer diffraction

The physical layout that is usually employed for observing Fraunhofer diffraction is illustrated in figure 5.2. The aperture is coherently illuminated from a monochromatic point source and a collimating lens [44]. The setup ensures that both the incident and

diffracted waves are planar. The following assumptions are made before applying the Fresnel-Kirchoff formula (equation 66) to the calculation of the diffraction patterns [44]:

1. The obliquity factor $[cos(\mathbf{n}, \mathbf{r}) - cos(\mathbf{n}, \mathbf{r}')]$ is negligible.
2. The quantity $\frac{e^{ikr'}}{r'}$ is approximately constant and is taken outside the integral.
3. The variation of the factor $\frac{e^{ikr}}{r}$ is approximately e^{ikr} with the term $1/r$ averaged out of the integral.

The consequence of applying the specified assumptions leads to a simplified version of the Fresnel-Kirchoff formula, as stated in the following

$$U_p = C \int \int e^{ikr} dA. \tag{67}$$

The assumptions produce factors in the integral which are lumped into the constant C. The simplified equation describes the resulting diffraction pattern as a result of the phase factor e^{ikr} integrated over the aperture.

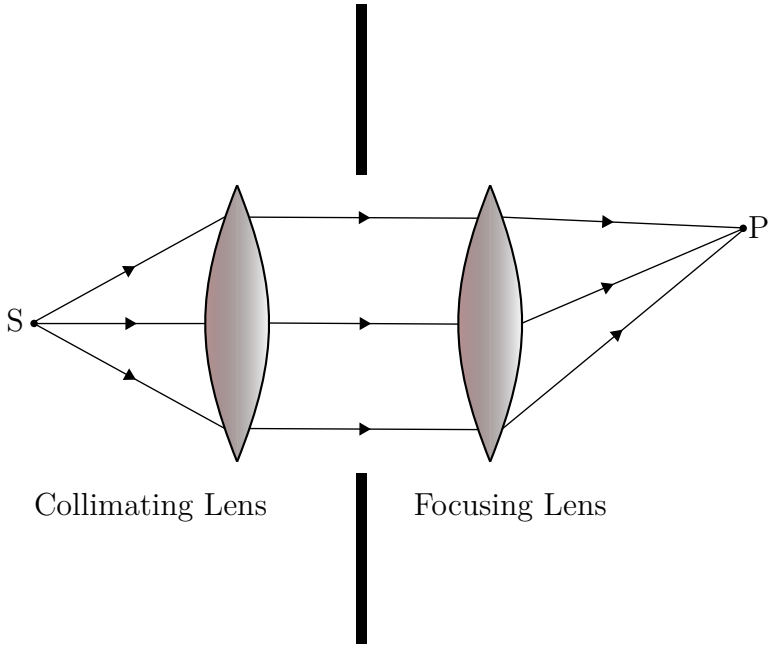


Figure 5.2: Physical setup for observing Fraunhofer diffraction.

5.1.2 The Fourier transform and diffraction

The next procedure is to consider a setup for an aperture with an arbitrary shape located in the x–y plane and the diffraction pattern produced at a point P on the X–Y plane, which is located at a relatively large distance. The rays are therefore assumed to be

parallel and are shown in figure 5.3 leaving the aperture at the origin O and Q in the x–y plane.

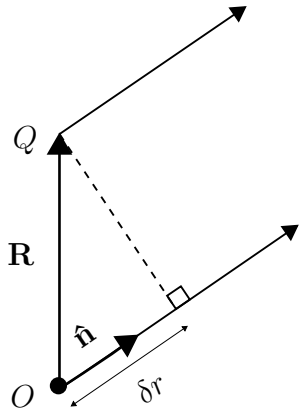


Figure 5.3: Two parallel light rays originating from points O and Q in the xy plane.

The path difference δr between the parallel rays is given by the component of vector $\mathbf{R} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}}$ in the direction of the unit vector $\hat{\mathbf{n}} = \alpha\hat{\mathbf{i}} + \beta\hat{\mathbf{j}} + \gamma\hat{\mathbf{k}}$, using the following

$$\begin{aligned} \delta r &= \mathbf{R} \cdot \hat{\mathbf{n}} \\ &= x\alpha + y\beta \\ &= x\frac{X}{L} + y\frac{Y}{L}. \end{aligned} \tag{68}$$

The direction of the rays are specified by the direction cosines α, β and γ. Using a focal length L for the lens, the X and Y coordinates can be approximated by $L\alpha$ and $L\beta$ respectively.

Using the fundamental diffraction integral, defined in equation 67, the intensity over the X–Y plane is given by

$$U(X,Y) = \int\int e^{ik\delta r} dA = \int\int e^{ik(xX+yY)/L} dxdy. \tag{69}$$

The previous equation holds for an uniform aperture, and by introducing the aperture function $g(x,y)$ can be extended for an arbitrarily shaped aperture. Using the definition for the aperture function and changing variables kX/L and kY/L into spatial frequencies μ and ν respectively, equation 69 can be rewritten in the following form

$$U(\mu,\nu) = \int\int g(x,y) e^{i(\mu x+\nu y)} dxdy. \tag{70}$$

The equation is important since it illustrates the Fourier transform relationship between

$U(\mu, \nu)$ and $g(x, y)$. The diffraction pattern, produced under the Franhofer conditions, is therefore simply the Fourier transform of the aperture through which the light is passed. The knowledge of this fact allows one to filter and reshape the diffraction pattern by altering the aperture. The technique is called spatial filtering and adjusts the diffraction pattern $U(\mu, \nu)$ by multiplying it by a transfer function $T(\mu, \nu)$ [44]. The resulting aperture function is obtained by taking the inverse Fourier transform, as demonstrated by

$$g'(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} T(\mu, \nu) U(\mu, \nu) e^{-i(\mu x + \nu y)} d\mu d\nu. \tag{71}$$

5.2 Formalisation of Clustering Algorithm

The formalisation of the clustering algorithm is based on the diffraction properties of light. The first step is to define a space, in this case a real d-dimensional Euclidean space $\mathbb{R}^d$. Defining a vector function $a(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^d$, to be an impulse function located at each datum point $\mathbf{p}_1, \ldots, \mathbf{p}_n$. The function $a(\mathbf{x})$ is defined and shown in equation 72

$$a(\mathbf{x}) = \sum_{i=1}^{n} \delta(\mathbf{x} - \mathbf{p}_i). \tag{72}$$

In terms of the diffraction setup $a(\mathbf{x})$ represents the aperture function, with the diffraction pattern $A(\boldsymbol{\xi})$ obtained using the Fourier transform over the aperture. Using the spatial filtering technique the aperture function can be altered (apodised) using a suitable filtering function $G(\boldsymbol{\xi})$. The result of the filtered diffraction pattern $Y(\boldsymbol{\xi})$ is given by

$$\begin{aligned} Y(\boldsymbol{\xi}) &= A(\boldsymbol{\xi}) G(\boldsymbol{\xi}), \\ &= \sum_{i=1}^{n} e^{-2\pi i (\boldsymbol{\xi} \cdot \mathbf{p}_i)} G(\boldsymbol{\xi}). \end{aligned} \tag{73}$$

Using the Fourier transform properties of a shifted delta function, the filtered aperture function is obtained

$$y(\mathbf{x}) = \sum_{i=1}^{n} g(\mathbf{x} - \mathbf{p}_i). \tag{74}$$

The chosen filter function is a d-dimensional Gaussian filter, which is represented as

$$G(\boldsymbol{\xi}) = e^{-\sigma \|\boldsymbol{\xi}\|_2^2}. \tag{75}$$

The choice of a Gaussian filter satisfies the criteria that the optimal (Wiener) filter must be the derivative of a symmetric low-pass filter under the assumption that the noise on the data set is white i.e. constant across all frequencies [19]. The Gaussian filter also fulfills the requirement on the form of the maximum likelihood solution, as described by Roberts [19]. The Gaussian function is variable separable and as such each dimension can be treated and filtered separately.

The Gaussian filter is justifiable as it is commonly used in scale-space and information theory where it satisfies important axioms for the former and maximises the entropy for a given covariance in the latter [45]. The Gaussian distribution for each data point maximises the entropy, which is stated as

$$H\left(y\right) = -\int_{\mathbb{R}^d} y(\mathbf{x}) log(y(\mathbf{x}))d\mathbf{x}. \tag{76}$$

The Gaussian kernel is also found to satisfy the diffusion equation which describes how point particles disperse with uncertainty in their locations [45].

The Fourier transform pair is taken to be unitary, which means that the inverse Fourier transform of a transformed function is the function itself. The definition for the d-dimensional Fourier transform pair used in the analysis is given by

$$\mathcal{F}\left\{f(\mathbf{x})\right\} = \int_{\mathbb{R}^d} f(\mathbf{x}) e^{i\mathbf{x}\cdot\boldsymbol{\xi}} d\mathbf{x}, \tag{77}$$

and

$$\mathcal{F}^{-1}\left\{F(\boldsymbol{\xi})\right\} = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} F(\boldsymbol{\xi}) e^{-i\mathbf{x}\cdot\boldsymbol{\xi}} d\boldsymbol{\xi}. \tag{78}$$

The inverse Fourier of the Gaussian filter is itself, a Gaussian function with the width determined by the inverse of the parameter σ, as stated by the following

$$g(\mathbf{x}, \sigma) = \frac{1}{(4\pi\sigma)^{\frac{d}{2}}} e^{\frac{-\|\mathbf{x}\|_2^2}{4\sigma}}. \tag{79}$$

The apodised aperture function, as a result of the Gaussian filter, is given by

$$y(\mathbf{x}, \sigma) = \frac{1}{(4\pi\sigma)^{\frac{d}{2}}} \sum_{i=1}^{n} e^{\frac{-\|\mathbf{x}-\mathbf{p}_i\|_2^2}{4\sigma}}. \tag{80}$$

The filtering operation on the diffraction pattern using the Gaussian function eliminates any spurious data points and smooths the aperture function, which allows for the association of the peaks of the aperture function with the cluster centroids. The peaks, or

cluster centres, of the filtered aperture function are found by taking the spatial derivative of equation 80 and setting the result to zero, as shown by

$$\frac{\partial y}{\partial \mathbf{x}} = \frac{1}{2\sigma\left(4\pi\sigma\right)^{\frac{d}{2}}} \sum_{i=1}^{n} (\mathbf{p}_i - \mathbf{x}) e^{\frac{-\|\mathbf{x}-\mathbf{p}_i\|_2^2}{4\sigma}} \;=\; \mathbf{0}. \tag{81}$$

5.2.1 Hierarchical diffractive clustering

As the parameter σ increases the width of the Gaussian filter decreases and as such higher frequencies in the measured data are removed. The resulting evolution of the parameter σ leads to the entire data set becoming a single cluster. The following theorem, as proved by Roberts [46], ensures that their is no splitting as σ increases. The theorem therefore satisfies the requirement that the number of turning points must not increase as the parameter σ increases.

Theorem: Define $\Phi(\sigma)$ to be the number of zero crossings at cutoff σ. The number of turning points of the function $y(\mathbf{x}, \sigma)$ at σ_1, $\Phi(\sigma_1)$, is greater or equal to the number of turning points at σ_2, $\Phi(\sigma_2)$, provided that $\sigma_1 < \sigma_2$.

The form of the dendrogram that results as the parameter σ evolves is similar to that of a hierarchical clustering method. The two main types of hierarchical clustering methods are nested and non-nested hierarchical clustering. In nested hierarchical clustering each data point at $\sigma = 0$ represents a single cluster. As σ increases the cluster centres merge and form new clusters, which in the nested case, are the union of the data points that belonged to the previous clusters. The nested form of hierarchical clustering has been critiqued by the fact that once a cluster is formed its members cannot subsequently be removed.

The non-nested form of hierarchical clustering eliminates the problem of nested hierarchical clustering by assigning each data point to the nearest centroid each time the parameter σ increases. The nested form of hierarchical clustering however has a more elegant and clear hierarchical structure, while the non-nested form has a more natural and consistent representation of the clustering procedure.

5.2.2 Numerical clustering solution

The presented clustering algorithm can be implemented by treating equation 81 as a gradient dynamic system, which is stated as follows

$$\frac{d\mathbf{x}}{dt} = \nabla_x y(\mathbf{x}, \sigma) = \frac{1}{2\sigma\left(4\pi\sigma\right)^{\frac{d}{2}}} \sum_{i=1}^{n} (\mathbf{p}_i - \mathbf{x}) e^{\frac{-\|\mathbf{x}-\mathbf{p}_i\|_2^2}{4\sigma}}. \tag{82}$$

The derivative can be approximated using the Euler difference method to solve the equation and find the solutions for the cluster peaks, which are located at points $\boldsymbol{\nu}_k$. The evolution of the solutions as the parameter σ changes is given by

$$\mathbf{x}[n+1] = \mathbf{x}[n] + h\nabla_x y(\mathbf{x}[n], \sigma), \tag{83}$$

where h is a small number that can be decreased each iteration to ensure convergence.

The determination of the cluster centres and association of the data points to those centres is achieved by comparing the length between two points to a small, arbitrarily chosen, positive number ϵ. A data point $\mathbf{x}[n+1]$ is considered a centre point if $\|\mathbf{x}[n+1] - \mathbf{x}[n]\| < \epsilon$. If two cluster centroids satisfy $\|\mathbf{x}_1 - \mathbf{x}_2\| < \epsilon$, then it is considered that the two centres have merged and form a single cluster centre.

5.2.3 Algorithm implementation

The implementation of the clustering algorithm deals with the discretisation of the parameter σ and an iterative scheme. The clustering algorithm is given as two types, non-nested and nested hierarchical clustering, as presented below.

Algorithm: Nested Hierarchical Clustering

1. Initialize $\sigma_i = 0$ for $i = 0$.
2. At $\sigma_i = 0$ each datum is a cluster.
3. Set $i = 1$.
4. Cluster data, using equation 83, at σ_i by finding the new cluster centres using the old cluster centres at σ_{i-1}. Take the union of the data points whose cluster centres at σ_{i-1} arrive at the same location.
5. If the number of clusters is greater than one let $i := i + 1$ and go to step 4.
6. Stop if the number of clusters is one.

Algorithm: Non-Nested Hierarchical Clustering

1. Initialize $\sigma_i = 0$ for $i = 0$.
2. At $\sigma_i = 0$ each datum is a cluster.
3. Set $i = 1$.
4. Cluster data, using equation 83, at σ_i by finding the new cluster centres using the old cluster centres at σ_{i-1}. If two cluster centres arrive at the same point then a new cluster is formed and the old clusters disappear.
5. If the number of clusters is greater than one let $i := i + 1$ and go to step 4.
6. Stop if the number of clusters is one.

To reduce the computational costs only the cluster centres are traced along the maximal curves. The solution to equation 83 for σ_i is therefore found by substituting $x[0] = \nu_{i-1}$, where ν_{i-1} is the cluster centre obtained at σ_{i-1}.

5.2.4 Cluster number selection

An important aspect of the algorithm is to determine the correct number of clusters. The suggested and commonly used method involves using the σ-lifetime for a cluster as a validity criterion [47]. The definition of the cluster σ-lifetime, as stated by Leung et al, follows [47].

Definition: Cluster Lifetime
The σ-lifetime of a cluster is the range of σ values over which a cluster survives and does not merge i.e. it is the difference in σ values between the point of cluster formation and cluster merging.

It is noted in literature that the logarithmic difference for testing the lifetime of a cluster is preferred, as shown in detail by Leung et al [47]. The justification for using the logarithmic scale is that the number of clusters or zero-crossing points as a function of scale σ tends to be an exponential decay for data that is distributed uniformly [19] i.e. $\pi(\sigma) = \pi(0)e^{-\beta\sigma}$, where $\pi(\sigma)$ is the number of solutions to equation 81.

The parameter β is a function of the dimensionality and is usually unknown [47]. If the logarithmic scale $k = \pi(0)log\left(\frac{\sigma}{\epsilon}\right)$ is used, where ϵ is a small positive constant, then the plot of $\pi(\sigma)$ is approximately linear and free of the parameter β [47]. The reason for

changing the scale to logarithmic is that it is much simpler to discern if the plot is linear as apposed to an exponential with the unknown parameter β.

The major steps in selecting the valid cluster structure in a data set are summarised in the following five steps.

1. Observe the plot of $\pi(\sigma)$ and if it is constant over a wide range of σ values then structure exists in the data set, otherwise the data set is uniformly distributed.
2. If the data has an inherent structure then the lifetime can determine the correct number of clusters and the corresponding clustering.
3. Find a clustering that has the longest σ-lifetime together with the smallest ratio of cluster separation to within cluster scatter, as discussed in section 4.
4. The validity of the cluster can be determined by the lifetime and other defined validity indices.
5. If required the measure of outlierness can be used to detect any spurious points or outliers in the data set.

5.2.5 Classification

The assumption for classification is that a significant partition exists and can be found at some resolution σ^*. The problem is to assign each iterated data point $\mathbf{x}_i$ to each of the $\pi(\sigma^*)$ partitions [19]. The proposed method is based on the metric used by Roberts [19], which is stated for the k^{th} cluster as

$$P\left(\mathbf{x}_i|C_k\right) = \frac{1}{2}\left(1 + \left\langle \frac{\nabla_x y(\mathbf{x},\sigma^*)}{\|\nabla_x y(\mathbf{x},\sigma^*)\|}, \frac{\mathbf{d}_{i,k}}{\|\mathbf{d}_{i,k}\|} \right\rangle\right) \times e^{\frac{-\left\|\mathbf{d}_{i,k}\right\|_2^2}{4(\sigma^*)}}, \tag{84}$$

where

$$\mathbf{d}_{i,k} = \bar{\mathbf{x}}_k - \mathbf{x}_i.$$

The exponential distance is chosen since it has been found to perform superior to the commonly used Euclidean distance, as shown in the study performed by Yao [48]. The datum points can be assigned to a specific partition by finding which partition produces the maximum value for the metric defined in equation 84.

5.2.6 Algorithm illustration and properties

The clustering algorithm and its properties are presented by using a simple data set in two dimension with three well defined clusters for clarity purposes, as shown in figure 5.4.

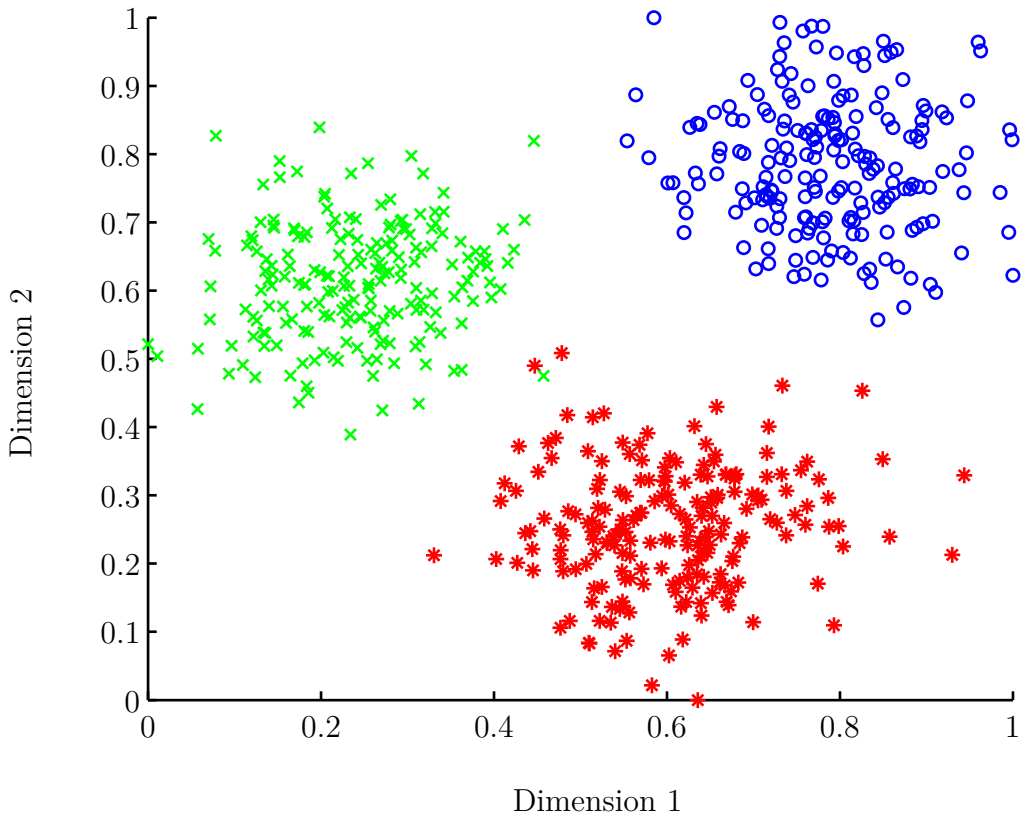


Figure 5.4: Artificial data set used to illustrate the properties of diffractive clustering.

As illustrated in figure 5.4 the data set has three separate clusters and has been normalised using the range as the scale measure with the minimum of the data set as the location measure. The following step determines the cluster number that has the longest lifetime and the corresponding σ value. The value for the constant $\pi(0)$ using the logarithmic scale is chosen to be the same as Leung et al i.e. a value of $\pi(0) = \frac{1}{log(1.05)}$, whereas the constant $\epsilon = 1 \times 10^{-4}$. A plot of the number of clusters as a function of logarithmic σ scale is shown in figure 5.5.

The plot in figure 5.5 shows that $\pi(\sigma)$ approximately decreases linearly for $k < 45$. At $k = 45$, known as the critical point, the filtering parameter σ is large enough to expose the inherent structure in the data set. In this case the longest lifetime, as shown in figure 5.5, occurs when the cluster number is three. The σ value is either chosen to be the minimum

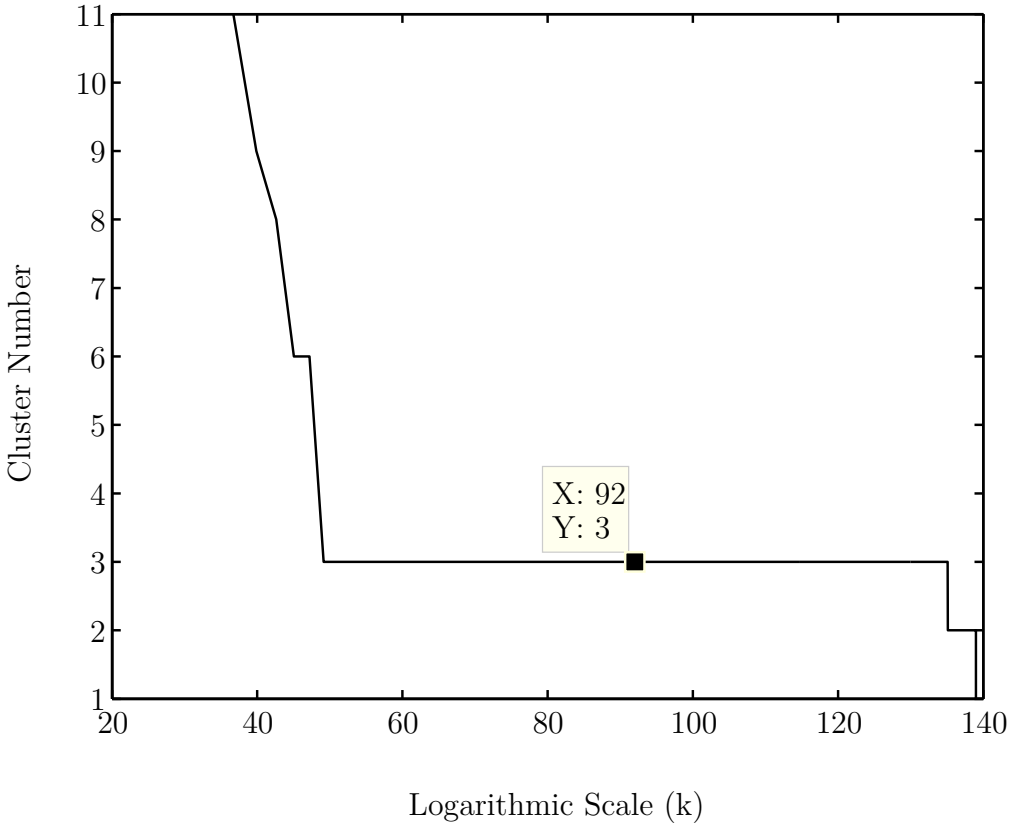


Figure 5.5: Logarithmic scale plot of the cluster number $\pi(\sigma)$.

point or the middle point of the longest lifetime, which in this case is 8.9×10^{-3}.

The aperture function for the data set at $\sigma = 8.9 \times 10^{-3}$ is shown in figure 5.6. The aperture function has three distinctive peaks which are indicative of the cluster centres and the spread of each peak covering the points associated with the cluster centres.

The diffraction-based algorithm tracks the centres of the clusters which significantly reduces the computational effort. The evolutionary tree of the cluster centres as the parameter σ increases is shown in figure 5.7. The evolutionary tree shows how the cluster centres merge into a single cluster as σ increases. The longest σ time is three, as shown in the evolutionary tree diagram.

The transformation of the aperture function as σ increases is shown in figure 5.8. Initially all the data points are separate clusters but as σ increases the points begin to merge to form an aperture function with a single peak.

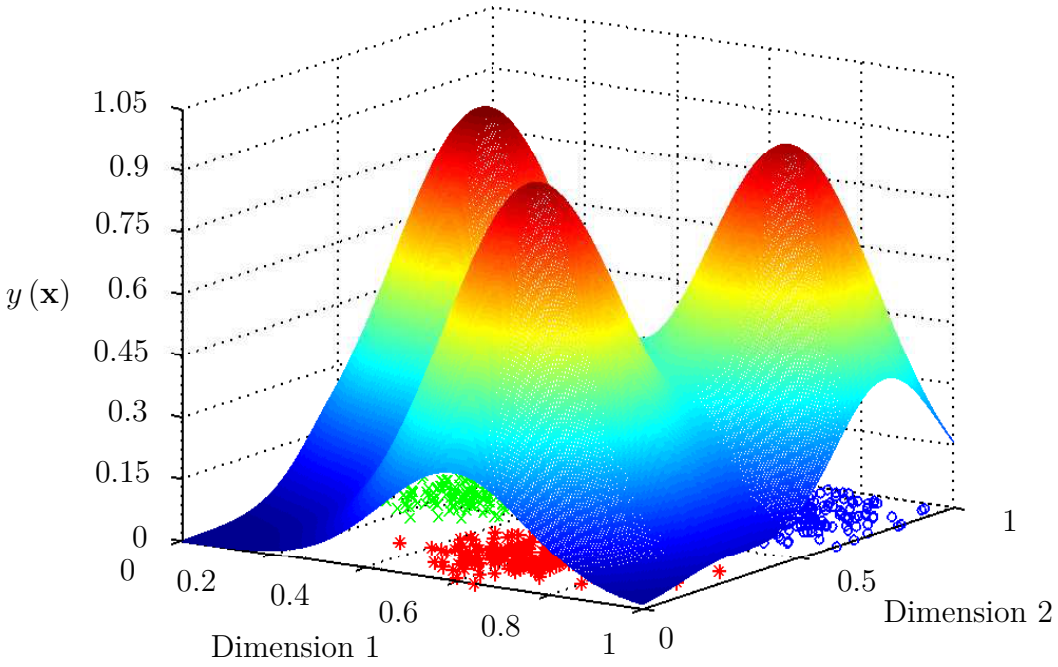


Figure 5.6: Aperture function for the artificial data set for $\sigma = 8.9 \times 10^{-3}$.

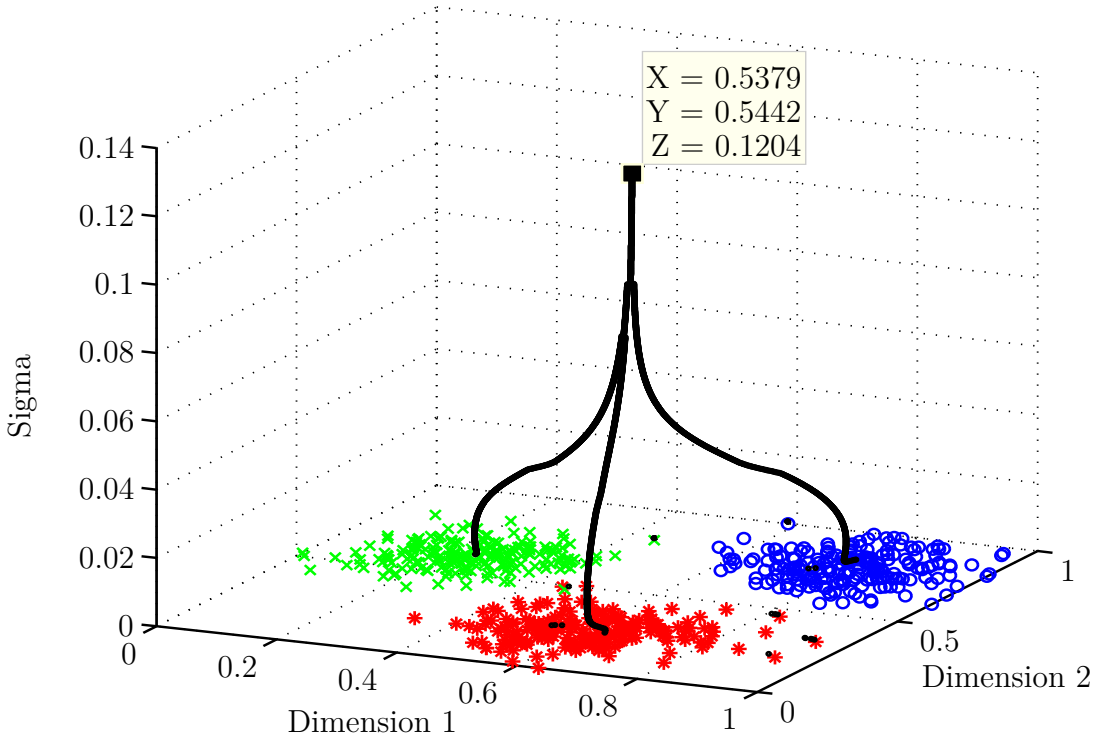


Figure 5.7: Evolutionary tree diagram illustrating the convergence of the cluster centres.

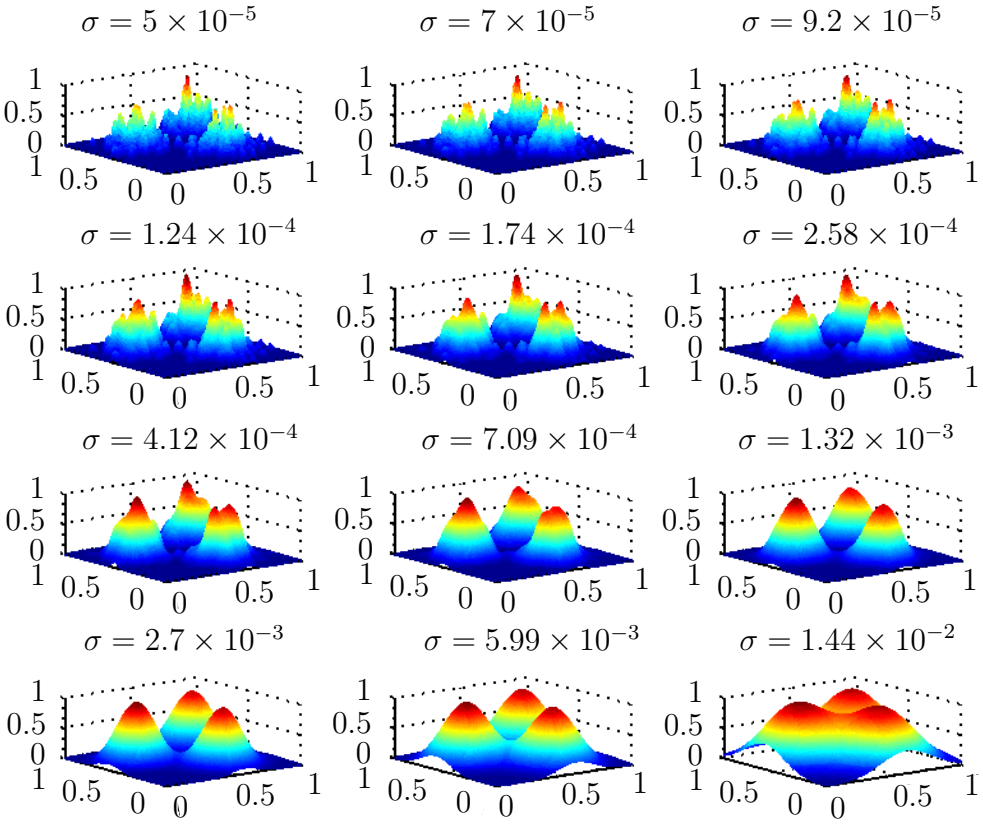


Figure 5.8: Sigma evolution of the aperture function.

5.3 Summary

The diffractive clustering algorithm together with its properties were derived from the basic principals of light and applied to an artificially generated data set to illustrate its operation. The diffractive clustering algorithm is able to resolve the data at different scales, determine and then cluster the inherent structure without any bias or *a priori* information. The algorithm is well suited for gene expression analysis since there is a large amount of information unknown about genes with their complex interactions. The proposed and developed clustering algorithm therefore offers a new exploratory tool for microarray data analysis.

The algorithm includes two paradigms namely nested and non-nested hierarchical clustering, with the latter preferred, since the data points are able to change cluster membership as the free parameter evolves. The algorithm is also modified such that only the cluster centres are traced, which significantly reduces the amount of iterations and computational time. The exponential distance metric used to classify the data points to their respective

clusters is also well suited for the gradient ascent method in the diffractive clustering algorithm.

The overall result is a clustering algorithm that utilises metrics and methodologies similar to those found in successful clustering algorithms. The main difference being that the diffractive clustering algorithm is unbiased in the sense that it does not require the number of clusters like the k-means algorithm. The diffraction principal of light also offers a novel derivation and view of the processes involved during clustering.

6 CLUSTERING RESULTS AND ANALYSIS

The developed diffractive clustering technique is tested on the following gene expression data sets pertaining to cancerous tissue samples. The results obtained are compared to commonly applied clustering techniques which include: hierarchical clustering, k-means, self-organising map and the fuzzy c-means algorithm. The methods for data normalisation and evaluation of the clustering algorithms are kept the same throughout the analysis. The main focus of the algorithm is that of class discovery i.e. determining the number of classes assuming no *a priori* knowledge of the data set. The two main issues associated with class discovery are:

1. Developing a clustering algorithm.
2. Testing that the putative results are meaningful and valid.

The focus is primarily on clustering the samples and comparing the results to a specified structure which is imposed on the data. The clustering algorithms are also evaluated using relative criteria to determine their performance assuming that no *a priori* knowledge of the data set exists.

The clustering algorithms were all implemented in Matlab 7.6.0 (R2008a) on a Intel Pentium 2.3 GHz, 4 GB RAM, computer. An additional toolbox was downloaded to implement the self-organising clustering algorithm called the SOM toolbox for Matlab [49].

6.1 Golub Data Set

The Golub (et al) data set is a well known and established data set for testing classifiers and class discovery algorithms. The data set is comprised of acute lymphoblastic leukaemia (ALL) samples and acute myeloid leukaemia (AML) samples. Patient samples are currently classified using techniques such as histochemistry, immunophenotyping and cytogenetic analysis [10]. The Golub data set was classified by classical observation of the nuclear morphology, enzyme-based histochemical analysis and antibody attachment to specific surface molecules pertaining to either lymphoid or myeloid cells.

The data set is divided into two types one for training and one for testing the classifier. The initial training set contains 38 samples of which 27 are ALL and 11 are AML samples. The independent testing set contains 34 samples of which 20 are ALL and 14 are AML samples. The RNA was prepared from bone marrow mononuclear cells with the samples hybridised to a high-density oligonucleotide Affymetrix array containing 6 817 probes.

The expression profile for each sample was then recorded and quantified using quality control standards [10].

The Golub data was first filtered using the call markers to find genes that were present more than 1% out of all the samples. The dimensionality of the data was then reduced using the ISOMAP algorithm to a suitable dimension. The residual variance of the data set is shown in figure 6.1. The residual variance plot shows that the correct dimension is two as this is where the curve begins to linearly decay.

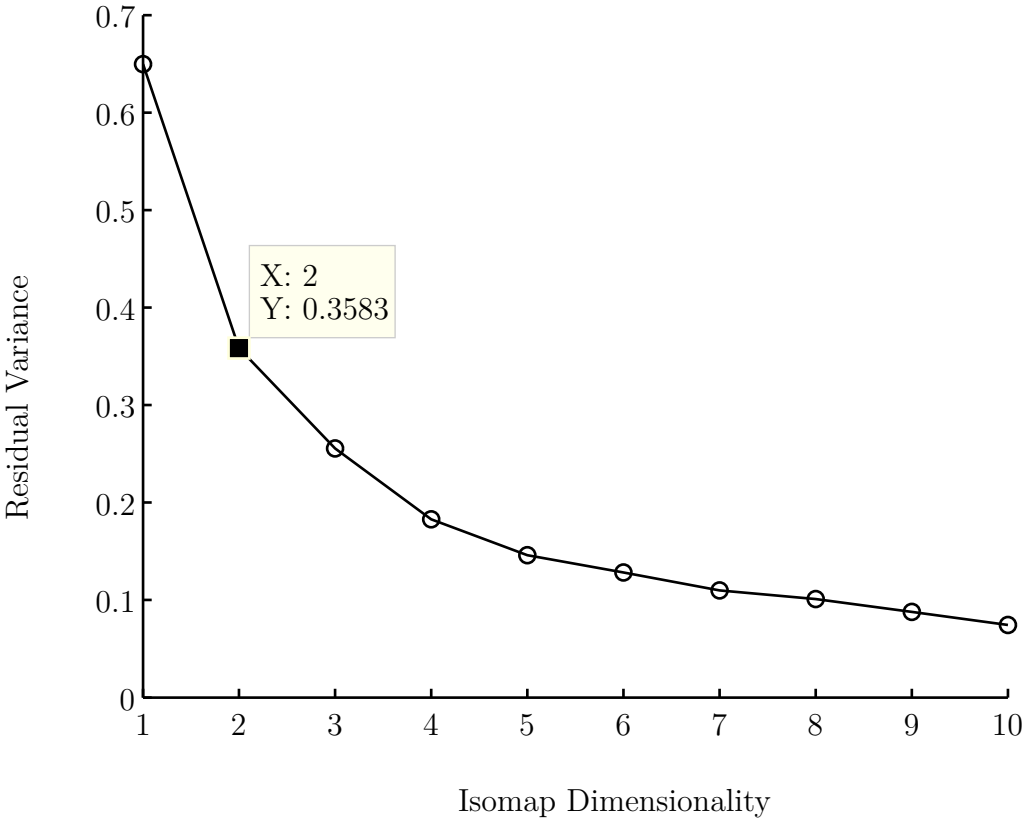


Figure 6.1: Residual variance of the ISOMAP algorithm for the Golub data set.

The data was normalised using the range as the scale measure and the minimum as the location measure. The *a priori* classification of the samples is shown in figure 6.2, with the red markers indicating the ALL subtypes and the green samples indicating the AML subtypes. The first two dimensions of the data set are labelled using X and Y, as shown in figure 6.2.

The diffractive clustering algorithm was applied to the filtered two dimensional Golub data set for optimal results. The lifetime curve for the clustering algorithm is shown in figure 6.3. The lifetime curve shows that the inherent number of clusters matches the expected amount. The value of σ was selected to be the minimum value of the range

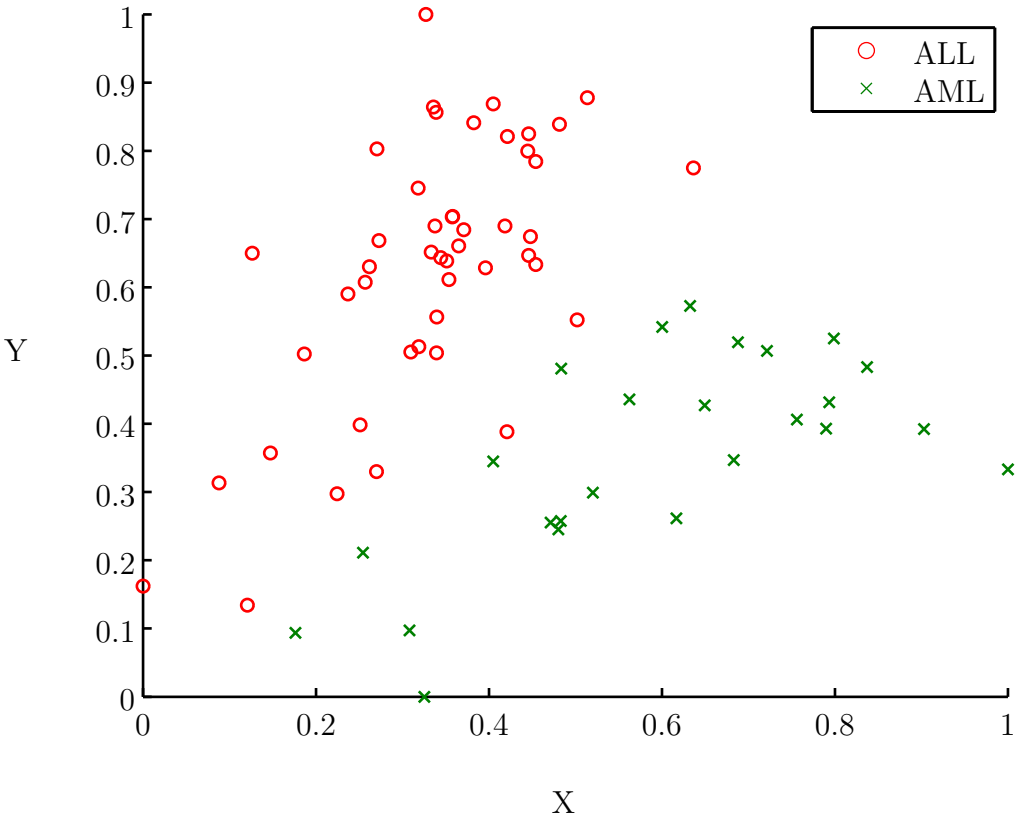


Figure 6.2: Scatter plot of the *a priori* classification for the Golub data set.

where the cluster number remains constant for the longest time, in this case the value of σ is 9.2×10^{-3}.

The performance of the clustering algorithms are determined using three main measures: average external criterion, average validity index and accuracy. The average external criterion is the average of the three main external criteria, covered in section 4.2, and defined as

$$\text{Average External Criterion} = \frac{J + R + F}{3}, \tag{85}$$

where

J = Jaccard Coefficient,

R = Rand Score,

F = Folkes and Mallows Index.

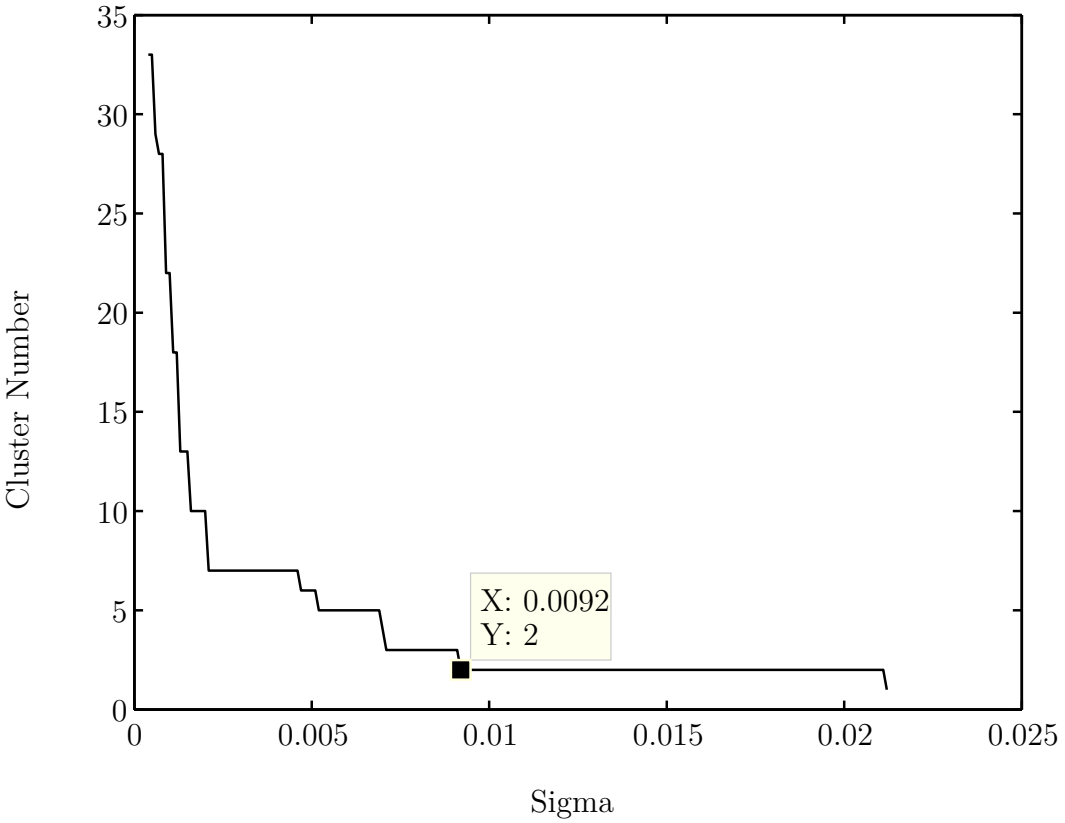


Figure 6.3: Cluster lifetime plot for the two dimensional Golub data set.

The average validity index is the average of the four main indices, as covered in section 4.4, and is given as

$$\text{Average Validity Index} = \frac{\frac{1}{I_B} + I_D + I_C + I}{4}, \tag{86}$$

where

$$\begin{aligned} I_B &= \text{Davies-Bouldin Index}, \\ I_D &= \text{Dunn's Index}, \\ I_C &= \text{Calinski Harabasz Index}, \\ I &= \text{I Index}. \end{aligned}$$

It is noted in equation 86 that the Davies-Bouldin index is inverted, since the index is minimised when there is a suitable clustering result. The average validity index should therefore be maximised for a good clustering result.

The accuracy of the clustering results are determined using a simple misclassification ratio as shown by

$$\text{Accuracy} = \frac{N_s - N_m}{N_s} \times 100\ \%, \tag{87}$$

where N_s is total number of samples and N_m is the total number of misclassified samples produced by the algorithm.

The diffractive clustering algorithm results were compared to the other main clustering schemes, as shown in table 6.1. The k-means algorithm was used with the number of expected clusters equated to two, similarly for the fuzzy c-means algorithm. The hierarchical clustering algorithm implemented used the standard Euclidean distance with single linkage as the merging measurement. The topology of the self-organising map was 2×1, such that two cluster centroids could be found [10].

Table 6.1: Comparison of the clustering results for the Golub data set.

Algorithm	**Average External Criterion**	**Average Validity Index**	**Accuracy (%)**
Diffractive clustering	87.5	73.5	94.4
k-means	76.1	62.6	88.9
Fuzzy c-means	76.1	57.3	88.9
Hierarchical clustering	59.3	63.7	63.9
Self-organising map	60.5	12.6	65.3

The results, as shown in table 6.1, demonstrate that the diffractive clustering algorithm outperforms the other algorithms in terms of accuracy and validity. An accuracy of 94.4% for diffractive clustering implies that only 4 samples were misclassified, whereas in fuzzy c-means and k-means 8 samples were misclassified which is double that of the diffractive clustering algorithm.

The SOM and hierarchical clustering algorithms both perform relatively poorly compared to the other algorithms. The reason being that perhaps the incorrect choice of neurons and topology for the SOM was used, or the cutoff level for the hierarchical dendrogram was not optimal. The main problem with these algorithms is the choice for the parameters and determining the cluster number *a priori*. The diffractive clustering algorithm bypasses these problems by plotting the lifetime for the cluster number, which gives the optimal parameter choice for clustering the selected data set.

6.2 MILEs Data Set

The Microarray Innovations in LEukaemia (MILE) study is a collection of 204 analyses from an international standardisation programme that was conducted in 11 laboratories [13]. The study analyses 16 subtypes of leukemia, myelodysplastic syndrome (MDS) and normal bone marrow tissue in over 4 000 patients. The samples were classified *a priori* using gold standard techniques such as morphology, cytogenetics, immunophenotyping etc [13].

The main subtypes of acute lymphoblastic leukaemia (ALL) that are analysed follow those from the study by Li et al and include: t(4;11)MLL-rearrangement, t(9;22)BCR-ABL, T-ALL, t(12;21)TEL-AML1, t(1;19)E2A-PBX1 and Hyperdiploid > 50 [12]. The number of samples were evenly distributed as much as possible resulting in 276 samples with a total of 54 675 genes. The number of samples was also limited by the memory capacity in Matlab. The lymphoblastic leukaemias result from the failed differentiation of the haematopoietic cells, specifically the lymphoid stem cells [13]. The occurrence and risk associated with each subtype is shown in table 6.2.

Table 6.2: The six major subtypes of ALL, obtained from [12].

Subtype	**Occurrence (%)**	**Clinical Nature**
t(4;11)MLL	5-8	Infant ALL high risk
t(9;22)BCR-ABL	2-3	High risk
T-ALL	10-13	Moderate risk
t(12;21)TEL-AML1	16-22	Low risk
t(1;19)E2A-PBX1	5	Low risk
Hyperdiploid > 50	25-35	Low risk

The dimensionality of the MILEs dataset was reduced to three using the ISOMAP algorithm and the information provided by the residual variance curve, as shown in figure 6.4. The figure shows that the inherent dimensionality is three as the curve begins to decay linearly at that point.

The data was background corrected and normalised using the robust multiarrary average (RMA) technique. The data was then normalised again using the range method such that the gene expression values range was $[0, 1]$. The *a priori* classification of the data set is shown in the figure 6.5, with each of the six subtypes designated their own specific coloured marker.

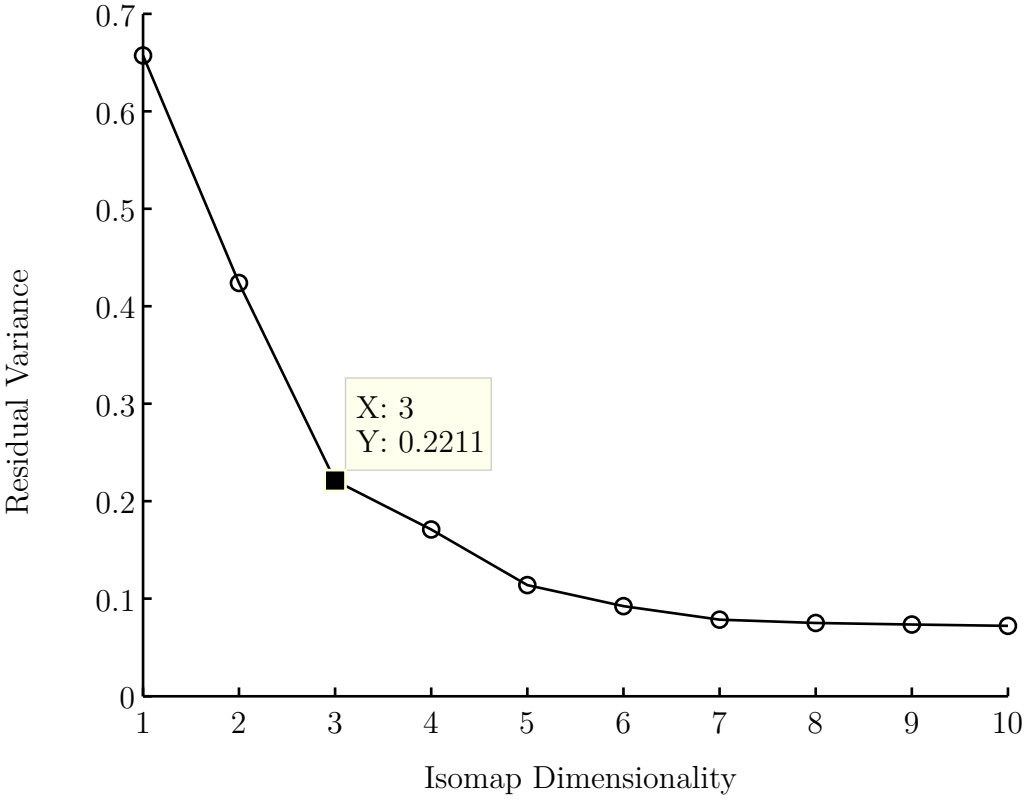


Figure 6.4: Residual variance of the ISOMAP algorithm for the MILEs data set.

The lifetime curve for the Miles data set is shown in figure 6.6. The curve shows that the cluster number stays constant at six for a large range of σ. The graph is not complete since the cluster number stays fixed at six for more than one order of magnitude in σ, and is therefore assumed to be the correct cluster number. The chosen value of σ is the minimum value at which the cluster number stays constant, which in this case is 17.3×10^{-3}.

The results of the diffractive clustering algorithm were compared to the other clustering algorithms, as shown in table 6.3. The number of expected clusters in the k-means algorithm was equated to six, similarly for the fuzzy c-means algorithm. The hierarchical clustering algorithm used the standard Euclidean distance with single linkage as the merging measurement and a maximum cluster level of six. The topology of the self-organising map was set to 6×1 such that six cluster centroids could be found.

The results show that the diffractive clustering algorithm outperforms the other algorithms in terms of validity and accuracy. The fuzzy c-means algorithm is the closest to the diffractive clustering algorithm with an accuracy of 71.4%. The fuzzy c-means algorithm is similar to the k-means algorithm in the sense that there is random initialisation

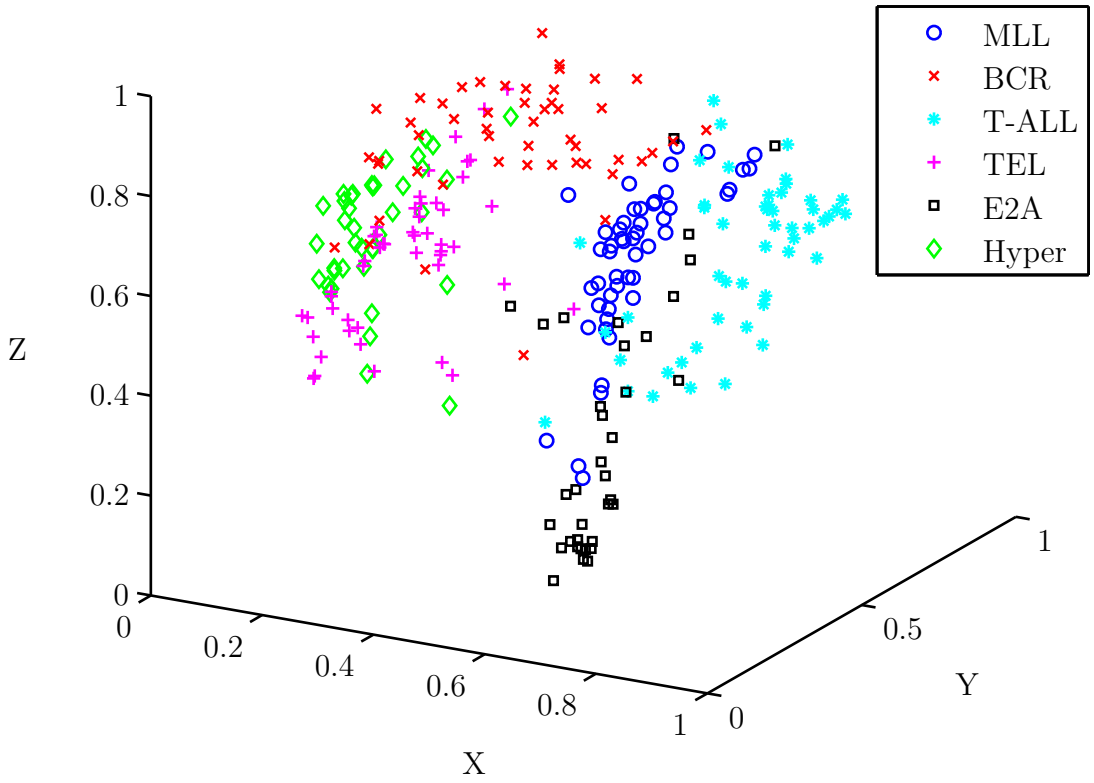


Figure 6.5: Scatter plot of the *a priori* classification for the MILEs data set.

and minimisation of intra-cluster variance [50]. The problem is that the minimum found by these algorithms is generally not the global minimum, which seems to be the case in this study.

Table 6.3: Comparison of the clustering results for the MILEs data set.

Algorithm	**Average External Criterion**	**Average Validity Index**	**Accuracy (%)**
Diffractive clustering	66.6	179.0	73.1
k-means	59.1	152.6	61.6
Fuzzy c-means	62.9	171.4	71.4
Hierarchical clustering	47.7	64.2	47.1
Self-organising map	46.1	10.6	47.1

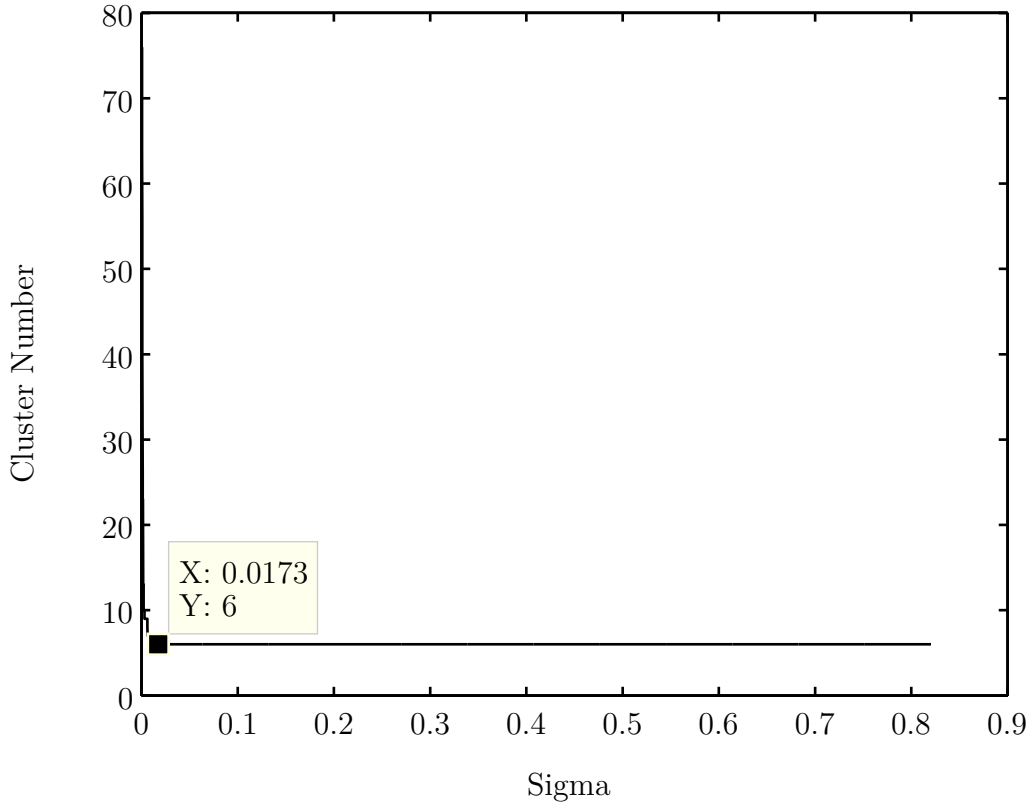


Figure 6.6: Cluster lifetime plot for the three dimensional MILEs data set.

The fuzzy c-means and k-means algorithm also both require the *a priori* number of clusters, which defeats the purpose of unsupervised clustering. The resulting accuracy is low when compared to the Golub data set, as 74 out of the 276 samples are misclassified. The reason for this low accuracy is attributable to the large number of different subtypes and the high-dimensionality in which the samples are situated [7].

6.3 Khan Data Set

The Khan data set was obtained from the study performed on classifying small, round blue-cell tumors (SRBCT) of childhood [14]. The tumors belong to four distinct diagnostic categories which present challenges for clinical diagnostics [14]. The four classes are neuroblastoma (NB), rhabdomyosarcoma (RMS), Burkitt's lymphoma (BL) and the Ewing family of tumors (EWS). The correct diagnosis of which class the tumor belongs is important since treatment options, responses to therapy and prognoses vary significantly depending on the diagnosis [14].

The gene-expression data was obtained from cDNA microarrays that each contained 6 567 genes, and a sample size of 83. The data was normalised to a range of $[0, 1]$ using the minimum as the location measure and the range as the scale measure. The dimensionality for the data set was reduced using the ISOMAP algorithm with the resulting residual variance curve shown in figure 6.7. The dimensionality for the rest of the analysis was chosen to be three, since the residual variance decays linearly at this point. The predetermined classes of SRBCT tumors are shown in figure 6.8.

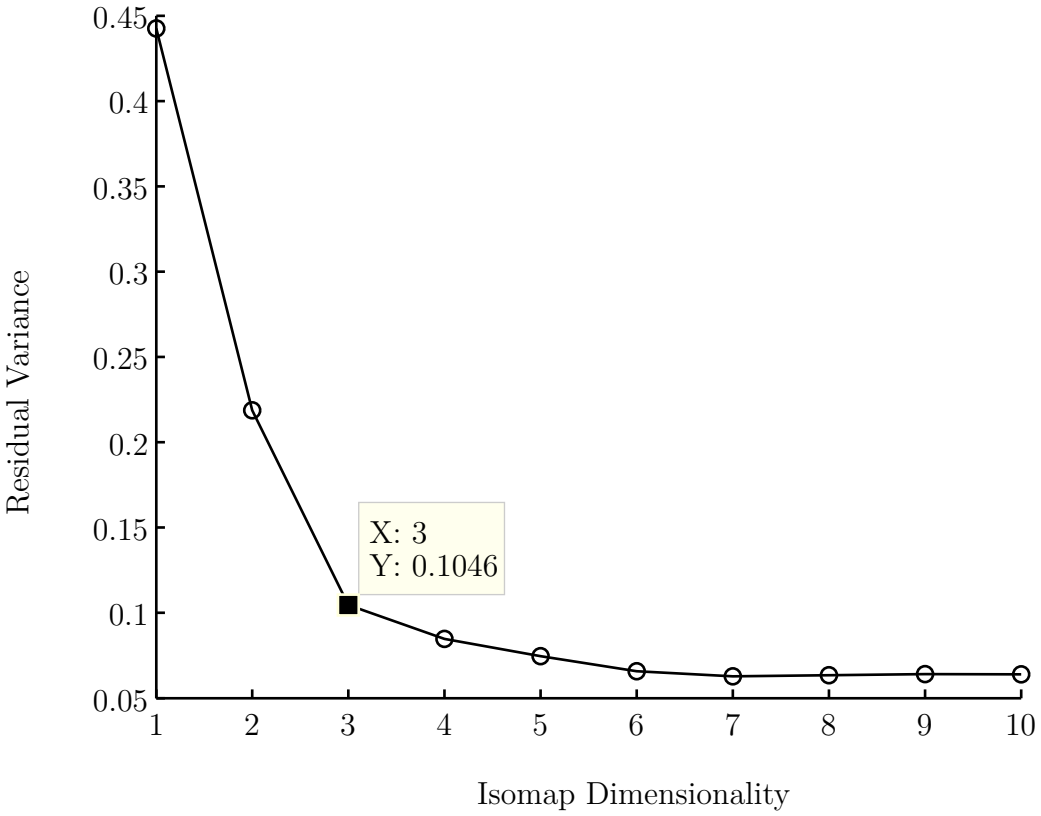


Figure 6.7: Residual variance of the ISOMAP algorithm for the Khan data set.

The correct number of clusters for the unsupervised diffractive clustering algorithm was determined using the lifetime plot in figure 6.9. The plot shows that the longest lifetime corresponds to the correct number of classes which is four. The value of σ used in the diffractive clustering algorithm is determined from the minimum of the longest lifetime plot which is 9.9×10^{-3}.

The diffractive clustering (DC) algorithm was compared to the clustering algorithms for the Khan data set, as shown in table 6.4. The results show that the diffractive clustering algorithm is able to accurately separate the data into four distinct classes. The average validity index of the diffractive clustering algorithm is also relatively high indicating a

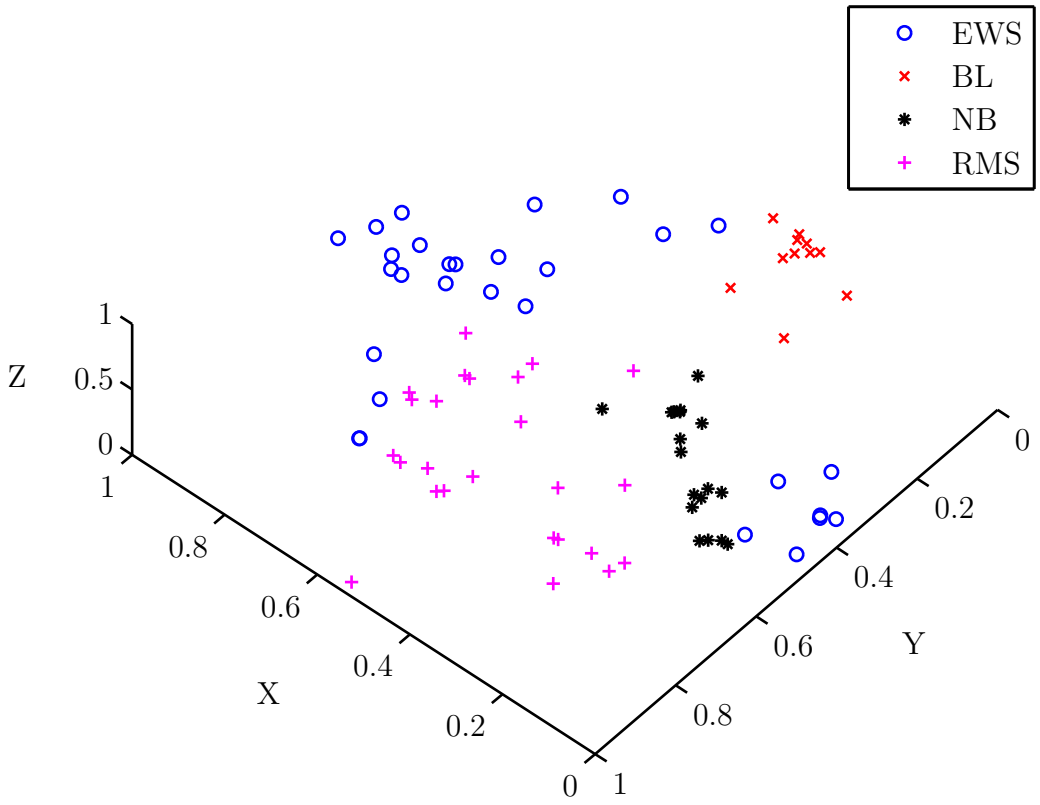


Figure 6.8: Scatter plot of the *a priori* classification for the Khan data set.

suitable clustering solution.

The DC algorithm was able to correctly classify 58 out of the 83 samples as apposed to the fuzzy c-means algorithm which only classified 54 out of the 83 samples correctly. A reason being that the fuzzy c-means algorithm possibly found a local minimum for its cost function as apposed to a global minimum.

Table 6.4: Comparison of the clustering results for the Khan data set.

Algorithm	**Average External Criterion**	**Average Validity Index**	**Accuracy (%)**
Diffractive clustering	53.3	98.4	70.0
k-means	49.1	105.7	63.0
Fuzzy c-means	50.9	113.7	65.1
Hierarchical clustering	40.7	79.8	43.2
Self-organising map	46.3	15.7	54.2

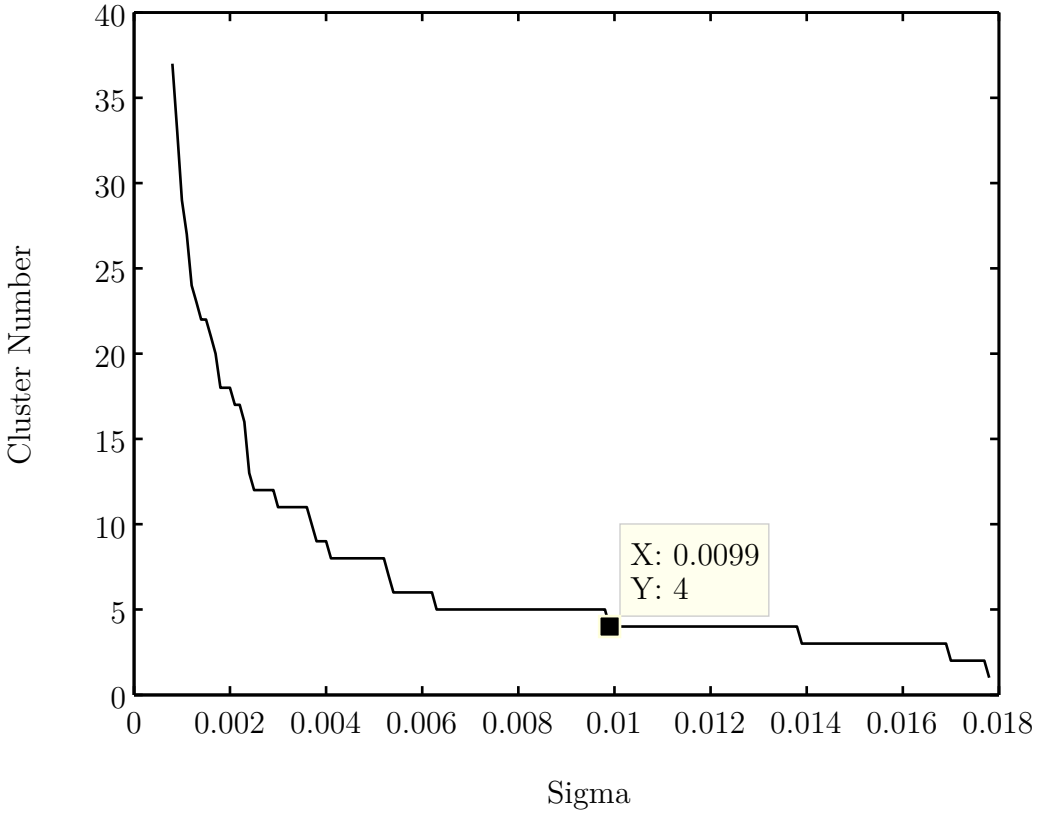


Figure 6.9: Cluster lifetime plot for the three dimensional Khan data set.

6.4 Shipp Data Set

The Shipp data set is a study performed on diffuse large B-cell lymphoma (DLBCL), which is the most common malignancy in adults and is curable in less than 50% [15]. The experiment performed by Shipp et al indentified tumours in a single B-cell lineage, specifically the distinction of DLBCL from a related germinal centred B-cell lymphoma, follicular lymphoma (FL) [15].

The clinical distinction of the two types of lymphomas is usually difficult as FLs acquire the morphologic and clinical characteristics of DLBCLs over time [15]. The micorarray transcription study, containing 6 817 genes, of the lymphomas was performed on 77 patients, of which 58 were diganosed with DLBCL and other 19 with FL.

The dimensionality of the Shipp data set was reduced to two dimensions as suggested by the linear decay of the resulting residual variance curve, which is produced by the ISOMAP algorithm and shown in figure 6.10. The *a priori* classification of the 77 samples in two dimensions is shown in figure 6.11. The similarity of the tumour lineage between DLBCLs and FLs is evident by the amount of mixing of the different data points, as

shown in figure 6.11.

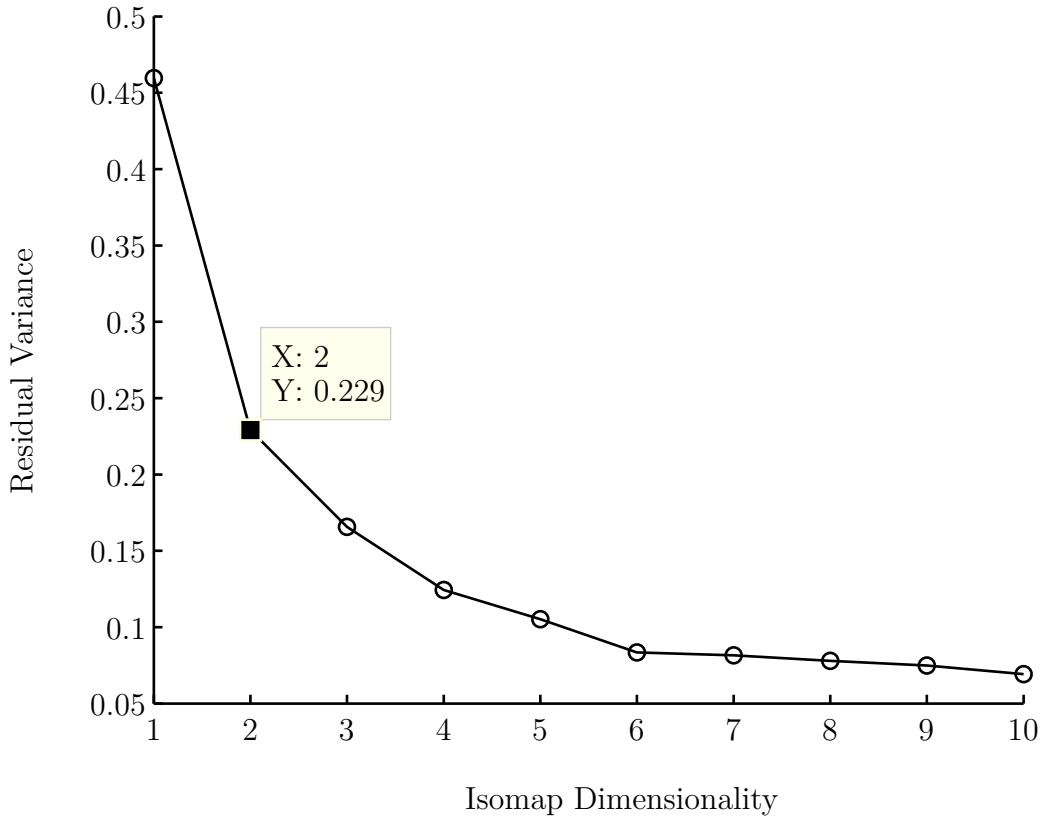


Figure 6.10: Residual variance of the ISOMAP algorithm for the Shipp data set.

The cluster number used for the diffractive clustering algorithm was determined using the lifetime plot as shown in figure 6.12. The lifetime plot suggests that the correct number of clusters is two, which corresponds to the correct *a priori* number of clusters in the data set. The value for σ in the DC algorithm is 7.3×10^{-3}, which is the minimum of the longest lifetime range in figure 6.12.

The diffractive clustering algorithm, applied to the Shipp data set, was compared to the other main types of clustering algorithms. The clustering results are shown in table 6.5, with the most accurate and valid results pertaining to the novel diffractive clustering algorithm.

The diffractive clustering algorithm correctly classifies 49 out of the 77 samples, as apposed to the SOM and hierarchical clustering algorithm which correctly classify only 41 out of the 77 samples. The accuracy however of the diffractive algorithm, although 10% larger than the rest, is still relavitily low.

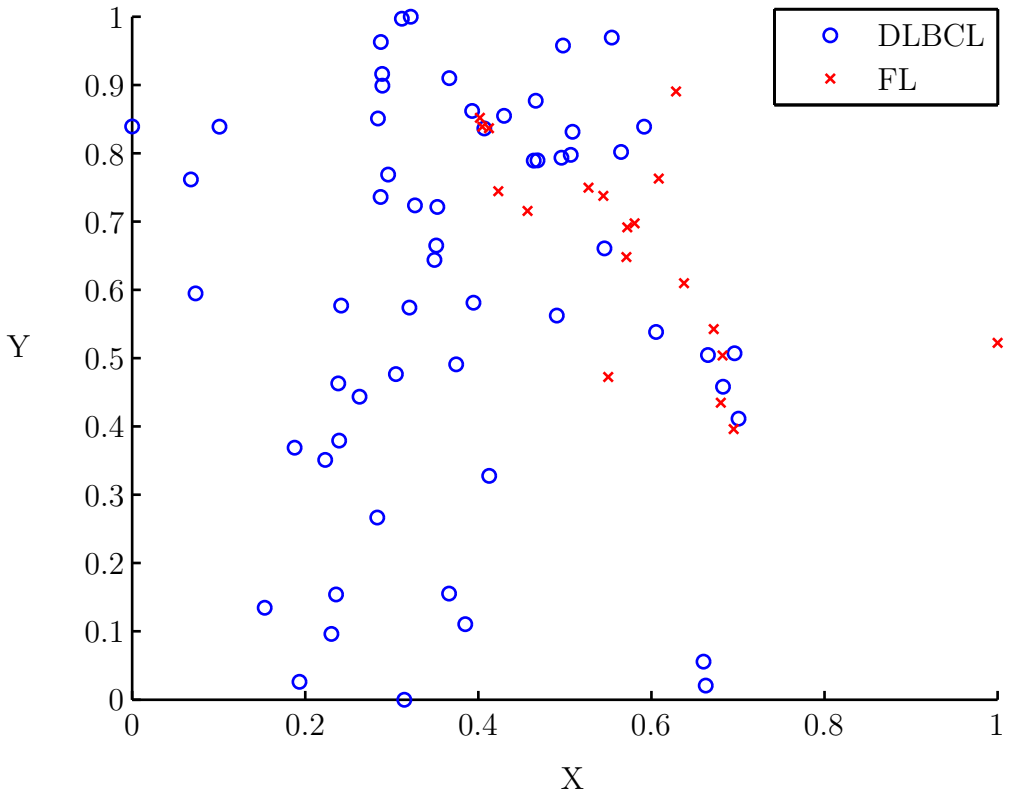


Figure 6.11: Scatter plot of the *a priori* classification for the Shipp data set.

The low accuracy is a result of the lymphomas being inherently indistinguishable due to their similar B-cell lineage. The genes could be filtered using supervised techniques, such as the t-statistic, to obtain better results, however the price of requiring prior classification would be incurred.

Table 6.5: Comparison of the clustering results for the Shipp data set.

Algorithm	**Average External Criterion**	**Average Validity Index**	**Accuracy (%)**
Diffractive clustering	55.5	160.6	63.6
k-means	48.0	79.7	51.9
Fuzzy c-means	48.7	82.0	51.9
Hierarchical clustering	51.1	112.2	53.3
Self-organising map	47.7	14.7	53.3

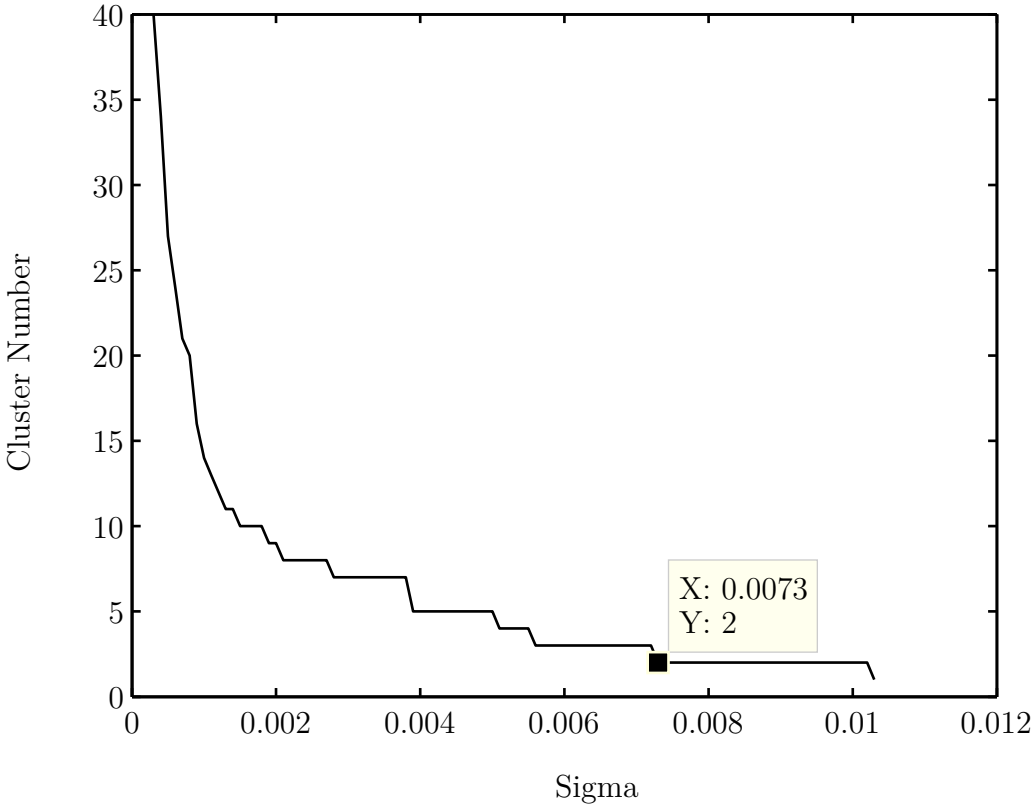


Figure 6.12: Cluster lifetime plot for the two dimensional Shipp data set.

6.5 Pomeroy Data Set

The Pomeroy data set is a study performed on embryonal tumours of the central nervous system (CNS) [16]. Medulloblastomas, a highly malignant brain tumour that originates in the cerebellum or posterior fossa, are most common in pediatrics with very little known about their response to treatment and pathogenesis [16]. The study performed by Pomeroy et al analysed the transcription levels of 99 patients to identify any expression differences between medulloblastomas (MED), primitive neuroectodermal tumours (PNETs), atypical teratoid/rhabdoid tumours (AT/RTs), malignant gliomas (MAL) and normal tissue [16].

The Pomeroy study analyses the DNA from 99 patients on oligonucleotide microarrays with 6 817 genes. The data was also split into three data sets with varying amounts of samples. The data set known as A2 is used in this clustering analysis and contains 90 samples, of which 60 are MED, 10 are MAL, 10 are AT/RTs, 6 are PNETs and 4 are normal [16].

The dimensionality of the data set was reduced prior to cluster analysis using the ISOMAP algorithm. The residual variance curve, as shown in figure 6.13, produced by the algorithm illustrates that the correct dimension is two. The *a priori* classification of the samples using clinical methods is shown in figure 6.14

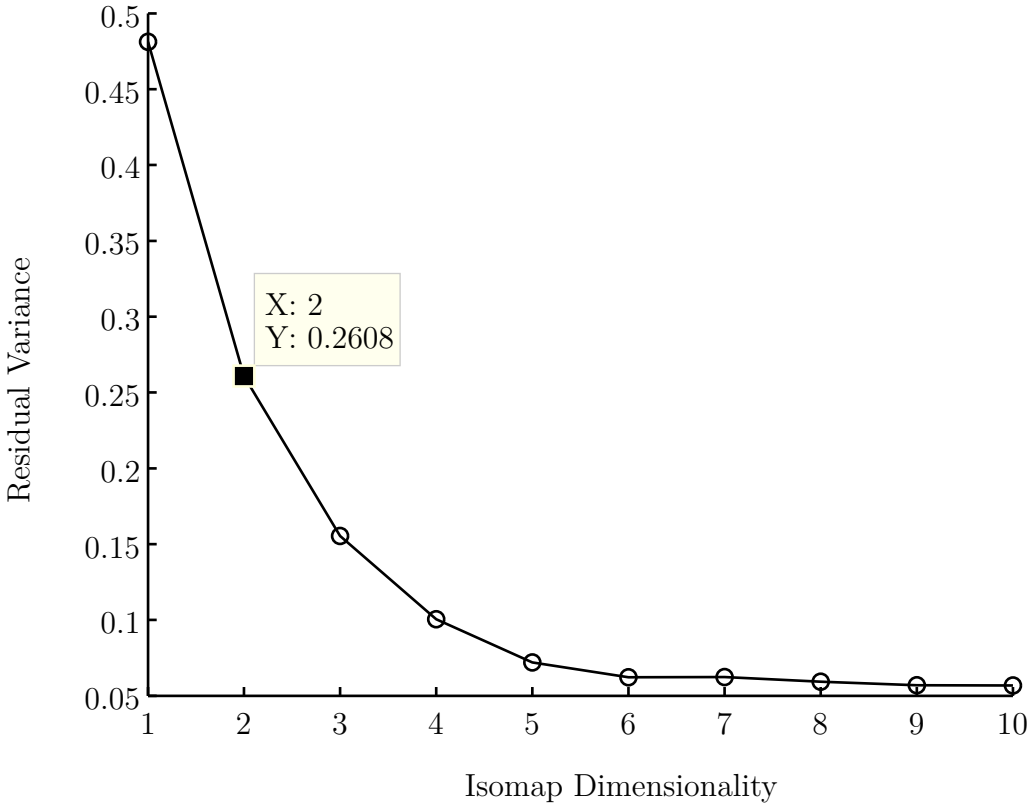


Figure 6.13: Residual variance of the ISOMAP algorithm for the Pomeroy data set.

The cluster number was determined using the cluster lifetime plot produced by the diffractive clustering algorithm, which is shown in figure 6.15. The longest lifetime once again corresponds to the correct cluster number which is five. A value for $\sigma = 2.81 \times 10^{-3}$ is used in the DC algorithm as it is the minimum of the longest lifetime range.

A cluster analysis was performed on the Pomeroy data set using the diffractive clustering algorthm and compared to the other main types of clustering algorithm. The results are shown in table 6.6 with the most accurate and valid results pertaining to the diffractive clustering algorithm. The DC algorithm correctly classifies 61 out of the 90 samples as apposed to the hierarchical clustering algorithm which only classifies 51 out of the 90 samples correctly.

The validity of the clustering solution produced by the DC algorithm is also remarkably high compared to the other algorithms. The main reason for the accuracy being low,

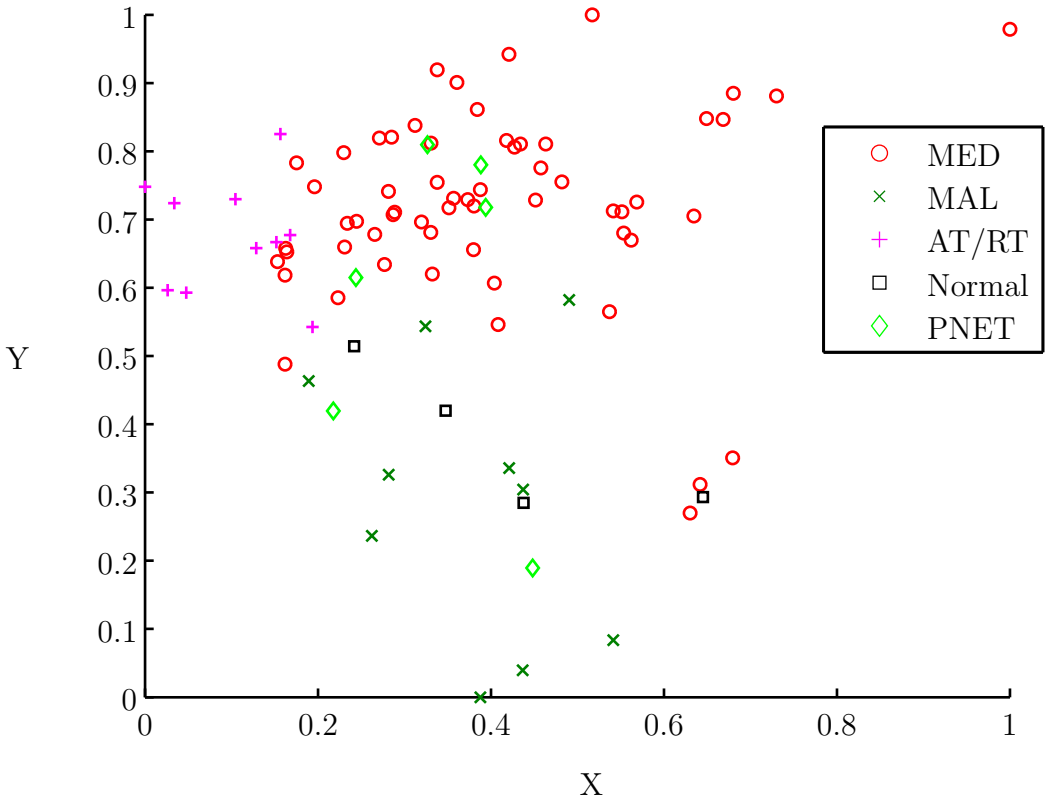


Figure 6.14: Scatter plot of the *a priori* classification for the Pomeroy data set.

although high relative to the other clustering results, is the unbalanced distribution of samples, which is a result of the large number of medulloblastoma samples in the data set. The clustering solution can also be improved if feature genes are selected prior to analysis, however this will require the *a priori* classification of the samples which defeats the definition of unsupervised learning.

Table 6.6: Comparison of the clustering results for the Pomeroy data set.

Algorithm	**Average External Criterion**	**Average Validity Index**	**Accuracy (%)**
Diffractive clustering	63.1	258.7	67.8
k-means	42.7	173.6	48.9
Fuzzy c-means	39.9	154.6	43.3
Hierarchical clustering	49.8	256.2	56.7
Self-organising map	41.2	13.7	44.4

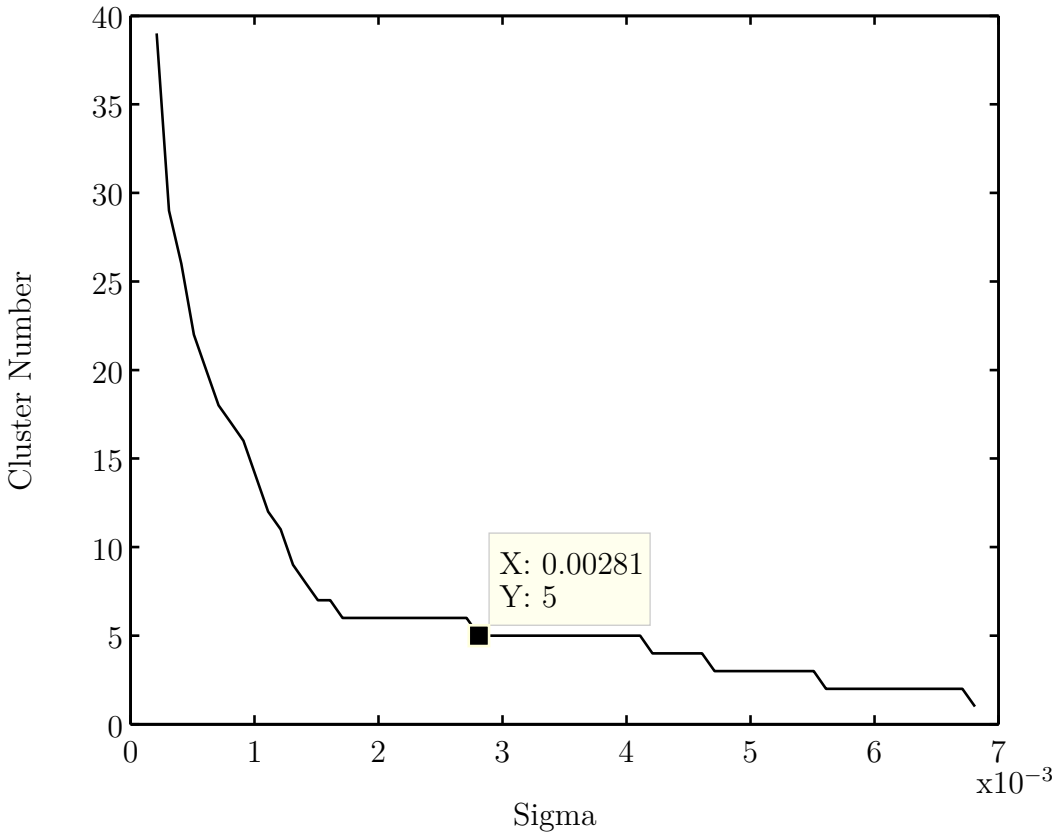


Figure 6.15: Cluster lifetime plot for the two dimensional Pomeroy data set.

6.6 Summary

The results show that overall the diffractive clustering (DC) algorithm outperforms the other clustering algorithms: k-means, fuzzy c-means (FC), hierarchical clustering (HC) and the self-organising map (SOM). The results were averaged across all the data sets for each algorithm and for each performance measure, as shown in figure 6.16. The overall results in figure 6.16 show that the diffractive clustering algorithm is higher in terms of all three criteria.

The average performance across all data sets for the diffractive clustering algorithm was 65.2% for the average external criterion, 154 for the average validity index and 73.8% for the accuracy. The numbers are relatively good when compared to the second highest scoring algorithm the fuzzy c-means with 55.7% for the average external criterion, 115.8 for the average validity index and 64.1% for the accuracy. The diffractive clustering algorithm is therefore 10% higher in terms of accuracy and more than 30% higher in terms of validity than the fuzzy c-means algorithm.

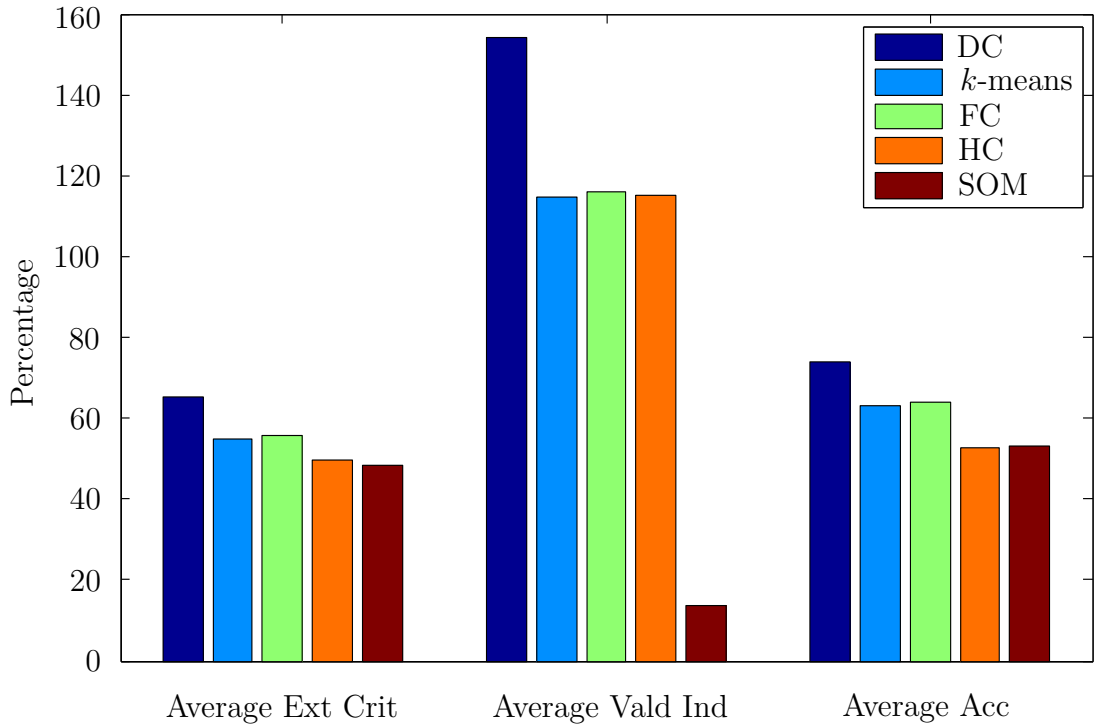


Figure 6.16: Overall performance results of the clustering algorithms.

The success of the diffractive clustering algorithm is due to the ability of algorithm to deal with non-spherical or arbitrarily shaped clusters, and in this sense is similar to the popular density-based clustering algorithms. The k-means and fuzzy c-means algorithm also both require the predetermined number of clusters which in itself can sometimes produce erroneous results.

The k-means and fuzzy c-means algorithms also minimise intra-cluster variance and as a result the global minimum is not always discovered. The other problem associated with k-means and fuzzy c-means is that the initial conditions are randomly selected which can result in different solutions each time the algorithm is run.

The overall performance of the diffractive clustering algorithm is noticeably better than the other main clustering algorithms such as the SOM and hierarchical clustering. The diffractive clustering algorithm although better still has computational issues in terms of time due to the complexity of the algorithm. The computational time however can be reduced significantly using lower-level programming languages such as c++.

7 CONCLUSION AND RECOMMENDATIONS

A summary of the research together with the work performed on developing and testing the diffractive clustering algorithm is given. The future development and applications of the algorithm is also provided with insight into some fundamental problems that should be investigated.

7.1 Discussion

The recent development in microarray technology has given rise to a large amount of data on the genetic expressions of cells. The parallel processing of this technology however has a disadvantage in that there is too much information to analyse. The statistical tools for dealing with the large amounts of data have proven useful, however the low number of samples still remains a serious problem. The feature selection techniques combined with supervised learning frameworks although successful continue to have serious bias and dependency on acquired knowledge of the genome and its functions. The supervised techniques fail when there is no *a priori* information about the samples or genes under investigation.

A solution to this problem of discovery and analysis in gene expression data is the application of unsupervised techniques such as cluster analysis. The clustering of samples allows one to find the inherent structure in the genome without filtering or representing the data with only a select few genes. The clustering of genes however remains an issue since the amount of probes on a microarray is typically above 6 000, which reduces the signal-to-noise ratio significantly.

The validation and performance of the clustering results, although poorly defined, can be addressed successfully with relative and external criteria. The number of clustering algorithms is large and as such the choice of the correct or preferred algorithm remains ambiguous. The languid approach is usually to choose the fastest algorithm such as the k-means algorithm. The classical algorithms although fast lack the insight to the clustering process, and rely on the predetermined, usually biased number of clusters to work.

The solution to this problem of determining the correct number of clusters is to use a hierarchical scheme in which there exist multiple solutions. The problem then is to determine the correct solution, which fortunately the developed diffractive clustering algorithm manages to achieve. The cluster number that survives the longest is chosen

indicating that the selected partitioning is indeed the inherent structure in the data set. The idea or hypothesis that the cluster number which survives the longest is the correct cluster number was tested using multiple cancerous tissue data sets with the results for each indicating the validity of the hypothesis.

The diffractive clustering algorithm is therefore independent of the number of clusters as the algorithm searches the space and requires no other form of feedback. The results in terms of accuracy and validity also outperformed the other classical algorithms with 10% difference in accuracy and external criteria, as well as more than 30% difference in terms of validity. The algorithm is therefore well suited for the cluster analysis of gene expression data.

The success of the diffractive clustering algorithm is due to the ability of the algorithm to cluster arbitrarily shaped clusters, and resolve the data at multiple scales. The only drawback is the computational time which could easily be solved using a more basic programming language such as C++. It was also found during the analysis of gene expression data that it is exceptionally hard not to include bias such as a favoured dimension or preferred set of feature genes which can lead to outstanding results. The diffractive clustering algorithm therefore utilised the ISOMAP algorithm which bypassed the need for arbitrarily selecting the number of genes or the dimension of the clustering space.

7.2 Future Work

The diffractive clustering algorithm, including the other clustering algorithms, often perform poorly in a high-dimensional space. The reason being is expressed by the relative contrast between data points, $\mathbf{x} \in \mathbb{R}^d$, in the following equation

$$\lim_{d\to\infty} \frac{\max(\|\mathbf{x}\|_p) - \min(\|\mathbf{x}\|_p)}{\min(\|\mathbf{x}\|_p)} \to 0, \tag{88}$$

where $\|\mathbf{x}\|_p$ is the L^p norm [7]. The stated equation shows that the relative contrast between data points is degraded and has no meaning in a high-dimensional space. A solution to this problem is to use a norm that performs better in a higher dimension, such as the fractional norm [7].

The problem of the norm was further investigated using exponential metrics and the fractional norm with surprising results. The results showed significant improvement in the relative contrast and suggested that an improvement could be made to the current diffraction-based clustering algorithm.

The diffraction clustering algorithm also uses the Euclidean norm to measure the similarity between cluster centres and iteration points. An improvement can therefore be made by replacing the Euclidean norm with the exponential and fractional norm. The results from Aggarwal suggest that this improvement is possible, since it was found that the k-means algorithm significantly improved when the fractional norm was implemented in place of the Euclidean norm [7].

The application of the diffractive clustering algorithm to the clustering of genes instead of samples remains unknown and an interesting question. The application to other scientific fields is also untested and should be investigated. The diffractive clustering algorithm is in theory applicable to the other aforementioned fields since the framework of the algorithm is general and consistent.

The other improvement which could be made to the diffractive clustering algorithm is the searching algorithm used for locating the maxima of the aperture functions. The current gradient ascent method using the Euler approximation ,although accurate, is not the fastest. Quasi-Newton gradient search algorithms therefore could possibly improve the computational speed of the algorithm.

References

[1] K. Somasundaram, S. K. Mungamuri, and N. Wajapeyee. "DNA Microarray Technology and its Applications in Cancer Biology." Review, Department of Microbiology and Cell Biology, Indian Institute of Science, 2002.

[2] M. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. "Cluster Analysis and Display of Genome-Wide Expression Patterns." *The National Academy of Sciences*, vol. 95, pp. 14863–14868, 1998.

[3] E. Domany. "Cluster Analysis of Gene Expression Data." *Journal of Statistical Physics*, vol. 110, no. 3–6, pp. 1117–1139, 2003.

[4] S. Drăghici. *Data Analysis Tools for DNA Microarrays*. London: Chapman and Hall/CRC, first ed., 2003.

[5] N. Belacel, Q. Wang, and M. Cuperlovic-Culf. "Clustering Methods for Microarray Gene Expression Data." *Journal of Integrative Biology*, vol. 10, no. 4, pp. 507–531, 2006.

[6] G. Gan, C. Ma, and J. Wu. *Data Clustering Theory, Algorithms, and Applications*, chap. 4, pp. 43–52. SIAM, Philadelphia, ASA, Alexandria: ASA-SIAM Series on Statistics and Applied Probability, 2007.

[7] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. "On the Surprising Behaviour of Distance Metrics in High Dimensional Space." Lecture notes, Institute of Computer Science, University of Halle, 2001.

[8] K. Y. Yeung and W. L. Ruzzo. "Principal Component Analysis for Clustering Gene Expression Data." *Bioinformatics*, vol. 17, no. 9, pp. 763–774, January 2001.

[9] J. Shi and Z. Luo. "Nonlinear dimensionality reduction of gene expression data for visualisation and clustering analysis of cancer tissue samples." *Computers in Biology and Medicine*, vol. 40, pp. 723–732, 2010.

[10] T. R. Golub, D. K. Slonim, P. Tamayo, et al. "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring." *Science*, vol. 286, pp. 531–537, 1999.

[11] M. Gutkin. *Feature selection methods for classification of gene expression profiles*. MSc Dissertation, Tel-Aviv University, 2008.

[12] Z. Li, W. Zhang, M. Wu, et al. "Gene Expression–Based Classification and Regulatory Networks of Pediatric Acute Lymphoblastic Leukemia." *BLOOD*, vol. 114, no. 20, pp. 4486–4493, 2009.

[13] K. Mills. "The MILE Study: Expression Microarray Analysis for Diagnosis of Leukaemia." Tech. rep., Queen's University Belfast, 2007.

[14] J. Khan, J. S. Wei, M. Ringnér, et al. "Classification and Diagnostic Prediction of Cancers using Gene Expression Profiling and Artificial Neural Networks." *Nature Medicine*, vol. 7, no. 6, pp. 673–679, 2001.

[15] M. A. Shipp, K. N. Ross, P. Tamayo, et al. "Diffuse Large B-cell Lymphoma Outcome Prediction by Gene Expression Profiling and Supervised Machine Learning." *Nature Medicine*, vol. 8, no. 1, pp. 68–74, 2002.

[16] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, et al. "Prediction of Central Nervous System Embryonal Tumour Outcome Based on Gene Expression." *Nature Medicine*, vol. 415, pp. 436–442, 2002.

[17] M. Riley, S. Letovsky, J. Garavelli, et al. "The Gene Ontology." `http://www.geneontology.org/`, 1999. Accessed March 2010.

[18] N. Blŭthgen, B. Cajavec, H. Herzel, et al. "Biological Profiling of Gene Groups utilizing Gene Ontology." *Genome Informatics*, vol. 1, no. 16, pp. 106—115, January 2005.

[19] S. J. Roberts. "Parametric and Non-parametric Unsupervised Cluster Analysis." *Pattern Recognition*, vol. 30, no. 2, pp. 261–272, May 1997.

[20] E. P. Solomon, L. R. Berg, and D. W. Martin. *Biology*. California: Brooks/Cole, Thomson Learning, 2005.

[21] S. Mitra, S. Datta, T. Perkins, and G. Michailidis. *Intoduction to Machine learning and Bioinformatics*. London: Chapman and Hall/CRC, first ed., 2008.

[22] P. Baldi and G. W. Hatfield. *DNA Microarrays and Gene Expression: from experiments to data analysis and modeling*. London: Cambridge University Press, first ed., 2002.

[23] Agilent Technologies. "Multiple Testing Corrections." `http://www.chem.agilent.com/cag/bsp/products/gsgx/Downloads/pdf/mtc.pdf`, 2005. Accessed December 2010.

[24] K. Cobb. "Microarrays: The Search for Meaning in a Vast Sea of Data." Tech. rep., Biomedical Computation Review, 2006.

[25] E. Bair, T. Hastie, D. Paul, and R. Tibshirani. "Prediction by Supervised Principal Components." Tech. rep., Stanford University, 2004.

[26] T. V. Prasad and S. I. Ahson. "Visualization of Microarray Gene Expression Data." Tech. rep., Bioinformation, 2006.

[27] T. Hastie, R. Tibshirani, B. Narasimhan, and G. Chu. "Supervised Learning from Microarray Data." Tech. rep., Stanford University.

[28] Y. Chen and Y. Zhao. "A Novel Ensemble of Classifiers for Microarray Data Classification." *Journal on Applied Soft Computing*, vol. 1, no. 8, pp. 1664—1669, January 2008.

[29] R. Rojas. *Neural Networks: A Systematic Approach*. Berlin: Springer, 1996.

[30] K. Mumtaz, S. A. Sheriff, and K. Duraiswamy. "Evaluation of the Three Neural Network Models using Wisconsin Breast Cancer database." *IEEE*, pp. 1–7, 2010.

[31] M. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler. "Knowledge-Based Analysis of Microarray Gene Expression Data by using Support Vector Machines." *Proceedings of National Academy of Sciences*, vol. 97, no. 1, pp. 262–267, 2000.

[32] C. Shou. "Supervised Learning of Microarray Gene Expression Data." Tech. rep., Program of Computational Biology and Bioinformatics, Yale University.

[33] Z. Ghahramani. *Advanced Lectures on Machine Learning*, p. 3. 2004.

[34] D. Jiang, C. Tang, and A. Zhang. "Cluster Analysis for Gene Expression Data: A Survey." *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1370–1386, November 2004.

[35] J. Vesanto and E. Alhoniemi. "Clustering of the Self-Organizing Map." *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586–600, 2000.

[36] A. M. Martinez and A. C. Kak. "PCA versus LDA." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, February 2001.

[37] R. E. Madsen, L. K. Hansen, and O. Winther. "Singular Value Decomposition and Principal Component Analysis." Tech. rep., Intelligent Signal Processing, 2004.

[38] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, third ed., 2001.

[39] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, fifth ed., 2007.

[40] S. Weng, C. Zhang, Z. Lin, and X. Zhang. "Mining the Structural Knowledge of High-Dimensional Medical Data using Isomap." *Medical & Biological Engineering & Computing*, vol. 43, pp. 410–412, 2005.

[41] J. B. Tenenbaum, V. de Silva, and J. C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction." *SCIENCE*, vol. 290, pp. 2319–2323, 2000.

[42] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. "On Cluster Validation Techniques." *Intelligent Information Systems*, vol. 17, no. 2, pp. 107–145, 2001.

[43] U. Maulik and S. Bandyopadhyay. "Performance Evaluation of Some Clustering Algorithms and Validity Indices." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, December 2002.

[44] G. R. Fowles. *Introduction to Modern Optics*, chap. 5, pp. 105–150. New York: Dover Publications, second ed., 1975.

[45] T. Sakai, T. Komazaki, and A. Imiya. "Scale-Space Clustering with Recursive Validation." pp. 288–299. Springer-Verlag, 2007.

[46] S. J. Roberts. "Scale-Space Unsupervised Cluster Analysis." pp. 106–110. IEEE, 1996.

[47] Y. Leung, J. Zhang, and Z. Xu. "Clustering by Scale-Space Filtering." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1396–1410, 2000.

[48] Z. Yao, W. Peng, C. Gao-Yun, et al. "Quantum Clustering Algorithm based on Exponent Measuring Distance." In *International Symposium on Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008.*, IEEE Catalog Number 10560087. IEEE, 2008. ISBN 978-1-4244-3530-2.

[49] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas. "SOM Toolbox for Matlab 5." Tech. rep., Helsinki University of Technology, 2000.

[50] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, 1981.