

Imperial College London
Department of Computing

Dimensionality Reduction Methods For Microarray Cancer Data Using Prior Knowledge

Zena Maria Hira

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College and
the Diploma of Imperial College, May 18, 2016

To my parents,

Yiannoula & Kyriacos

Declaration

This thesis presents my work in the Department of Computing at Imperial College London between March 2013 and February 2016. This work was done under the supervision of Professor Duncan Fyfe Gillies. I declare that the work presented in this thesis is my own, except where acknowledged. This thesis has not been submitted for the award of any degree or diploma in any other tertiary institution.

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

“It does not do to rely too much on silent majorities, Evey, for silence is a fragile thing, one loud noise, and its gone. ”

V, V for Vendetta

“Occam’s Razor should be called the Occam’s Principle of Limited Imagination.”

Agent Fox W. Mulder, The X-Files

Abstract

Microarray studies are currently a very popular source of biological information. They allow the simultaneous measurement of hundreds of thousands of genes, drastically increasing the amount of data that can be gathered in a small amount of time and also decreasing the cost of producing such results. Large numbers of high dimensional data sets are currently being generated and there is an ongoing need to find ways to analyse them to obtain meaningful interpretations. Many microarray experiments are concerned with answering specific biological or medical questions regarding diseases and treatments. Cancer is one of the most popular research areas and there is a plethora of data available requiring in depth analysis. Although the analysis of microarray data has been thoroughly researched over the past ten years, new approaches still appear regularly, and may lead to a better understanding of the available information. The size of the modern data sets presents considerable difficulties to traditional methodologies based on hypothesis testing, and there is a new move towards the use of machine learning in microarray data analysis.

Two new methods of using prior genetic knowledge in machine learning algorithms have been developed and their results are compared with existing methods. The prior knowledge consists of biological pathway data that can be found in on-line databases, and gene ontology terms. The first method, called “*a priori* manifold learning” uses the prior knowledge when constructing a manifold for non-linear feature extraction. It was found to perform better than both linear principal components analysis (PCA) and the non-linear Isomap algorithm (without prior knowledge) in both classification accuracy and quality of the clusters. Both pathway and GO terms were used as prior knowledge, and results showed that using GO terms can make the models over-fit the data. In the cases where the use of GO terms does not over-fit, the results are better than PCA, Isomap and *a priori* manifold learning using pathways.

The second method, called “the feature selection over pathway segmentation algorithm”, uses the pathway information to split a big dataset into smaller ones. Then, using AdaBoost, decision trees are constructed for each of the smaller sets and the sets that achieve higher classification accuracy are identified. The individual genes in these subsets are assessed to determine their role in the classification process. Using data sets concerning chronic myeloid leukaemia (CML) two subsets based on pathways were found to be strongly associ-

ated with the response to treatment. Using a different data set from measurements on lower grade glioma (LGG) tumours, four informative gene sets were discovered. Further analysis based on the Gini importance measure identified a set of genes for each cancer type (CML, LGG) that could predict the response to treatment very accurately ($> 90\%$). Moreover a single gene that can predict the response to CML treatment accurately was identified.

Acknowledgements

Doing a PhD needs more bricks than Pink Floyd's The Wall. It cannot be done without proper guidance. I would like to thank my supervisor Professor Duncan Gillies for taking me on and helping me through my PhD. Duncan has been an incredible supervisor, he is a constant source of advice and understanding. His vast array of knowledge can only be matched by his kind and caring nature. I do not think I would have made it without him.

Certainly it has been an intense experience with many ups and lows. I would like to thank everyone who was there for the good and the bad, to exchange ideas and problems. Everyone who has been a friend to me inside and outside Imperial. The people in the office: Kate, Loizos, Tran, Xhongliou, George R., Marily, George T., April, Nicolai, Claudia, Francis, Ramin, Oana, Paulina, Mark, Maria, Victoria, Fidelis, Amani, Marc, Jeremy, Francesca and Marek.

My friends in the "real world" Emma, Derek, Nicos, Elena, Caroline, Charis, Rania, George P. and Joanna for keeping me sane, out of the Imperial environment.

Tristan for all his care and attention both in the office and at home.

Last but not least I would like to thank my family and more than everyone parents, Yian-noula and Kyriacos, for their continuous love and support both emotional and financial throughout my years as a student. I also want to thank my aunt Maria for being there for me when I needed her.

Thanks everyone. It wouldn't be the same without you.

Contents

Acknowledgements	xi
1 Introduction	1
1.1 Motivation	1
1.1.1 Contribution	2
1.1.2 Thesis Scope	3
2 Microarray Data	5
2.1 From DNA To Gene Expression	5
2.1.1 From DNA To Cancer	7
2.2 DNA Methylation	7
2.3 Measuring Gene Expression Using Microarrays	8
2.3.1 Gene Expression And Methylation Profiling	10
2.3.2 Microarrays	12
2.4 Microarray Analysis	13
2.4.1 Statistical Analysis	17
2.4.2 Conclusion	25

3	Mathematical Background	27
3.1	Dimensionality Reduction	27
3.1.1	Principal Component Analysis	27
3.1.2	Manifold Learning - Isomap	29
3.2	Classification	33
3.2.1	Support Vector Machines	33
3.2.2	Linear Discriminant Analysis	34
3.2.3	K - Nearest Neighbours	34
3.3	Ensemble Methods	35
3.3.1	AdaBoost With Decision Trees As The Weak Classifiers	37
3.4	Internal Cluster Evaluation	38
3.4.1	Dunn Index	38
3.4.2	Davies–Bouldin index	38
3.4.3	Silhouette	39
3.5	Cross-Validation	40
3.5.1	Stratified Cross-Validation	40
3.5.2	Receiver Operating Characteristic Curves	41
4	Dimensionality Reduction For High-Dimensional Microarray Datasets	42
4.1	Introduction	42
4.2	The Curse Of Dimensionality	43
4.3	Feature Subset Selection In Microarray Cancer Data	44
4.3.1	Filters	45

4.3.2	Wrappers	49
4.3.3	Embedded Techniques	54
4.4	Feature Extraction In Microarray Cancer Data	57
4.4.1	Linear	57
4.4.2	Non-Linear	58
4.4.3	Affinity Matrices	61
4.5	Prior Knowledge	63
4.5.1	Gene Ontology	64
4.5.2	Protein-Protein Interaction	65
4.5.3	Gene Pathways	65
4.6	Discussion	67
5	<i>A priori</i> Manifold Learning	69
5.1	Introduction	69
5.2	The <i>A Priori</i> Manifold Learning Algorithm	70
5.3	Datasets	73
5.4	Results	75
5.4.1	Internal Evaluation	76
5.4.2	Ten Fold Cross-Validation	79
5.4.3	Pathway Robustness	90
5.5	Discussion	91

6	<i>A Priori</i> Manifold Learning Using GO Terms	93
6.1	Introduction	93
6.2	GO Terms Similarity Measures	95
6.2.1	Information Content-Based Methods	95
6.2.2	Graph-Based Method	97
6.3	Results Using The Jaccard Coefficient	98
6.3.1	Internal Evaluation	98
6.3.2	Ten Fold Cross-Validation	100
6.4	Results Using Wang Similarity	105
6.4.1	Comparison With <i>A Priori</i> Manifold Learning Using Pathways	106
6.5	Discussion	106
7	Identifying Significant Features In Cancer Methylation Data	108
7.1	Introduction	108
7.2	Feature Selection Over Pathway Segmentation Algorithm	110
7.3	Datasets	111
7.4	Experiments Using The Pathway Algorithm	113
7.4.1	Baseline Experiments	113
7.4.2	TCGA Lower Grade Glioma	114
7.4.3	Chronic Myeloid Leukaemia	122
7.5	Discussion	127
7.6	<i>A priori</i> Manifold Learning Over Pathway Segmentation	129

8 Conclusion	130
8.1 Thesis Summary	130
8.2 Future Work	133
8.2.1 <i>A Priori</i> Manifold Learning Using Locally-Linear Embedding	133
8.2.2 Learning Causality Using Bayesian Networks	133
Bibliography	138
Appendix	168
A Cell Cycle	169
B <i>A priori</i> Manifold Learning	171
B.1 Variances	171
B.2 Accuracy error	174
B.3 Pathway Robustness	177
C <i>A priori</i> Manifold Learning using GO terms	182
C.1 ROC Curves	182
D Acronyms	189

List of Tables

2.1	Gene expression & Methylation profiling differences	12
2.2	One - Way ANOVA Test	23
2.3	Two - Way ANOVA Test Observation Sums	23
2.4	Two - Way ANOVA Test	23
2.5	ANOVA vs Regression	24
4.1	Deterministic VS Randomised Wrappers	50
4.2	Feature Selection methods applied on microarray data	55
4.3	Advantages and Disadvantages between Feature Selection And Feature Ex- traction Approaches	61
5.1	Datasets Used	75
5.2	Cancer classification accuracy measured using the k -Nearest Neighbours al- gorithm and Ten Fold Cross validation	81
5.3	Cancer classification accuracy measured using the k -Nearest Neighbours al- gorithm and Ten Fold Cross validation	81
5.4	Cancer classification accuracy measured using Linear Discriminant Analysis and Ten Fold Cross validation	82
5.5	Cancer classification accuracy measured using Linear Discriminant Analysis and Ten Fold Cross validation	82

6.1	Genes HAAO Mapped to GO Terms	94
6.2	10 Fold Cross Validation Accuracy using k -Nearest Neighbours	101
6.3	Cancer classification accuracy measured using the k -Nearest Neighbours algorithm and Ten Fold Cross validation	101
6.4	Cancer classification accuracy measured using the k -Nearest Neighbours algorithm and Ten Fold Cross validation	102
6.5	10 Fold Cross Validation Accuracy using Support Vector Machines	102
6.6	10 Fold Cross Validation Accuracy using Linear Discriminant Analysis	103
6.7	Cancer classification accuracy measured using Linear Discriminant Analysis and ten fold cross validation	103
6.8	Cancer classification accuracy measured using Linear Discriminant Analysis and ten fold cross validation	104
7.1	Datasets Used	112
7.2	Accuracy of the original datasets	114
7.3	Gene pathway sets with the highest scores for LGG	115
7.4	Logistic Regression applied on LGG gene pathway sets	116
7.5	Gene List for LGG	121
7.6	Gene pathway sets with the highest scores for CML	122
7.7	Logistic Regression applied on CML gene pathway sets	124
7.8	Gene List for CML	127
7.9	Computational Complexity Using Pathway Segmentation	127
8.1	Comparison between Isomap and Locally Linear Embedding	133
A.1	The stages of the cell cycle	170

B.1	10 Fold Cross Validation Variance On a Gene-by-Gene Transformation using k -Nearest Neighbours	172
B.2	10 Fold Cross Validation Variance On a Sample-by-Sample Transformation using k -Nearest Neighbours	172
B.3	10 Fold Cross Validation Variance On a Gene-by-Gene Transformation using Linear Discriminant Analysis	173
B.4	10 Fold Cross Validation Variance On a Sample-by-Sample Transformation using Linear Discriminant Analysis	173

List of Algorithms

2.1	Bootstrapping	25
3.1	k -Fold Cross Validation	41
4.1	Genetic Algorithms	52
4.2	Simulated Annealing Algorithm	53
4.3	Best Incremental Ranked Subset (BIRS)	53
5.1	Calculation of the k -Nearest points of the manifold	73
7.1	Gene Selection Algorithm based on accuracy thresholds and how important each feature is when constructing the decision tree	120
8.1	Kruskal's algorithm	135
8.2	Boosting algorithm	135
8.3	Marginal Independence	137
8.4	Cyclic Bayesian Networks	138

List of Figures

2.1	DNA Functions	6
2.2	Microarray Chips	8
2.3	How hybridisation takes place on microarrays	9
3.1	Manifold example with the 3D Swiss roll dataset with $N = 1000$ data points .	30
3.2	Support Vector Machines	33
3.3	The Boosting Algorithm	36
3.4	The Bagging Algorithm	36
3.5	The Stacking Algorithm	37
4.1	Comparison between EWUSC, USC and SC on breast cancer data	46
4.2	Comparison between Relief-F, Information Gain, Information Gain Ratio and X^2 test on ALL and MLL Leukaemia datasets	49
4.3	Computational Complexity Comparison	56
4.4	Dimensionality reduction using linear matrix factorization	57
4.5	Visualisation of a Leukaemia dataset with PCA, manifold LLE and manifold Isomap	59
4.6	Feature Selection And Feature Extraction	61

5.1	Endometrium Cancer: How the η value affects the value for the Dunn index	72
5.2	Evaluation benchmark	76
5.3	Dunn Index for cancer classification results	78
5.4	Dunn Index for cancer classification results	79
5.5	ROC curves for cancer classification using k -Nearest Neighbours	84
5.6	ROC curves for cancer classification using k -Nearest Neighbours	85
5.7	ROC curves for cancer classification using support vector machines	86
5.8	ROC curves for cancer classification using support vector machines	87
5.9	ROC curves for cancer classification using Linear Discriminant Analysis	88
5.10	ROC curves for cancer classification using linear discriminant analysis	89
5.11	Leukaemia cells	90
6.1	Gene Product Association with GO Term	94
6.2	GO Terms Relationships	94
6.3	Dunn Index for cancer classification results	99
6.4	Dunn Index for cancer classification results	100
7.1	Pathway Algorithm	110
7.2	ROC Curve for the original LGG and CML datasets	114
7.3	ROC curves for the LGG Pathway Sets	115
7.4	Pantothenate and CoA biosynthesis and Retinoate Biosynthesis II pathway set	116
7.5	Comparison between Pantothenate and CoA biosynthesis and two other pathway sets	116
7.6	50% of the genes were removed from the IL2 and Pantothenate and CoA biosynthesis pathway set	117

7.7	20% of the genes were removed from the IL2 and the Pantothenate and CoA biosynthesis pathway set	118
7.8	50% of the genes were removed from the IL2 pathway set	118
7.9	Comparison between the IL2 and the Pyrimidine metabolism with random sets	118
7.10	ROC curves for the CML Pathway Sets	122
7.11	Regulation of KIT signalling pathway set	123
7.12	Comparison between Regulation of KIT signalling and two other pathways .	123
7.13	Comparison between Regulation of KIT signalling and Acetylcholine Synthesis	124
7.14	60% of the genes were removed from the Regulation of KIT signalling pathway	124
7.15	80% of the genes were removed from the Regulation of KIT signalling pathway	125
7.16	Comparison between Regulation of KIT signalling with random sets	125
7.17	Comparison between Regulation of KIT signalling and a random pathway set of 728 genes	126
8.1	A Collider	136
A.1	Cell Cycle	169
B.1	Accuracy with variance for all nine datasets using the gene-by-gene affinity matrices k -Nearest Neighbours	174
B.2	Accuracy with variance for all nine datasets using the sample-by-sample affinity matrices using k -Nearest Neighbours	175
B.3	Accuracy with variance for all nine datasets using the gene-by-gene affinity matrices using Linear Discriminant Analysis	176
B.4	Accuracy with variance for all nine datasets using the sample-by-sample affinity matrices using Linear Discriminant Analysis	177

B.5	Pathway Robustness (Endometrium)	178
B.6	Pathway Robustness (Prostate)	178
B.7	Pathway Robustness (Lung)	179
B.8	Pathway Robustness (Breast)	179
B.9	Pathway Robustness (Colon)	180
B.10	Pathway Robustness (Kidney)	180
B.11	Pathway Robustness (Omentum)	181
B.12	Pathway Robustness (Ovary)	181
C.1	ROC curves for gene-by-gene affinity matrices using k -Nearest Neighbours .	183
C.2	ROC curves for sample-by-sample affinity matrices using k -Nearest Neighbours	184
C.3	ROC curves for gene-by-gene affinity matrices using Linear Discriminant Analysis	185
C.4	ROC curves for sample-by-sample affinity matrices using Linear Discriminant Analysis	186
C.5	ROC curves for gene-by-gene affinity matrices using Support Vector Machines	187
C.6	ROC curves for sample-by-sample affinity matrices using Support Vector Machines	188

Chapter 1

Introduction

1.1 Motivation

New microarray technologies have revolutionised the way genetic experiments have been conducted by allowing the measurement and analysis of many thousands of genes in parallel. Analysing microarray data is a continuous challenge due to the large number of variables and the noise often associated with the measurements. This high throughput method generates thousands of variables that must be properly analysed to get some meaningful results. The relations among the different variables can make the analysis very difficult [Mur02]. Pairwise relations exist [VP13, RGS08] along with clusters of many related genes [MCYC06, ESBB98]. For any experiment there is little evidence to suggest in advance how the genes are related. Despite these difficulties, microarrays have proven useful when it comes to understanding the genetics of a disease, and have led to the development of new drugs and therapies. Microarrays have been used to assess many different types of cancerous material including leukaemia (acute myeloid leukaemia and acute lymphoblastic leukaemia) [GST⁺99] breast cancer [GST⁺99, SPT⁺01] and lung cancer [GTS⁺01, BRS⁺01].

A lot has been done since the first analytical attempts to cluster microarray data in order to reveal biologically meaningful patterns [ESBB98]. A number of studies have used statistical and machine learning techniques. Some studies came up with promising results that can

be used to improve the treatment of certain diseases. There is however a lot of room for improvement since due to the number of variables and experimental variance many relationships and potential information can go unnoticed.

We will focus on cancer data since cancer research is currently a very active field. Many biological microarray experiments are conducted with the sole purpose of gaining more information on what causes cancer, how to improve treatment and to discover how people respond to treatment. There is still a lot to be done due to the multi-factorial nature of cancer. It depends on genetic, environmental, medical and life style factors which can be very difficult to capture in one study.

1.1.1 Contribution

This thesis is focused on methods for reducing the dimensionality of the microarray datasets in order to make classification easier and less computationally expensive. Principal Components Analysis (PCA) and manifold learning (Isomap) are popular methods for linear and non-linear dimensionality reduction respectively. The novel aspect of the work is investigating ways in which prior knowledge about gene behaviour can be incorporated in dimensionality reduction to improve classification. One main source of prior knowledge that was used was extracted from genetic pathway databases. Several techniques such as Support Vector Machines, Linear Discriminant Analysis, k - Nearest Neighbours, Decision Trees and boosting were used in the development of two novel algorithms which are called “*a priori* manifold learning ” and “Feature selection over pathway segmentation”. The first one changes the distance metric between genes to incorporate pathway information when building a manifold, and the second uses pathway information to split a huge dataset that previously could not be analysed due to its size, into potentially informative subsets of genes.

***A priori* Manifold Learning Algorithm**

For some datasets, especially high-dimensional cancer datasets with a lot of noise, classifiers often fail in producing accurate results. To overcome this *a priori* manifold learning was proposed for finding a manifold on which gene relationships can be identified. A representative set of microarray data is fused with relevant data taken from the KEGG pathway [Kan97] or the GO term databases [ABB⁺00, Con15]. Once the manifold has been constructed the raw microarray data is projected onto it and clustering and classification can take place.

Feature Selection Over Pathway Segmentation

This approach incorporates pathway information found on the ConsensusPath [KSLH13, KPG⁺11, KWLH09, POH⁺10] database with boosting techniques to predict the response of patients to treatment. Analysing the complete data sets using standard machine learning methods is currently infeasible due to the huge number of variables and small number of patients. Adaboost was used with decision trees as weak classifiers on a high-dimensional dataset of methylation profiles of a set of patients suffering from chronic myeloid leukaemia (CML) or lower grade glioma (LGG). The genes from two individual pathways were found to be associated with CML, along with a gene that seems to be predicting the response to treatment with a very high accuracy. In addition gene sets from four pathways were identified related to LGG. Investigating the decision trees further, it was possible to synthesise two lists of genes that can perform very well in classifying response in LGG and CML treatment (> 90% accuracy).

1.1.2 Thesis Scope

The focus of this thesis is on the development of machine learning methods that can identify patterns of interactions in genes involved in cancer. The main contributions concern the analysis of microarray data based on the assumption that the data was gathered and normalised correctly. The datasets used have been suitably processed by the biologists who

published them. The main results are in demonstrating the viability of the methods proposed, and further work is required to validate them in a clinical context.

Chapter 2

Microarray Data

2.1 From DNA To Gene Expression

Cells are the basic structural, functional and biological units of living organisms. There are two types of cells: prokaryotic and eukaryotic. Eukaryotic cells are more evolved than prokaryotic cells. Their main difference is the absence of a nucleus in the prokaryotic cells. Deoxyribonucleic acid (DNA) is contained in the nucleus, and is formed by two complementary strands of nucleotides composed of adenine, thymine, cytosine and guanine [Alb08].

The central dogma of genetics is: “DNA makes RNA and RNA makes protein”, a positive statement which was originally termed “the sequence hypothesis” by Crick, first stated in 1956 [Cri58, CRI70]. DNA contains the code for making proteins, needed by the living cells. DNA can be copied (DNA replication), it can be transcribed into messenger RNA (transcription), and proteins can be synthesized using the information in mRNA as a template (translation) [Alb08] as shown in figure 2.1. The production of mRNA is often referred to as gene expression.

DNA replication produces two identical replicas from one original DNA molecule. This biological process occurs in all living organisms and is the basis for biological inheritance. Transcription is the process by which the information contained in a section of the DNA is

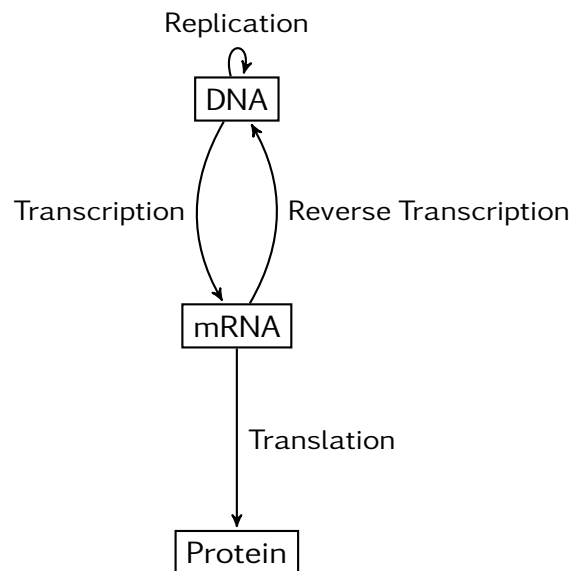


Figure 2.1: **DNA Functions**

replicated in the form of a newly assembled piece of mRNA. mRNA takes part in protein generation by transporting the genetic information of DNA out of the cell. There are three steps to the process of DNA transcription:

1. **Initiation:** RNA Polymerase (enzyme) binds to DNA. Specific nucleotide sequences tell RNA polymerase where to begin and where to end (promoter region).
2. **Elongation:** Proteins, called transcription factors unwind the DNA and allow RNA polymerase to transcribe only a single strand of DNA into a single stranded RNA polymer called messenger RNA (mRNA).
3. **Termination:** RNA polymerase moves along the DNA until it reaches a terminator sequence.

Translation is the process in which cellular ribosomes create proteins. mRNA transcription from DNA is decoded by a ribosome to produce a specific amino acid chain, or polypeptide. The polypeptide later folds into an active protein and performs its functions in the cell [Alb08].

2.1.1 From DNA To Cancer

Cells have the property to divide and produce new cells either for repairing the old ones or for letting the body grow. Changes in the DNA attributed to both genetic and epigenetic factors¹ can cause the uncontrolled division of cells. This is called cancer. Cancer cells displace normal cells and use up all of the nutrients in the body to fuel their own uncontrolled growth, which can cause problems to the correct function of the different organs. The term cancer is used to describe the abnormal growth of cells that can, for example, form extra tissue called mass and attack other organs [dMFF⁺12]. s Cancer is among the leading causes of death worldwide accounting for more than 8 million deaths according to the World Health Organization. It is expected that the deaths from cancer will rise to 14 million in the next two decades. Cancer is not a single disease. There are more than 100 known different types of cancer and probably many more.

Microarray databases are a large source of genetic data, which upon proper analysis, could enhance our understanding of biology and medicine. Many microarray experiments have been designed to investigate the genetic mechanisms of cancer. In the last ten years, machine learning techniques have been investigated in microarray data analysis. Several approaches have been tried in order to: (i) distinguish between cancerous and non-cancerous samples; (ii) classify different types of cancer and (iii) to identify subtypes of cancer that may progress aggressively. All these investigations are seeking to generate biologically meaningful interpretations of complex datasets that are sufficiently interesting to drive follow-up experimentation.

2.2 DNA Methylation

The exact role of methylation in gene expression is unknown, but it is known that DNA methylation is essential for cell differentiation and embryonic development [Cai08]. Methy-

¹Changes in the regulation of gene activity and expression that are not dependent on gene DNA sequence but on external factors (i.e. exposures to toxins).

lation occurs at CpG islands when a methyl group is added to a cytosine residue to convert it to 5-methylcytosine. A CpG site is a place on the linear sequence of the bases of the DNA that has a cytosine and guanine separated by only one phosphate. Methylation of these sites that are in the promoter regions of genes can affect their expression and lead to their silencing, a feature found in a number of human cancers [JL99]. Methylation can be used to identify biomarkers for a number of diseases, since it can provide information about environmental exposures [WH12]. Methylated sites can be detected using bisulfite conversion, methylation-sensitive restriction enzymes, methyl-binding proteins and anti-methylcytosine antibodies. These techniques are combined with microarray technology and high throughput sequencing in order to provide means for genome-wide analysis of the given samples.

2.3 Measuring Gene Expression Using Microarrays

Microarray technology appeared in mid 90s. A DNA microarray or a DNA chip is a high-throughput technology for measuring the expression levels of genes. It consists of a small membrane or glass slide containing many genes arranged in a regular pattern (see figure 2.2). Probes are short strands of DNA, that are used to hybridise the target of interest (usually complementary DNA - cDNA) as shown in figure 2.3. A high number of complementary base pairs in a nucleotide sequence means stronger bonding between the two strands.

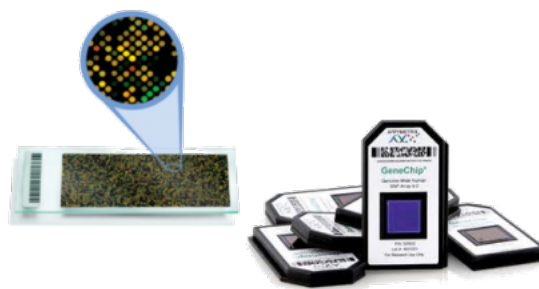


Figure 2.2:

²Microarray Chips

²Adapted from http://www.nucleomics.be/wp-content/uploads/et_temp/microarrays_AgAndAffy_transparent-108569_406x226.png

The samples in a microarray experiment are fluorescently labelled, and the measured fluorescence at each spot on the microarray indicates the amount of the corresponding genetic material in the sample. The fluorescence intensity is captured by scanners into an image [Lee04].

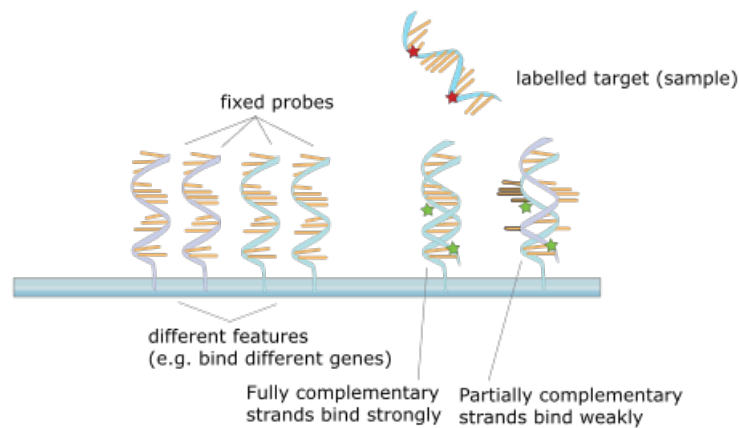


Figure 2.3:

³How hybridisation takes place on microarrays

Microarray technologies are available for measuring gene expression, DNA copy number, methylation, chromatin state, protein binding, SNPs, and other aspects of gene physiology [Rei10] and they have a number of uses including:

1. Gene Discovery: Identification of new genes and their functions.
2. Disease Diagnosis: Characterising different diseases and their genetic differences.
3. Drug Discovery / Pharmacogenomics: Finding relations between therapeutic responses to drugs and the genetic profiles of the patients.
4. Toxicological Research: Investigating the impact of toxins on the cells.

Microarrays have several advantages that explain why they are so widely used such as: affordability, no specific equipment is required for their use and, flexibility of design as they can be manufactured according to the specific experiment [HN09].

³Adapted from http://upload.wikimedia.org/wikipedia/en/thumb/a/a8/NA_hybrid.svg/500px-NA_hybrid.svg.png

2.3.1 Gene Expression And Methylation Profiling

During this research two different types of data were used: i) gene expression profiling and ii) methylation profiling data.

Gene Expression Profiling Using Microarrays

Expression profiling is the process of correlating the genotype (genetic make-up) and the phenotype (a characteristic of a gene). Not all genes are active and therefore not all genes are expressed. Changes in the expression of the gene can result in a number of genetic diseases. Gene Expression Profiling provides a comprehensive view of the genes' activity in the samples. It quantifies the concentration of a gene's mRNA transcript in a cell which is the gene's expression.

Microarrays immobilise DNA probes on the glass using cDNA, utilising the property of cDNA sequences to attach to DNA by forming hydrogen bonds⁴ in a process called hybridisation, which detects the presence of nucleotide sequences that are complementary to the sequence in the probe. The DNA that was bound to each probe is measured as described above.

Methylation Profiling Using Microarrays

Three major techniques exist for detecting methylated sites for microarray analysis:

- Methylation-sensitive restriction enzymes: Restriction enzyme-based methods either enrich for methylated DNA or unmethylated DNA. They are able to detect and cleave methylation sites for further analysis by looking at the presence of specific enzymes [LB].
- Bisulfite conversion: Methylated cytosine, one of the four main bases found in DNA and RNA, has almost the same base-pairing characteristics as unmethylated cytosine,

⁴Electrostatic attraction between polar molecules

and is thus indistinguishable by standard sequencing approaches. To overcome this, genomic DNA can be treated with sodium bisulfite. Under appropriate conditions, this treatment causes deamination⁵ of unmethylated cytosine to uracil, while leaving methylated cytosine intact [PMQC11, SHPF94].

- Affinity purification: Methyl-binding domain (MBD) proteins, such as Mbd1, are used to obtain DNA fractions that are methylated by detecting CpG islands. Mbd1 is particularly useful since it recognizes all CpG sites while most methods for methylation detection are only able to recognise a small fraction of them. Affinity purification also works with an antibody, instead of a protein, that can recognize methylated cytosines [RMM, WDW⁺05].

The process of methylation profiling using microarrays measures the methylation levels at specific locations within the genome using the same process as gene expression profiling. The probes on the array now represent CpG islands. Every probe is composed of two beads: one for the methylated cytosine region and another one for the unmethylated. Fluorescent methylated (using the above methods) and unmethylated DNA is used to hybridise the microarray. The fluorescence intensity ratios between the two beads are used to calculate the methylation level.

Differences Between Gene Expression And Methylation Profiling

The difference between the gene expression and the methylation data is shown in table 2.1. There is a connection between the expression and the methylation data since the presence of methylation near the transcription site of a gene is associated with reduced expression [WBG⁺14] and methylation near the gene promoters varies depending on cell type, with more methylation of the promoters correlated with low or no transcription [SB08].

⁵Removal of an amine group from a molecule

Gene expression	Methylation
Gene expression levels are usually related to the production of a protein	Methylation levels are related to identifying methylated regions on the DNA which could be related to silencing a gene and embryonic development
Genetic mechanism	Epigenetic mechanism ⁶
Adjacent gene pairs on a chromosome tend to be co-expressed [KYQG03]	Adjacent probes are often correlated due to the regional nature of methylation [PSYK12]

Table 2.1: **Gene expression & Methylation profiling differences**

2.3.2 Microarrays

The following microarrays are widely used for both methylation and gene expression data.

- Bead arrays (Illumina) : gene-specific probes attached to beads and assembled into arrays.
- Short oligonucleotide⁷ arrays (Affymetrix): produced by printing short oligonucleotide sequences designed to represent a single gene or family of gene splice-variants⁸ by synthesizing this sequence directly onto the array surface instead of depositing intact sequences (25-nucleotides long)
- Long oligonucleotide⁹ arrays (NimbleGen and Agilent): Same as Short oligonucleotide arrays but with longer probes (60-nucleotides long).

⁶Changes in the regulation of gene activity and expression that are not dependent on gene DNA sequence but on external factors (i.e. exposures to toxins).

⁷Oligonucleotides are short, single-stranded DNA or RNA molecules that have a wide range of applications in genetic testing, research, and forensics.

⁸Gene splicing is a post-transcriptional modification in which a single gene can code for multiple proteins.

⁹Longer probes are more specific to individual target genes, shorter probes may be spotted in higher density across the array and are cheaper to manufacture.

One-Colour Microarrays And Two-Colour Microarrays

1. In one-colour microarrays or single-channel microarrays, each probe or probe set provides a relative level of hybridisation with the labelled target. They indicate the relative presence of a gene when compared to other samples or conditions when processed in the same experiment.
2. Two-colour microarrays or two-channel microarrays have complementary DNA (cDNA¹⁰) prepared from two samples to be compared (e.g. diseased tissue versus healthy tissue) and they are labelled with two different fluorophores [SSB96]. The two different types of DNA samples are mixed and hybridised to a single microarray. Relative intensities of each fluorophore may then be used in ratio-based analysis to identify up-regulated and down-regulated genes [TFG⁺07].

2.4 Microarray Analysis

Microarrays contain between 10,000 and 400,000 different probes. This allows researchers to assess simultaneously nearly all the genes in the genome. However they generate large and complicated data sets that are not easy to analyse and validate [Lar05]. The microarray analysis process works as follows:

1. Experimental Design: An experiment is designed to answer a question. A crucial issue is whether the number of samples is enough to make the differences in the expression obvious. Good design principles involve:
 - Replication: to reduce uncertainty and variability. Replication should be both biological and technical. Biological replication means having multiple samples per group (e.g. healthy, diseased) and technical replication involves RNA from one sample to be hybridised in multiple arrays.

¹⁰Synthesized from a messenger RNA (mRNA) template in a reaction catalysed by the enzyme reverse transcriptase [AJL⁺02]

- Blocking: to minimise the effect of existing blocks. This is when samples or processing is not homogeneous due to a number of factors (treated under different conditions, different medication). Local control or blocking is used to reduce the effects of this. The scientists group experimental units together so that the variability in the groups is less than among the different units before the grouping.
 - Randomisation: to ensure the validity of the results. Randomisation can be over the individuals assigned to experiments or the order in which the experiments run.
2. Perform Experiment: DNA is taken from the different samples and is combined with the microarrays.
 3. Quality Assurance: After the experiment the microarrays are checked to determine the data that is worth analysing. The chips are checked against some ideal reference values and using deviation plots the technician decides what to keep.
 4. Normalisation: Normalisation is the attempt to compensate for systematic technical differences between chips, to see more clearly the biological differences between samples [DR04, DHHR02, HvHS⁺02]. Normalising the microarray data can be done in three different ways [TOR⁺01, YS02, YBS01]: Global, Linear and Non-Linear (or LOWESS). Let $\log G$ and $\log R$ be the Green and Red background-corrected intensities, where G and R are the pixel intensities for Green and Red colours respectively. For constructing the MA plot¹¹: $M = \log_2(R/G)$ (log-ratios) and $A = 1/2\log_2(RG)$ (mean average).
 - (a) Global normalisation: Normalisation parameters are chosen from the whole of the array using the global median of log intensity ratios. a_0 is the median of M as shown in equation 2.1.
 - (b) (Intensity dependent) Linear normalisation: For two-channel microarrays it is assumed that the intensities of the two channels have an equal median and therefore the normalisation is done by introducing a constant scaling factor making

¹¹Plot for visualising two channel microarrays

one channel to match the other. (β_0, β_1) are the least squares estimations for M and A as shown in equation 2.1.

- (c) (Intensity dependent) Non-Linear normalisation (LOWESS): This procedure assumes that the differences are non-linear. It was originally known as locally weighted polynomial regression, which applies a smooth curve through a set of data points. LOWESS fits a low-degree polynomial to a subset of the data, using weighted least squares giving more weight to the points whose response is being estimated and less to those that are far away.

The normalisation factor for each probe is calculated individually: $M_i^*(A_i) = M_i - \hat{c}(A_i)$ where $\hat{c}(A_i)$ is the normalisation coefficient for the probe.

A summary is shown in equation 2.1 [PYK⁺03].

$$M = \begin{cases} a_0, & \text{for global normalisation} \\ \beta_0 + \beta_1 A, & \text{for linear normalisation} \\ c_A, & \text{for non-linear normalisation} \end{cases} \quad (2.1)$$

Measurements can be affected because of a number of things:

- Different amounts of RNA are used for the samples and this can affect the amount of RNA that is being attached to the probe and consequently its measurement.
- Differences in the dyes: One dye is more readily incorporated than the other (in 2-colour systems) as Red and Green colours have different scanning properties (detection efficiencies, frequencies); Different amounts of labelling may occur (in one-colour systems) as there tends to be variation of the dye intensity when the microarray is being printed depending on the position of the probe on the glass.
- Hybridization conditions may vary (e.g. room temperature, hybridisation time or even physical differences of the glass of each microarray).
- There may be scanning errors.

The last two points are called background noise. The scanning laser could be affected by a surface reflection (this can never be completely eliminated) or by changing the sensitivity of the scanning device. It could also be affected by debris or salts left over from hybridisation [Van07]. Noise is handled at the normalisation step. A number of methods exists to deal with noise: removing potential outliers [KH10], applying a hierarchical Bayesian ANOVA model [PFS11] and the MAQC-I, MAQC-II, MAQC-III (SEQC) [MAQ06] projects which provide a collection of tools for quality control, normalisation and predictive models. It is uncertain how much the noise reduction helps with the analysis [LK07]. *“...the random fluctuations of gene expression signals caused by technical noise are quite low and the effect of such fluctuations on the results of statistical inference from Affymetrix GeneChip microarray data is negligibly small”* [LK07].

5. Summarisation: Some microarrays like Affymetrix and NimbleGen use many probes to target one gene (multiple probe oligonucleotide) in order to provide a better measure of the expression. There are many statistical algorithms that are used to summarise the gene expression in various probes (MBEI/dChip (Model-Based Expression Intensities) [LW01], RMA (Robust MultiChip Average) [IBC⁺03], gcRMA (GeneChip Robust MultiChip Average) [WI05], FARMS (Factor Analysis for Robust Microarray Summarization) [HCO06], gMOS [MML03] and BGX [Hei05]) [LEAK11]. In addition there are simpler methods such as: maxf (takes the probe with the highest fold change¹²), minp (takes the probe with the minimum p-value) and wmean (finds the weighted mean based on the probes p-values).

6. Analysis of the expression results by statisticians.

Despite their vast use microarrays have disadvantages. Identifying correlations in results does not mean causation. They could as well be accidental or derive from a common cause. Microarrays do not have the ability to show changes at the protein level. They only allow profiling at the mRNA level [PTOK00]. Most of the changes on the microarray gene expression do not correspond to any protein production and they are influenced by other minor

¹²The ratio of expression values between two conditions

changes in the experimental conditions [PKMJ07].

2.4.1 Statistical Analysis

The format of the data extracted from the microarray experiment is normally the following:

$$\mathbf{I}_n = \underbrace{\begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{y,1} & x_{y,2} & \cdots & \cdots & x_{y,n} \end{pmatrix}}_X \Bigg\} Y$$

where $X \gg Y$ since rows represent patients or samples and columns represent genes. We invariably have more genes than patients in most of these studies.

Microarray data analysis is used to identify genes that are:

- affected by a treatment
- marker genes that discriminate a diseased from a healthy subject

These genes are used for further experiments and more precise studies regarding a disease or an experimental treatment.

There are several ways microarrays are used with the most common being having a reference cell (control) A that is compared to another cell B which has been receiving treatment for a specific condition. Both cells have the same set of genes in them.

One of the first important experiments using microarrays was done by Spellman [SSZ⁺98]. He used microarrays for investigating how the gene expression in yeast changes under different stress conditions (heat, carbon, etc.) and during different phases of the cell cycle. The cell cycle is the process by which the cell divides (more information can be found in

appendix A). Ideally in the cell cycle there is a balance of dividing and dying cells. Cells dividing and growing abnormally is a characteristic of cancer [WS12].

Statistical Tests

Statistical tests can be used: for example the z-score, the t-test [Box87] and the Wilcoxon sign rank / rank-sum tests, that calculate the difference in the means of two groups and therefore give the distance between the two groups in units of standard deviation. Having a hypothesis H_0 (null hypothesis) for two groups T and C :

$$H_0 : \mu_T = \mu_C$$

where μ is the mean.

Hypothesis testing is often associated with 2 types of error:

Type I: For falsely rejecting the null hypothesis;

Type II: For falsely accepting the null hypothesis

To accept or reject a null hypothesis the p-value is used. The p-value is just the probability that a data point belongs to the distribution representing a null hypothesis. In our case, small p-values indicate that it is very unlikely that the observed pattern is random (null hypothesis). This is because very small p-values are found in the tails of the normal distribution. The decision to accept or reject a null hypothesis is given by the confidence interval. Confidence intervals can be used for answering the question of statistical significance, regardless of whether a p-values is calculated. A confidence interval is used for the purpose of estimating a population parameter with a level of confidence. Since sample results vary, a measure of variability, or margin of error, needs to be added in the results. The sample statistic, plus or minus a margin of error, gives a range of likely values for the parameter. This is the confidence interval. The 95% confidence interval consists of all values less than 1.96 standard errors away from the sample value. Similarly, the 99% confidence interval consists of all values less than 2.58 standard errors away from the sample value.

Z - Score

The z-score is a measurement of a score's relationship with the mean. Using the standard normal distribution it converts a group of data such that the mean is 0 and the standard deviation is 1. The z-score represents the distance between a sample's raw score and the population's mean in units of the standard deviation. The z-score is shown in equation 2.2, where μ is the mean and σ is the standard deviation.

$$z = \frac{x - \mu}{\sigma} \quad (2.2)$$

T - Test

The T - test can be used to establish whether the difference in the mean is significant or not using the t-test formula in equation 2.3.

$$t = \frac{\bar{X}_T - \bar{X}_C}{\sqrt{\frac{var_T}{n_T} + \frac{var_C}{n_C}}} \quad (2.3)$$

where $\bar{X}_T$ is the mean of group T and $\bar{X}_C$ the mean for group C, var_T and var_C are the variances for the different groups and n denotes the number of samples in the group. The value t can be used to check whether the difference is significant and accept or reject the null hypothesis H_0 . The T-test assumes that the groups compared follow a normal distribution and their standard deviation should be equal.

Wilcox Signed Rank Test

The Wilcox signed rank test [Wil45] is different to the t-test as it does not compare the means of the groups. It assesses whether the population mean ranks differ. It is a non-parametric test that does not assume a normal distribution on the data.

1. Calculate the absolute difference and the sign function *sign* between each pair of values:

for $i = 1, \dots, N$:

$$|x_{Ti} - x_{Ci}|$$

$$\text{sign}(x_{Ti} - x_{Ci})$$

excluding pairs where:

$$|x_{Ti} - x_{Ci}| = 0$$

2. Order the remaining pairs from smallest to largest absolute difference (N)
3. Rank the pairs with the smallest first. If there is a tie the rank received is equal to the average of the ranks they span across. Rank is denoted by R_i
4. Calculate the W test statistic as shown in equation 2.4.

$$W = \sum_{i=1}^N [\text{sign}(x_{2,i} - x_{1,i}) \cdot R_i] \quad (2.4)$$

5. For a large N : W can be used to calculate the Z-score and accept or reject the null hypothesis:

$$z = \frac{W}{\sigma_W} \quad (2.5)$$

where σ_W is:

$$\sigma_W = \sqrt{\frac{N_r(N_r + 1)(2N_r + 1)}{6}} \quad (2.6)$$

Regression Analysis

Regression Analysis is often used with hypothesis testing. It fits a function to the data that describes the relationship between one or more independent variables and a dependent one.

Multiple Linear Regression

The equation for Multiple Linear Regression [Fre05] is shown in equation 2.7. The coefficients for the model can be estimated using various ways. The simplest is ordinary least

squares error minimisation. A more advanced way is using gradient descent.

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_{g-1} X_{g-1} + \beta_g X_g + \dots + \beta_{p-1} X_{p-1} \quad (2.7)$$

Multivariate Logistic Regression

The equation for Multivariate Logistic Regression [Fre05] is shown in equation 2.8. The coefficients for the model are estimated using maximum-likelihood estimation.

$$\begin{aligned} \text{logit}(\pi(x)) = \ln \frac{\pi(x)}{1 - \pi(x)} = \\ \beta_0 + \beta_1 X_1 + \dots + \beta_{g-1} X_{g-1} + \beta_g X_g + \dots + \beta_{p-1} X_{p-1} \end{aligned} \quad (2.8)$$

The simplest hypothesis testing using regression is to check whether the slope of a simple linear regression model (shown in equation 2.9) is 0. If there is a relationship between X and Y the slope will not be 0. The null hypothesis in this case is that the slope is equal to zero (shown in equation 2.10).

$$Y = B_0 + B_1 X \quad (2.9)$$

$$\begin{aligned} H_0 : B_1 &= 0 \\ H_1 : B_1 &\neq 0 \end{aligned} \quad (2.10)$$

A more complicated case would be to create a reduced model as shown in equation 2.11 and assume that the rest of the model does not contribute. Therefore $H_0 : \beta_g = \beta_{g+1} = \dots = \beta_{p-1} = \beta_p = 0$

$$k = \beta_0 + \beta_1 X_1 + \dots + \beta_{g-1} X_{g-1} + \beta_{g-1} \quad (2.11)$$

The F - Statistic is used to compare statistical models and it is therefore used in order to compare the two hypothesis as shown in equation 2.12.

$$F = \frac{(SSR_{Full} - SSR_{Reduced})/(p - g)}{MSE} \quad (2.12)$$

where SSR is sum of squares regression (the squared differences between the prediction for each observation and the population mean) and MSE is the mean squared error.

ANOVA

ANOVA [Fie07], is a set of statistical tools for hypothesis testing or statistical significance testing of the difference in means and variances of different groups. ANOVA makes four assumptions:

1. The mean of the response variable is influenced additively and linearly by the factors;
2. The variances of all errors are equal to each other;
3. The errors are independent;
4. The data is normally distributed;

One - Way ANOVA

One - Way ANOVA is used for comparing means between three or more sample groups. Table 2.2 shows the different equations for between groups and within groups for calculating the sum of squares and the mean square error which will later be used for the computation of the F - test. It is checking how the variation is different between the groups and within the groups. n_i is the sample size of a group i , $\bar{x}_i$ is the mean of group i . $\bar{x}$ is the grand mean (mean for all groups combined), k is the total number of groups and n is the sample size. The F - Test is calculated as the ratio of mean square error between groups and mean square error within groups as shown in equation 2.13. It is a statistical test of the variances of different groups(H_0 : the means of the groups are all equal).

Source	Sum Of Squares (SS)	Degrees Of Freedom	Mean Square Error(MSE)
Between Groups	$\sum_j n_j (\bar{x}_j - \bar{x}.)^2$	k - 1	$\frac{\sum_j n_j (\bar{x}_j - \bar{x}.)^2}{k-1}$
Within Groups	$\sum_i \sum_j (x_{ij} - \bar{x}_j)^2$	n-k	$\frac{\sum_i \sum_j (x_{ij} - \bar{x}_j)^2}{n-k}$
Total	$\sum_i \sum_j (x_{ij} - \bar{x}.)^2$	n-1	—

Table 2.2: One - Way ANOVA Test

$$F = \frac{MS_{Between}}{MS_{Within}} \quad (2.13)$$

The F value is then used to calculate the p-value.

Two - Way ANOVA

Two - way ANOVA determines how a response is affected by two variables and how they interact with each other. It is an extension of one - way ANOVA. Assuming a table (T) with number of rows (r) and number of columns (c) two - way ANOVA can be calculated using the information in table 2.3 and table 2.4.

Source	Sums
Sum of c observations in i th row	$T_{R_i} = \sum_j x_{ij}$
Sum of r observations in j th row	$T_{C_j} = \sum_i x_{ij}$
Sum of rc observations	$T = \sum_i \sum_j x_{ij}$

Table 2.3: Two - Way ANOVA Test Observation Sums

Source	Sum Of Squares (SS)	Degrees Of Freedom	Mean Square Error(MS)	F-Test
Factor A (rows)	$SS_R = \sum_i \frac{T_{R_i}^2}{c} - \frac{T^2}{rc}$	r - 1	$MS_A = \frac{SS_R}{r-1}$	$\frac{MS_A}{MS_E}$
Factor B (columns)	$SS_C = \sum_j \frac{T_{C_j}^2}{r} - \frac{T^2}{rc}$	c - 1	$MS_B = \frac{SS_C}{c-1}$	$\frac{MS_B}{MS_E}$
Interaction	$SS_{R,C} = \frac{\sum_i \sum_j (T_{ij}^2)}{n} - \frac{T^2}{rc}$	(r - 1)(c - 1)	$MS_{AB} = \frac{SS_{R,C}}{(r-1)(c-1)}$	$\frac{MS_{AB}}{MS_E}$
Total	$SS_T = \sum_i \sum_j (T_{ij} - \frac{T^2}{rc})^2$	N-1	$MS_T = \frac{SS_{R,C}}{(N-1)}$	—
Sum Of Errors	$SS_E = SS_T - SS_R - SS_C - SS_{R,C}$	N - rc	$MS_E = \frac{SS_E}{(N-rc)}$	—

Table 2.4: Two - Way ANOVA Test

One-Way ANOVA is used when the hypothesis considers population means based on one

characteristic (e.g. one variable is numerical and the other is categorical) while two-way ANOVA considers population means based on multiple characteristics (e.g. two categorical variables and a numerical variable). ANOVA is a fast and robust way of analysing data; however, as described above it makes certain assumptions about the data that need to hold for the analysis to be correct. Specifically it assumes that the variances of the errors (as well as the means and variances for each group) are roughly equal which is hardly the case in real world paradigms. In addition if the null hypothesis is rejected, we only know that at least one group is different from the others but we do not know which. Multiple testing can be employed to find out which it is, and it can be computationally expensive when many groups are considered.

ANOVA is considered to be a more specialised version of regression when the predictors are categorical. Both of the tests use the sum of squares which makes them similar. There are however significant differences. A comparison between ANOVA and Regression is shown in table 2.5.

ANOVA	Regression
Performs a test on the variances of different groups	Fits a function that describes the relation between different variables
Used for statistical significance testing	Can also be used for prediction
Based on one or more categorical predictor variables	Based on one or more continuous predictor variables
Parametric - Assumes a distribution over the data	Does not assume a distribution over the data

Table 2.5: ANOVA vs Regression

Bootstrapping

Bootstrapping [ET93], involves sampling from a greater population of data in order to approximate the properties of an estimator. It is particularly useful when the distribution of

the population is not known. Bootstrap works as shown in algorithm 2.1. Since the distribution is not known the probability of selecting any element is $\frac{1}{n+m}$ for two populations with n and m sample sizes. Bootstrapping draws a number of samples from the dataset and performs a statistical test on it. The process is repeated M times. The number of times the test statistic has a different value than the statistic given the null hypothesis is used to calculate the p-value. Bootstrapping has some important limitations when it comes to samples that are dependent on each other. It fails to capture any such relationships. In addition there could be occasions where big fluctuations in the data distribution can cause bootstrapping to fail (adding or removing data can yield very different results).

Algorithm 2.1. Bootstrapping

 $N = n + m$
for $i = 1..M$ **do**

 Draw a bootstrap sample X^* from N with replacement

 $\hat{\theta}^* = \hat{\theta}(X^*)$
end for

p value $\approx \frac{\#\hat{\theta}^* \geq \hat{\theta} | H_0}{M}$

2.4.2 Conclusion

Statistically identifying significant genes especially using the t-test can be highly affected by the variances in the dataset since variance estimates can be altered by genes that have very low variance. This can cause genes that are not differentially expressed to be identified as differentially expressed [TTC01]. One of the requirements of most of these tests is “homogeneity of variance”, which means that the spread of values for each group should be roughly comparable and the number of samples should also be equal. If this is not the case and the smallest sample group is associated with the largest variance then it tends to dominate the t-test. Outliers can also affect the mean and variance of the dataset, making the chance of rejecting the null hypothesis more difficult. In a dataset where several outliers exist, the sample variance will increase and therefore the t-test statistic will decrease.

Also most of these techniques assume that the data belongs to a normal distribution, which

might not always be the case [GK03]. If the dataset is not large enough, normality cannot be assumed and has to be verified using normality tests. If the number of data points is small, fluctuations from normality will affect the results of the statistical tests while if there are many data points, fluctuations will not matter as much. Reasons for non-normality include:

1. **Extreme Values:** Too many outliers (or extreme values) can cause skewness in the distribution.
2. **Measurement Errors:** Faulty equipment or measurement errors can make the data look non-normal.
3. **Undersampling:** the data represents only a subset of the actual dataset.

Techniques that involve sampling may not always reflect the actual population in terms of diversity and therefore the assumptions drawn might be misleading. Values like the mean, the variance and the quantiles tend to be different between the samples and the actual population. This is called sampling error. [MWLN09].

Most statistical methods are univariate, which means that they consider each gene individually and do not consider the effect of a gene in relation to the presence or absence of others. Multivariate methods measure the relative contribution of a gene to the classification by considering other genes it might interact with. The methods described above are all univariate methods. Some of them can be modified to be multivariate (logistic regression, Multivariate ANOVA). However the problem of choosing a set of features still exists, since without prior information as to which features are more important, all possible combinations have to be checked which can be computationally expensive and inefficient.

Chapter 3

Mathematical Background

3.1 Dimensionality Reduction

3.1.1 Principal Component Analysis

Principal Components Analysis (PCA) [Pea01] can be used to reduce the dimensionality of a dataset. It finds the principal variations among the data. Those vectors are orthogonal (i.e. uncorrelated with each other). The simplification of the data should be done in a way such that the important features are not lost. The PCA algorithm works as follows:

1. Subtract the mean for each data dimension (to get the mean-adjusted data). Mean is given by:

$$\bar{X} = 1/N \sum_{i=1}^N X[i] \quad (3.1)$$

2. Calculate the covariance matrix:

$$C = \frac{1}{n-1} \bar{X}^* \cdot \bar{X} \quad (3.2)$$

where $*$ is the conjugate transpose operator.

3. Calculate the eigenvectors and eigenvalues of the covariance matrix:

4. Order the eigenvectors from highest to lowest and choose the ones with the highest values i.e. the most significant ones. The number of vectors selected will represent the number of dimensions the new dataset will have.
5. Construct a projection matrix which is a matrix of vectors

$$featureVector = (eigenVector_1, eigenVector_2, \dots, eigenVector_n) \quad (3.3)$$

6. Derive the new dataset by transposing and multiplying the feature vector and the mean adjusted data.

$$finalDataset = feature\ Vector^T * Mean\ Adjusted\ Data^T \quad (3.4)$$

This will return the dataset in the axis system defined by the eigenvectors

PCA compares data in terms of similarities and differences.

When computing the principal components (PCs) of a dataset there is no guarantee that the PCs will be related to the class variable. Therefore, supervised principal component analysis (SPCA) was proposed [BGAZJ11, CW09a], which selects the PCs based on the class variables. This extra step was called the gene screening step. SPCA works as follows:

1. Compute the relation measure between each gene with the outcome using linear, logistic, or proportional hazards models.
2. Select genes most associated with the outcome using cross-validation of the models in step 1.
3. Estimate the principal component scores using only the selected genes.
4. Fit regression with the outcome using model in step 1

Even though the supervised version of PCA performs better than the unsupervised, PCA

has an important limitation: it cannot capture non-linear relationships that often exist in the data, especially in complex biological systems.

Kernel PCA [SSM99] is a non-linear variation of PCA. Kernel PCA uses a mapping $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ called the Kernel Trick. This avoids explicitly calculating the coordinates for each pair of data in the new space but it just calculates the inner products instead. Popular kernel functions include:

- Gaussian $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ for $\gamma > 0$
- Polynomial $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$
- Hyperbolic Function $k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x}_j + c)$

3.1.2 Manifold Learning - Isomap

The manifold learning algorithm is used for non-linear dimensionality reduction [Cay05]. Manifold learning works by taking inputs from a higher dimensional space and embedding them to a lower one while preserving their topological characteristics. It assumes that all data points are lying close to or on a manifold and it can be thought as a generalised principal components analysis (PCA) that can capture non-linear relations. Isomap [TdSL00], short for Isometric Mapping, was one of the first approaches to use a manifold and it is an extension of the Kernel PCA algorithm. An example of the Isomap is shown in figure 3.1

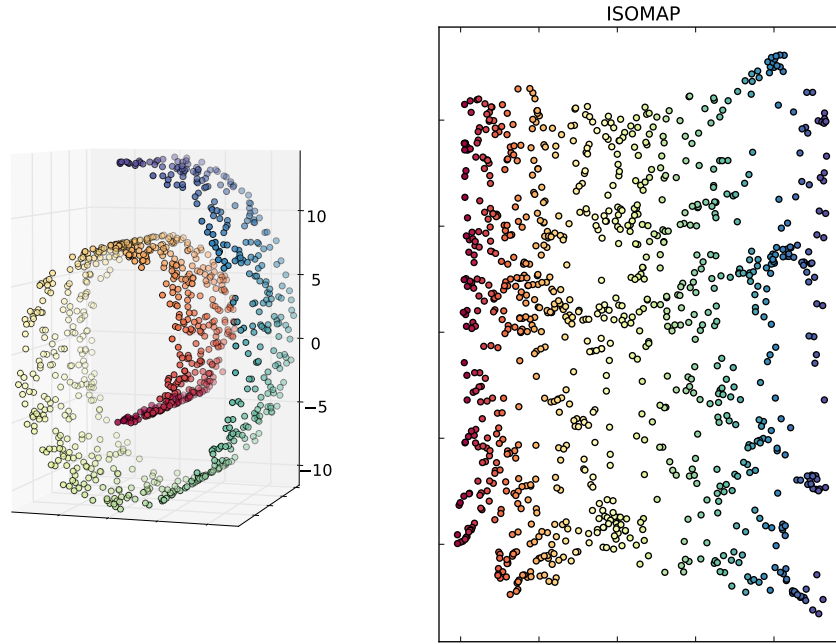


Figure 3.1: **Manifold example with the 3D Swiss roll dataset with $N = 1000$ data points:** The original dataset is illustrated on the left in the original three-dimensional space. The Isomap projection to the two-dimensional space on the right was done, by taking a number of neighbours equal to 8. It can be seen that nearby points in the 2D embedding are also nearby points in the original 3D manifold, as desired [PVG⁺11]

The Isomap algorithm works as follows:

1. Determine the neighbours: For all points in a fixed radius, find the k nearest points (k -Isomap) or the closest points based on distance (ϵ -Isomap)
2. Construct the neighbourhood graph: Points are connected to each other if they are k nearest points away with the edge length set to their Euclidean distance.
3. Find the shortest path between all the nodes on the graph using a graph algorithm (Dijkstra or Floyd-Warshall) to construct the matrix of pairwise geodesic distances between different points.
4. Construct the lower dimensionality mapping. This is the same procedure as classical

multi-dimensional scaling (MDS). Generally another matrix Θ is constructed using:

$$\Theta = -\frac{1}{2}H\Delta^2H \quad (3.5)$$

where Δ is the matrix of geodesic distances;

and H is the centring matrix:

$$H = I_n - \frac{1}{N}U_N \quad (3.6)$$

where U_N is an $N \times N$ matrix of 1's;

and I_n is the identity matrix of size n

5. Calculate the eigenvalues of Θ : Let λ_k be the k^{th} eigenvalue and v_k be the k^{th} eigenvector. The k^{th} component of the embedding Π is constructed by setting it to $\sqrt{\lambda_k}v_k$.

$$\Pi = \begin{pmatrix} \sqrt{\lambda_1}v_1 \\ \sqrt{\lambda_2}v_2 \\ \sqrt{\lambda_3}v_3 \\ \vdots \\ \sqrt{\lambda_d}v_d \end{pmatrix} \quad (3.7)$$

There are several methods for finding manifolds. Locally-Linear Embedding (LLE) [RS00] is similar to Isomap but uses faster optimization techniques for sparse matrices.

LLE works as follows:

1. For each data point $\vec{X}_i$, find the neighbours using either the k -nearest neighbours or by choosing all samples within a fixed distance ϵ .
2. Compute the weights that best reconstruct each data point from its neighbours by

minimising the cost function:

$$E(W) = \sum_i \left| \vec{X}_i - \sum_{j \neq i} \mathbf{w}_{ij} \vec{X}_j \right|^2 \quad (3.8)$$

For a point j find the weights that best describe its contribution to the reconstruction of the point i

3. Each high-dimensional observation $\vec{X}_i$ is mapped to a low-dimensional vector $\vec{Y}_i$ representing the embedding on the manifold by minimising the cost function:

$$C(Y) = \sum_i \left| \vec{Y}_i - \sum_{j \neq i} \mathbf{w}_{ij} \vec{Y}_j \right|^2 \quad (3.9)$$

In summary there are three steps in the LLE algorithm:

1. Construct the neighbourhood graph for each point in the data by using the k -NN algorithm.
2. Calculate the weights that linearly approximate the data in the neighbourhood.
3. Find the low-dimensional coordinates that best reconstruct these weights and return the low-dimensional embedding of the original space.

Other forms of manifold include: Hessian Locally-Linear Embedding (Hessian LLE) [DG03] which is more computationally expensive but uses more sparse matrix techniques for better results. Modified Locally-Linear Embedding (MLLE) [SPH07], another LLE variant that uses a localized weighting system. Laplacian Eigenmaps [BN01] uses spectral clustering to perform dimensionality reduction. Instead of projecting the data, like PCA, and calculating the distance, manifold learning methods use the intrinsic geometry of the data by assuming that the data lie on the low-dimensional manifold.

3.2 Classification

3.2.1 Support Vector Machines

Support Vector Machines are supervised classifiers trained on data. In their simplest form Support Vector Machines use labelled examples to create a model that will be used to allocate new data in a category using a non-probabilistic binary classifier [CV95] defined by a separating hyperplane. Given labelled training data, the algorithm outputs an optimal hyperplane which classifies new examples. Given a labelled training set x_i, y_i where $i = 1, \dots, n$ SVMs can find a solution to:

$$\begin{aligned} \max_w \frac{2}{\|w\|} &= \min_w \|w\|^2 \\ \text{subject to } y_i(w^T x_i + b) &\geq 1 \end{aligned} \quad (3.10)$$

where b is the bias and w is the weight vector.

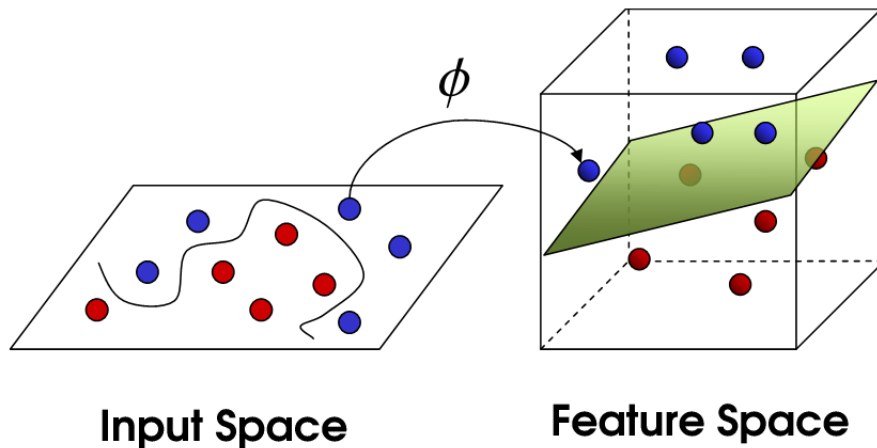


Figure 3.2: **Support Vector Machines:** The kernel function ϕ maps a lower dimensional input to a higher dimensional space in order to create the hyperplane for separating the classes¹

The equation that is maximising the distance between the support vectors is selected and that is why SVMs is considered to be an optimisation problem. The principle of Support

¹Adapted from <http://i.stack.imgur.com/1gvce.png>

Vector Machines is that they take a low dimensional input and map it, using a kernel function, to a higher dimensional space in order to be able to create the hyperplane as shown in figure 3.2. The points lying on the hyperplane are called support vectors. If a kernel is used the equation 3.10 changes to:

$$\begin{aligned} \min_w &= \|w\|^2 \\ \text{subject to } &y_i(w^T \phi(x_i) + b) \geq 1 \end{aligned} \quad (3.11)$$

for $i = 1..N$ and ϕ is the kernel function, w is the weights and b is the bias.

3.2.2 Linear Discriminant Analysis

Linear discriminant analysis (LDA) [Fis36] works by finding a linear combination of features which characterises or separates two or more classes using the Bayes rule:

$$P(y|X) = P(X|y)\dot{P}(y)/P(X) \quad (3.12)$$

The boundary between the two classes (k and l) is defined as:

$$\log[P(y = k|X)/P(y = l|X)] \quad (3.13)$$

which is a log-probability ratio. Linear Discriminant Analysis is similar to Principal Components Analysis since they both use linear combinations of variables to describe the data. Their difference is that LDA takes the class label into consideration while PCA does not.

3.2.3 K - Nearest Neighbours

K -Nearest Neighbours, is a classification algorithm that, based on some distance metric, most commonly Euclidean distance, selects k samples (neighbours) that lie closest to the sample to be classified and assigns to it the label of the majority of its neighbours [Die00].

3.3 Ensemble Methods

To improve the quality of the output, instead of using only one machine learning method, a combination can be used. These are the ensemble methods.

A classifier is a hypothesis about the true function in respect to the data. Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a vote on their predictions.

Each classifier is an individual hypothesis about the dataset. For ensemble methods to work as accurately as possible they need to be diverse. Diversity means that they make different errors in the newly classified data [Die00]. Generally ensemble classification works by:

- Train different classifiers on the training dataset
- Voting done in two ways:
 1. Weighted: Different classifiers have different voting power
 2. Un-Weighted: All classifiers have the same voting power

Ensemble methods tend to work fairly well since they combine several methods and they work with majority votes in order to decide on the output. Nevertheless using more than one classifier can be time consuming especially when the dataset is as large and complex as microarray data. Common types of ensembles are:

- Boosting: Boosting combines a number of weak classifiers and then takes a majority vote on their result. It focuses on misclassified samples by training more classifiers on the instances the previous classifiers failed to label correctly as shown in figure 3.3.

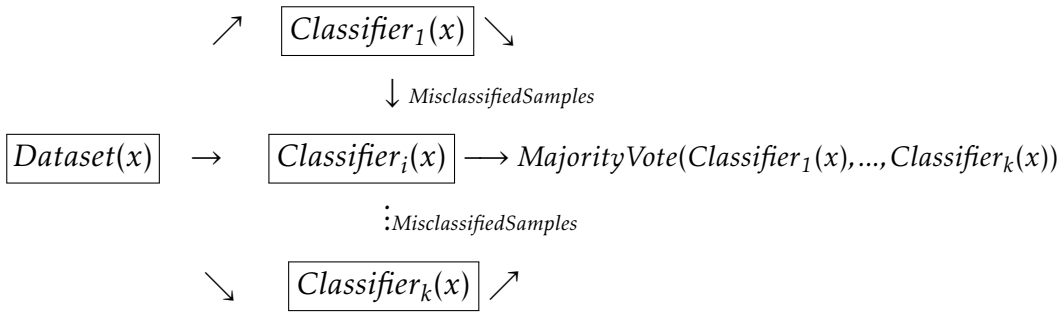


Figure 3.3: The Boosting Algorithm

- Bootstrap Aggregation (Bagging): A number of classifiers are trained using different subsets of the dataset and then a majority vote is taken as shown in figure 3.4.

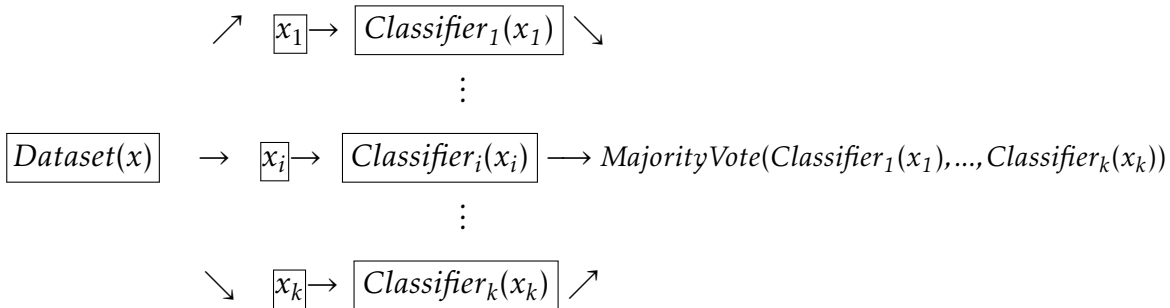


Figure 3.4: The Bagging Algorithm

- Stacked generalization (Stacking): A number of classifiers are trained using the existing dataset and then a combiner algorithm is trained on the classifiers to make the final prediction. The combiner algorithm can be the same as either bagging or boosting; however, it is most frequently just a regression model. The algorithm is shown in figure 3.5.

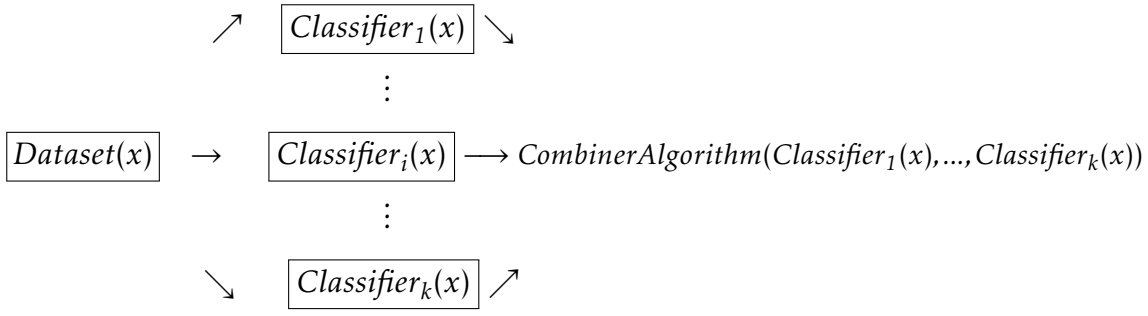


Figure 3.5: The Stacking Algorithm

3.3.1 AdaBoost With Decision Trees As The Weak Classifiers

Adaboost is an ensemble method for classifying that can improve the quality of the output. AdaBoost starts with one classifier fitted to the dataset and then trains different versions of it that are again applied on the data. The classifier's weights are adjusted according to the accuracy of the result and normally on subsequent runs of the program they are modified so that they can accommodate the most difficult cases [FS97]. The algorithm used in this thesis is the AdaBoost-SAMME algorithm [HRZZ09], a multi-class version of the original algorithm. Decision trees were used as the weak classifiers. They take as inputs tuples of the form $(X, Y) = (x_1, x_2, \dots, x_k, Y)$ and they create rules based on $(x_1, x_2, \dots, x_k)$ so that the target Y can be classified correctly. The tree is constructed by splitting the inputs recursively (recursive partitioning) and it ends when the subset at a node has items with the same label or when the accuracy can no longer be improved using the Gini Impurity shown in equation 3.14. The Gini Impurity measures how often a random element can be labelled incorrectly if a random label was assigned to it based on the distribution of labels on the set.

$$I_G(f) = \sum_{i=1}^m f_i(1 - f_i) = \sum_{i=1}^m (f_i - f_i^2) = \sum_{i=1}^m f_i - \sum_{i=1}^m f_i^2 = 1 - \sum_{i=1}^m f_i^2 \quad (3.14)$$

where m is the set of values for which the Gini Impurity is calculated and f is the proportion

of items labelled with the value i in the set. The decision tree algorithm used is the Classification And Regression Tree (CART) [BFOS84] which works with both categorical and numerical target variables.

3.4 Internal Cluster Evaluation

3.4.1 Dunn Index

The Dunn Index is an internal evaluation metric for clusters [Dun73]. Internal evaluation means that it only depends on the data of the cluster itself, mainly by considering the clusters with little variance to be better. It finds the smallest distance between samples in different clusters over the maximum distance of samples in the same cluster, and it is defined as:

$$DI_m = \min_{1 \leq i \leq m} \left\{ \min_{1 \leq j \leq m, j \neq i} \left\{ \frac{\delta(C_i, C_j)}{\max_{1 \leq k \leq m} \Delta_k} \right\} \right\} \quad (3.15)$$

where δ is the distance metric between the cluster C_i and C_j which can be: single, complete, average, centroid or Hausdorff [RW98].

and Δ is

$$\Delta_i = \frac{\sum_{x \in C_i} d(x, \mu)}{|C_i|}, \mu = \frac{\sum_{x \in C_i} x}{|C_i|} \quad (3.16)$$

3.4.2 Davies–Bouldin index

The Davies-Bouldin index is another internal evaluation metric [DB79]. It is given by the equation:

$$DB \equiv \frac{1}{N} \sum_{i=1}^N D_i \quad (3.17)$$

where D_i is the symmetry condition:

$$D_i \equiv \max_{j:i \neq j} R_{i,j} \quad (3.18)$$

and R_{ij} is a measure of how good the clustering scheme is:

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}} \quad (3.19)$$

where S is the measure of scatter inside cluster i and M is the measure of separation between clusters i and j : $M_{i,j} = \|A_i - A_j\|$ where A denotes the centroids of the clusters i and j . The Davies - Bouldin index measures the average similarity between each cluster and its most similar one, averaged over all the clusters. No cluster should be similar to another one and therefore a good clustering scheme should have a low Davies-Bouldin score.

3.4.3 Silhouette

Silhouette [Rou87] is given by:

$$s_i = \frac{b_i - a_i}{\max a_i, b_i} \quad (3.20)$$

where a_i is the average distance of data point i from all other data points within the same cluster. The closest distance of i from any other cluster is denoted as $b(i)$. Silhouette first calculates the dissimilarity of the cluster which is the average distance between all cluster members. Cluster members with low dissimilarity are comfortably within the cluster to which they have been assigned. The average dissimilarity for a cluster is a measure of how compact it is. The average distance to fellow cluster members is then compared to the average distance to members of the neighbouring cluster. The silhouette of a point is defined as the ratio between its dissimilarity to its own cluster and its dissimilarity to its nearest neighbouring cluster. The silhouette metric can range between -1 and 1. A value of -1 means that the cluster member is more similar to the members of its neighbouring cluster and a value

of 1 shows that it is identical to the members of its own cluster.

3.5 Cross-Validation

Cross-Validation [Gei93] is a statistical method for calculating the accuracy of a model by assessing how well it generalises over an independent data set. It partitions the data in two groups, the training group and the validation group. The model is created using the training group and validated over the validation group. To reduce the variance this process takes place in more than one round using different groups for training and validation and the results are averaged over the rounds. K-fold-cross-validation using $k = 10$, is the most common form [MDA05] since as the k increases the overlap between the training and the test dataset also increases. A smaller k means that the size of the test set is also smaller which can lead to less precision. If k is too large the training set size is closer to the full data size which is not good for training a more generalised algorithm. $k = 10$ is a good compromise between a more generalised training and good test sizes. The data is partitioned in ten groups where nine of them are used as training data and the other one as validation. This is repeated ten times so that all the groups will be the validation group exactly once. The results are averaged to produce a single estimation. It is considered to be a good method because all observations are used for both training and validation. The algorithm is shown in algorithm 3.1.

3.5.1 Stratified Cross-Validation

In stratified k-fold cross-validation, the training and validation groups are selected so that the mean response value is approximately equal in all the folds. Since the classification is binary this means that each fold contains the same number of the two types of class labels (if the classification is binary).

Algorithm 3.1. k -Fold Cross Validation

Shuffle the training samples (m)

Divide the training samples in k folds ($\approx m/k$)

for $i = 1..k$ **do**

 Train the classifier with the examples that do not belong in fold i

 Test the classifier on fold i

 Compute n_i , the number of examples in fold i that were wrongly classified

end for

Return the classifier error: $E = \frac{\sum_{i=1}^k n_i}{m}$

3.5.2 Receiver Operating Characteristic Curves

Receiver operating characteristic curves (ROC curves) are plots that are used for illustrating the performance of a binary classifier. In a two class classifier there are four possible outcomes:

1. True Positive: if the outcome from a prediction is p and the actual value is also p;
2. False Positive: if the outcome from a prediction is p however the actual value is n;
3. True Negative: if the outcome from a prediction is n and the actual value is also n;
4. False Negative: if the outcome from a prediction is n however the actual value is p;

The curve is created by plotting the true positive rate against the false positive rate at various threshold settings. The true-positive rate is also known as sensitivity or recall. The false-positive rate is also known as the fall-out [Faw06].

Chapter 4

Dimensionality Reduction For High-Dimensional Microarray Datasets

4.1 Introduction

Machine learning is concerned with programming systems that can learn from the data provided and improve their performance with experience. Learning involves generalising a behaviour so that when a new situation comes it can be identified and dealt with. To do that the system is designed so that it can detect patterns or rules in the data by making a model and adjusting its performance accordingly. A way to select the most suitable rule model is by utilising the Occam's Razor or the Principle of Parsimony [Tho15] which states that:

One should not increase, beyond what is necessary, the number of entities required to explain anything.

and it goes back to Aristotle, who wrote:

Nature operates in the shortest way possible.

Machine learning can be particularly useful when dealing with large amounts of data and

complex problems that are not easy for the human brain to solve. It normally proceeds in two steps:

1. **Training:** Giving the system enough data to learn from.
2. **Classification:** Running the learned model on a dataset to classify the data.

Machine learning techniques can be separated into three categories:

- **Supervised:** which uses only labelled data;
- **Unsupervised:** which uses unlabelled data;
- **Semi-Supervised:** which uses labelled and unlabelled data.

In some cases prior knowledge is used to simplify this process. Prior knowledge is all information that is available to the system in addition to the training data [LS02]. This information is integrated with the training data to get better models of the dataset.

4.2 The Curse Of Dimensionality

In machine learning as the dimensionality of the data rises, the amount of data required to provide a reliable analysis grows exponentially. Richard E. Bellman referred to this phenomenon as the “curse of dimensionality” when considering problems in dynamic optimisation [Bel57]. A popular approach to the problem of high-dimensional datasets is to search for a projection of the data onto a smaller number of variables (or features) which preserves the information as much as possible. Microarray data is typical of this type of small sample problem. Each data point (sample) can have up to 400,000 variables (gene probes) and processing a large number of data points involves high computational cost [KM09]. When the dimensionality of a dataset grows significantly there is an increasing difficulty in proving the result statistically significant due to the sparsity of the meaningful data in the dataset

in question. Large datasets with the so called “large p , small n ” problem (where p is the number of features and n is the number of samples) tend to be prone to over-fitting. An over-fitted model can mistake small fluctuations for important variance in the data which can lead to classification errors. This difficulty can also increase due to noisy features. Noise in a dataset is defined as “*the error in the variance of a measured variable*” which can result from errors in measurements or natural variation [Han05]. Machine learning algorithms tend to be affected by noisy data. Noise should be reduced as much as possible in order to avoid unnecessary complexity in the inferred models and improve the efficiency of the algorithm [SLW97]. Common noise can be divided into two types [ZW04]:

1. Attribute Noise
2. Class Noise

Attribute noise is caused by errors in the attribute values (wrongly measured variables, missing values) while class noise is caused by samples that are labelled to belong in more than one class and/or misclassifications.

As the dimensionality increases the computational cost also increases, usually exponentially. To overcome this problem it is necessary to find a way to reduce the number of features under consideration. Two techniques are often used:

1. Feature Subset Selection
2. Feature Extraction

4.3 Feature Subset Selection In Microarray Cancer Data

Feature Subset Selection works by removing features that are not relevant or are redundant. The subset of features selected should follow the Occam’s Razor principle and also give the best performance according to some objective function. In many cases this is an NP-hard

(Non-deterministic Polynomial-time hard) problem [Han89, BR92]. The size of the data to be processed has increased rapidly the past 5 years and therefore feature selection has become a requirement before any kind of classification takes place. Unlike feature extraction methods, feature selection techniques do not alter the original representation of the data [SILn07]. One objective for both Feature Subset Selection and Feature Extraction methods is to avoid over-fitting the data in order to make further analysis possible. The simplest is feature selection, in which the number of gene probes in an experiment is reduced by selecting only the most significant ones according to some criterion such as high levels of activity. Feature selection algorithms are separated into three categories [Das01, BL97]:

- The *Filters*: which extract features from the data without any learning involved.
- The *Wrappers*: that use learning techniques to evaluate which features are useful.
- The *Embedded Techniques*: which combine the feature selection step and the classifier construction.

4.3.1 Filters

Filters work without taking the classifier into consideration. This makes them very computationally efficient. They are divided into multivariate and univariate methods. Multivariate methods are able to find relationships among the features, while univariate methods consider each feature separately. Gene ranking is a popular statistical method for microarray data. The following methods were proposed in order to rank the genes in a dataset based on their significance [XJK01]:

- (Univariate) Unconditional Mixture Modelling: assumes two different states of the gene *on* and *off* and checks whether the underlying binary state of the gene affects the classification using mixture overlap probability;
- (Univariate) Information Gain Ranking: approximates the conditional distribution $P(C|F)$ where C is the class label and F is the feature vector. Information gain is used

as a surrogate for the conditional distribution;

- (Multivariate) Markov Blanket Filtering: finds features that are independent of the class label so that removing them will not affect the accuracy.

Methods based on correlation have also been suggested:

- (Multivariate) Error-Weighted Uncorrelated Shrunk Centroid (EWUSC): This method is based on the uncorrelated shrunk centroid (USC) and shrunk centroid (SC). The shrunk centroid is found by dividing the average gene expression for each gene in each class by the standard deviation for that gene in that class. This way higher weight is given to genes whose expression is the same among different samples in the same class. New samples are assigned to the label with the nearest average pattern (using squared distance). The uncorrelated shrunk centroid approach removes redundant features by finding genes that are highly correlated in the set of genes already found by SC. The EWUSC uses both of these steps and in addition adds error-weights (based on within-class variability) so that noisy genes will be downgraded and redundant genes are removed [YB05]. A comparison is shown in figure 4.1 where the three different methods are tested on a relatively small (25000 genes, 78 samples) breast cancer dataset. The algorithms perform well when the number of relevant genes is less than 1000 and generally EWUSC and USC perform better than SC (fewer genes are required for an accurate model).

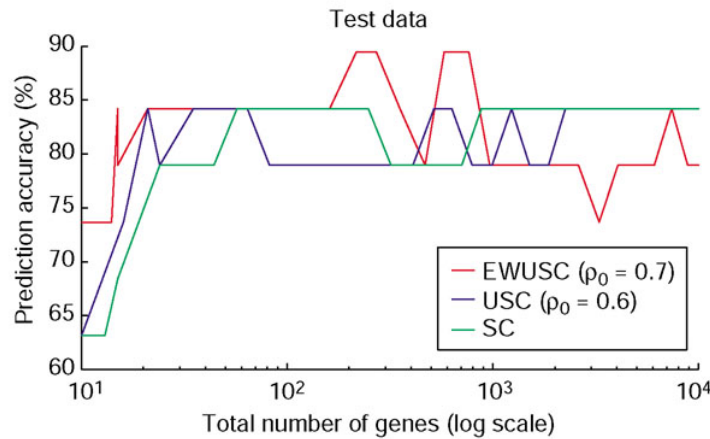


Figure 4.1: Comparison between EWUSC, USC and SC on breast cancer data [YB05]

- (Multivariate) Minimum Redundancy - Maximum Relevance (mRMR): mRMR is a method that maximises the relevancy of genes within the class label while minimising the redundancy in each class. To do so it uses several statistical measures. Mutual Information (MI) measures the information a random variable can give about another, in particular the gene activity and the class label. This method can be applied to both categorical and continuous variables. For Categorical (Discrete) Variables, MI is used to find genes that are not redundant (minimise W) and are maximally relevant (maximise V) with a target label, as shown in equations 4.1 and 4.2 respectively [DP03],

$$W = \frac{1}{|S|^2} \sum_{i,j \in S} I(i,j) \quad (4.1)$$

$$V = \frac{1}{|S|} \sum_{i \in S} I(h,i) \quad (4.2)$$

where I is the MI, i and j are genes, $|S|$ is the number of features in S and h is a class label.

Formally, the mutual information of two discrete random variables X and Y can be defined as:

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right), \quad (4.3)$$

where $p(x, y)$ is the joint probability distribution function of X and Y , and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively.

For Continuous Variables the F-statistic, that checks whether the means of two populations are significantly different, is used to find the maximum relevance between a gene and a class label and then the correlation of the gene pairs in that class is measured to minimise redundancy as shown in equations 4.4 and 4.5 respectively [DP03],

$$V = \frac{1}{|S|} \sum_{i \in S} F(i, h) \quad (4.4)$$

$$W = \frac{1}{|S|^2} \sum_{i,j \in S} |c(i,j)| \quad (4.5)$$

where F is the F-statistic, i and j are genes, h is a class label, $|S|$ is the number of features in S and c is the correlation.

Normalised mutual information can also be used to measure the relevance and redundancy of clusters of genes. The most relevant genes are combined and Leave-One-Out Cross Validation (LOOCV) is performed to find the accuracy [LKM05]. For continuous variables linear relationships are used instead of mutual information. MRMR methods give lower error accuracies for both categorical and discrete data.

- (Multivariate) Correlation-based feature selection (CFS) as stated by Hall [Hal00] follows the principle that “a good feature subset is one that contains features highly correlated with the class, yet uncorrelated with each other”. CFS evaluates a subset by considering the predictive ability of each one of its features individually and also their degree of redundancy (or correlation). The difference between CFS and other methods is that it provides a heuristic merit for a feature subset instead of each feature independently [WTH⁺05]. This means that given a function (heuristic), the algorithm can decide on its next move by selecting the option that maximises the output of this function. Heuristic functions can also be designed to minimise the cost of reaching the goal. CFS was used in combination with wrappers (Decision Trees (C4.5), Naive Bayes) and a number of filters (Relief-F, X^2 , Information Gain, Information Gain Ratio). Both filters and wrappers resulted in similar accuracy even though using filters is a much faster approach.

Relief-F [HS98] is also used with cancer microarray data. It is a multivariate method that chooses the features that are the most distinguishable among the different classes. It repeatedly draws an instance (sample) and based on its neighbours, gives more weight to the features that help discriminate it from the neighbours of a different class [MBM⁺04, WM04]. A comparison between Relief-F, Information Gain, Information Gain Ratio and X^2 is shown in figure 4.2. The methods perform similarly across the number of genes selected. Informa-

tion Gain Ratio is defined as the information gain over the intrinsic information. It performs normalisation to the information gain using split value information (entropy of the distribution of the instances). The Pearson X^2 test evaluates how likely it is that a difference between the sets has occurred by chance.

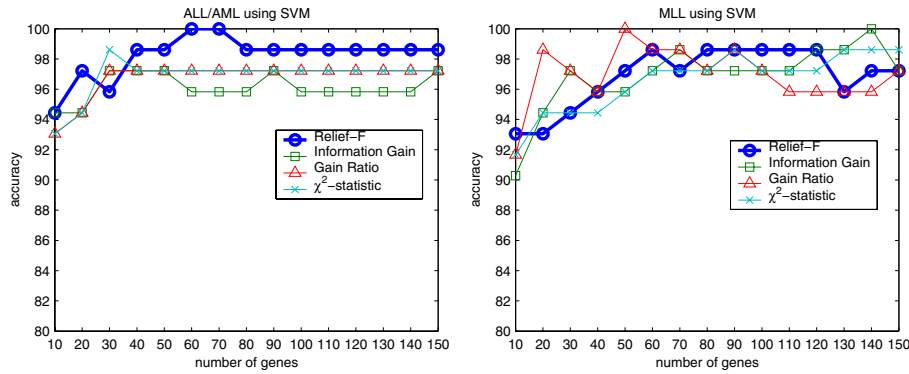


Figure 4.2: **Comparison between Relief-F, Information Gain, Information Gain Ratio and X^2 test on ALL and MLL Leukaemia datasets [WM04]**

A method using independent logistic regression with two steps was also proposed [WVOM04]. The first step is an univariate method in which the genes are ranked according to their Pearson correlation coefficients. The top genes are considered in the second phase, which is stepwise variable selection. This is a conditionally univariate method based on the inclusion (or exclusion) of a single gene at a time, conditioned on the variables already included.

Even though all these methods can be highly accurate in classifying information there is no biological significance proven for the genes that are identified by them. None of the above methods have indicated whether the results are actually biologically relevant or not. In addition filter methods are generally faster than wrappers but do not take into account the classifier which can be a disadvantage. Ignoring the specific heuristics and biases of the classifier might lower the classification accuracy.

4.3.2 Wrappers

Wrappers tend to perform better in selecting features since they take the model hypothesis into account by training and testing in the feature space. This leads to the big disadvantage

of wrappers, the computational inefficiency which is more apparent as the feature space grows. Unlike filters, they can detect feature dependencies. Wrappers are separated into two categories: Randomised and Deterministic. A comparison is shown in table 4.1.

Deterministic	Randomised
Small Over-fitting Risk	High Over-fitting Risk
Prone to Local Optima	Less Prone to Local Optima
Classifier Dependent	Classifier Dependent
Computationally Intensive	Computationally Intensive

Table 4.1: **Deterministic vs Randomised Wrappers**

Deterministic Wrappers

A number of deterministic wrappers¹ have been used to examine breast cancer including a combination of a wrapper and sequential forward selection (SFS). SFS is a deterministic feature selection method that works by using hill-climbing search to add all possible single-attribute expansions to the current subset and evaluate them. It starts from an empty subset of genes and sequentially selects genes, one at a time, until no further improvement is achieved in the evaluation function. The feature that leads to the best score is added permanently. The wrappers used were Naive Bayes (NB) and C4.5 in combination with cross-validation [GFHK09]. For classification, support vector machines (SVM), k -nearest neighbours and probabilistic neural networks were used in an attempt to classify between cancerous and non-cancerous breast tumours [OS10]. Very accurate results were achieved using SVMs. Methods based on SVMs are very popular when analysing microarray cancer datasets:

1. Gradient-based-leave-one-out gene selection (GLGS) [CVBM02, LSC⁺09, TSY06, XXL13], was originally introduced for selecting parameters for the SVMs. It starts by applying PCA on the dataset. A vector with scaling factors of the new low dimensional space is

¹Running them several times will give the same result

calculated and optimised using a gradient based algorithm and the leave one out error. Genes are sequentially selected based on a correlation factor.

2. Leave-one-out calculation sequential forward selection (LOOCSFS) is a feature selection method for cancer data based on Sequential Forward Selection (SFS). It adds features in an initially empty set and calculates the leave-one-out cross-validation error [AM02]. It is an almost unbiased estimator of the generalisation error using SVMs and C Bound. C Bound is the decision boundary and it is used as a supplementary criterion in the case where different features subsets have the same leave-one-out cross-validation error (LOOCVE) [LSC⁺09, GWBV02, LSC⁺11]. SFS can also add constraints [GFHK09] on the size of the subset to be selected. It can be used in combination with the recursive support vector machine (R-SVM) algorithm [JUA05]. Several subsets of the dataset with different sizes are considered. The R-SVM algorithm calculates the contribution factor, based on minimal error of the support vector machine, for each gene and the top ranked genes are chosen for the subset.

LOOCSFS is expected to be an accurate estimator of the generalization error while GLGS scales very well with high dimensional datasets. The number of the genes in the feature subset for both LOOCSFS and GLGS has to be given in advance which can be a disadvantage since the most important genes are not known in advance. GLGS is said to perform better than LOOCSFS [TSY06, LSC⁺09].

Randomised Wrappers

Most randomised wrappers² use Genetic algorithms (GA) (algorithm 4.1) and simulated annealing (algorithm 4.2).

Linear discriminant analysis was used in combination with genetic algorithms. Subsets of genes are used as chromosomes and the best 10% of each generation is merged with the previous ones. Part of the chromosome is the discriminant coefficient which indicates the im-

²Have an element of randomness in them so running them will give a slightly different result every time

Algorithm 4.1. Genetic Algorithms

```

Population = Encode(Dataset)
Fitness = f(Population)
while Fitness < Target do
    for 1 to Rand() do
        Populationi = Population(Rand())
        Populationi = Crossover OR Mutation
    end for
    Fitness = f(Population)
end while

```

portance of a gene for a class label [HDH08]. Genetic Algorithms - Support Vector Machines (GA-SVM) [PM12] create a population of chromosomes as binary strings that represent the subset of features that are evaluated using SVMs.

Simulated annealing is another wrapper technique that works by assuming that some parts of the current solution belong to a better one, and therefore proceeds to explore the neighbours seeking out solutions that minimise the objective function and therefore avoiding local optima. Hybrid methods with simulated annealing and genetic algorithms have also been used [RB12]. A genetic algorithm is run as a first step before the simulated annealing in order to get the fittest individuals as inputs to the simulated annealing algorithm. Each solution is evaluated using Fuzzy C-Means³. The problem with genetic algorithms is that the time complexity becomes $O(n \log(n) + nmpg)$ where n is the number of samples, m is the dimension of the data sets, p represents the population size and g is the number of generations. In order for the algorithm to be effective the number of generations and the population size must be quite large. In addition like all wrappers, randomised algorithms take up more CPU time and more memory to run.

Best Incremental Ranked Subset (BIRS) [RRAR06], is an algorithm that scores genes based on their value (using some evaluation measure, in this case Markov Blanket) and class label and then uses a classifier to identify the most accurate subset of genes from the ones previously identified. The algorithm is shown in algorithm 4.3.

³A clustering algorithm that uses coefficients to describe how relevant a feature is to a cluster [Dun73, Bez81].

Algorithm 4.2. Simulated Annealing Algorithm

```

Initialise State  $s = S(0)$ 
Initialise Energy  $e = E(S(0))$ 
Set time to zero  $k = 0$ 
while  $k < k_{\max}$  And  $e < e_{\max}$  do
    Temperature = temperature( $k/k_{\max}$ )
    NewState = neighbour( $s$ )
    NewEnergy =  $E(\text{NewState})$ 
    if AcceptanceProbability( $e$ , NewEnergy, Temperature) > random() then
         $s = \text{NewState}$ 
         $e = \text{NewEnergy}$ 
    end if
    if NewEnergy < EnergyBest then
        BestState = NewState
        EnergyBest = NewEnergy
    end if
     $k = k + 1$ 
end while

```

Algorithm 4.3. Best Incremental Ranked Subset (BIRS)

```

 $R = \emptyset$ 
for gene  $g_i \in \text{Dataset}$  do
    Score = rank( $g_i$ , Dataset)
    R.add(score)
end for
Sort(R)
BestClass = 0
BestSubset =  $\emptyset$ 
for  $i = 1..N$  do
    TempSubset = BestSubset  $\cup g_i (g_i \in R)$ 
    TempClassifier = WrapperClassifier(BestSubset)
    if TempClassifier > BestClass then
        BestSubset = TempSubset
        BestClass = TempClassifier
    end if
end for

```

4.3.3 Embedded Techniques

Embedded Techniques tend to perform better computationally than wrappers but they make classifier-dependent selections that might not work with any other classifier. That is because the optimal set of genes is built when the classifier is constructed and the selection is affected by the hypotheses the classifier makes. A well-known embedded technique is random forests. A random forest is a collection of classifiers. New random forests are created iteratively by discarding a small fraction of the genes that have the lowest importance [DUdA06]. The forest with the smallest amount of features and the lowest error is selected to be the feature subset. In addition SVMs can be used for both feature selection and classification. Features that do not contribute to classification are eliminated in each round until no further improvement in the classification can be achieved [MWB11]. This is called support vector machines with recursive feature elimination (SVM - RFE). It starts with all the features and gradually excludes the ones that do not identify separating samples in different classes. A feature is considered useful based on its weight resulting from training SVMs with the current set of features. In order to increase the likelihood that only the “best” features are selected, feature elimination progresses gradually and includes cross-validation steps [ZLS⁺06, TZH07, LSC⁺09, TSY05]. A major advantage of SVM-RFE is that it can select high-quality feature subsets for a particular classifier. It is however computationally expensive since it goes through all features one by one and it does not take into account any correlation the features might have [GWBV02]. SVM - RFE was compared against two wrappers: leave-one-out calculation sequential forward selection and gradient-based-leave-one-out [TSY06]. All three of these methods have similar computational times when run against a Hepatocellular Carcinoma dataset (7129 genes, 60 samples). GLGS outperforms the others, with LOOCSFS and SVM - RFE having similar performance errors [TSY06]. A comparison of the computational complexity of the algorithms is summarised in figure 4.3.

The most commonly used methods for microarray data analysis are shown in table 4.2.

Method	Type	Supervised	Linear	Description
<i>t</i> -test feature selection [JA06]	Filter	–	Yes	It finds features with a maximal difference of mean value between groups and a minimal variability within each group
Correlation-based feature selection (CFS) [Hal98]	Filter	–	Yes	Finds features that are highly correlated with the class label but are uncorrelated with each other
Bayesian Networks [RJFD10, JHE04]	Filter	Yes	No	Determines the causal relationships among features and removes the ones that do not have any causal relationship with the class
Information Gain (IG) [YZZZ10]	Filter	Yes	Yes	Measures how common a feature is in a class compared to all other classes
Genetic Algorithms (GA) [JUA05, OT03]	Wrapper	Yes	No	Finds the smaller set of features for which the optimization criterion (classification accuracy) does not deteriorate
Sequential Search [GC65]	Wrapper	–	–	Heuristic Base search algorithm that finds the features with the highest criterion value (classification accuracy) by adding one new feature to the set every time
SVM method of Recursive Feature Elimination (RFE) [GWBV02]	Embedded	Yes	No	Constructs the SVM classifier and eliminates the features based on their “weight” when constructing the classifier
Random Forests [DUdA06, IDC ⁺ 04]	Embedded	Yes	No	Creates a number of decision trees using different samples of the original data and uses different averaging algorithms to improve accuracy
Least Absolute Shrinkage and Selection Operator (LASSO) [MMS ⁺ 07]	Embedded	Yes	Yes	Constructs a linear model that “shrinks” many of the coefficients to zero and uses the non-zero ones as the selected features

Table 4.2: Feature Selection methods applied on microarray data

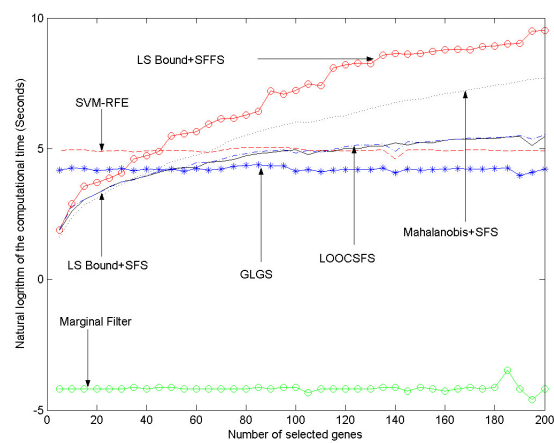


Figure 4.3: **Computational Complexity Comparison between SVM - RFE, LOOCSFS and GLGS [TSY06]**

4.4 Feature Extraction In Microarray Cancer Data

Early applications of machine learning to microarray data used simple clustering methods [ESBB98]. A widely used method was hierarchical clustering. Due to the flexibility of the clustering methods they became very popular among biologists. Bi-clustering followed hierarchical clustering as a way of simultaneously clustering both the samples and the features of a dataset leading to more meaningful clusters. It was shown that bi-clustering performs better than hierarchical clustering when it comes to microarray data but it is still a computationally demanding method [PBZ⁺06]. Many other methods have been implemented for extracting only the important information from the microarrays thus reducing their size. Feature Extraction creates new variables as combinations of others to reduce the dimensionality of the selected features. There are two broad categories for feature extraction algorithms: linear and non-linear.

4.4.1 Linear

Linear feature extraction assumes that the data lies on a lower dimensional linear subspace. It projects it on this subspace using matrix factorization. Given a dataset $X: N \times D$, there exists a projection matrix $U: D \times K$ and a projection $Z: N \times K$, where $Z = X \cdot U$. Using $UU^T = I$ (orthogonal property of eigenvectors), gives $X = Z \cdot U^T$. A graphical representation is shown in figure 4.4.

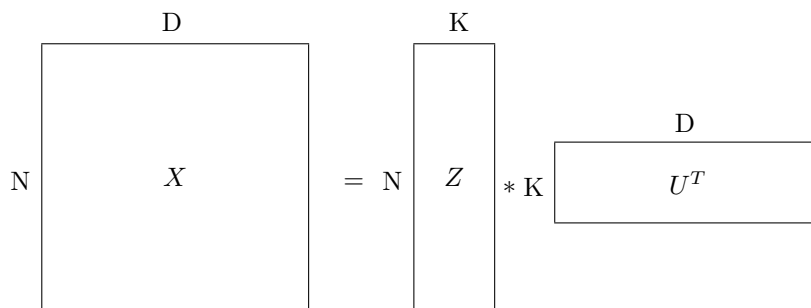


Figure 4.4: **Dimensionality reduction using linear matrix factorization:** Projecting the data onto a lower dimensional linear subspace

PCA, that was discussed in chapter 3, and many variations of it have been applied as a way of reducing the dimensionality of cancer microarray data [JS08, LWW02, EBES05, NM09, MSH⁺02, RSA00, WG05]. SPCA was also used on microarray datasets [BT04, BPT06]. The method was highly effective in identifying important genes and in cross validation tests was only outperformed by gene shaving, a statistical method for clustering similar to hierarchical clustering. The main difference is that the genes can be part of more than one cluster. The term “shaving” comes from the removal or shaving of a percentage of the genes (normally 10%) that have the smallest absolute inner-product with the leading principal component [HTE⁺00].

A similar linear approach is classical multidimensional scaling (classical MDS) or Principal Coordinates Analysis [BG05] which calculates the matrix of dissimilarities for any given matrix input. It was used for large genomic datasets because it is efficient in combination with Vector Quantization or K -Means [TLL08] which assigns each observation to a class, out of a total of K classes [HW79].

4.4.2 Non-Linear

Non-linear dimensionality reduction works in a different way. For example, a low dimensional surface can be mapped onto a high dimensional space so that a non-linear relationship among the features can be found. In theory, a lifting function $f(x)$ can be used to map the features onto a higher dimensional space. On a higher space the relationship among the features can be viewed as linear and therefore is easily detected. This is then mapped back on the lower dimensional space and the relationship can be viewed as non-linear. In practice kernel functions can be designed to create the same effect without the need to explicitly compute the lifting function. Another approach to non-linear dimensionality reduction is by using manifolds. It is based on the assumption that the data (genes of interest) lie on an embedded non-linear manifold which has lower dimension than the raw data space and lies within it. Several algorithms exist working in the manifold space and applied to microarrays. A commonly used method of finding an appropriate manifold, Isomap [TdSL00],

which was discussed in chapter 3, constructs the manifold by joining each point only to its nearest neighbours. Distances between points are then taken as geodesic distances on the resulting graph. Many variants of Isomap have been used, for example Balasubramanian proposed a tree connected version which differs in the way the neighbourhood graph is constructed [BS02]. The k -nearest points are found by constructing a minimum spanning tree using an ϵ -radius hypersphere. This method aims to overcome the drawbacks expressed by Orsenigo and Vercellis [OV12] regarding the robustness of the Isomap algorithm when it comes to noise and outliers. These could cause potential problems with the neighbouring graph, especially when the graph is not fully connected. Isomap has been applied on microarray data with some very good results [DRM05, OV12]. Compared to PCA, Isomap was able to extract more structural information about the data. In addition, other manifold algorithms have been used with microarray data such as Locally-Linear Embedding (LLE) [CL04] and Laplacian Eigenmaps [ERZ⁺11, KSO02]. PCA and similar manifold methods are also used for data visualisation as shown in figure 4.5. Clusters can often be better separated using manifold LLE and Isomap but PCA is a lot faster than the other two.

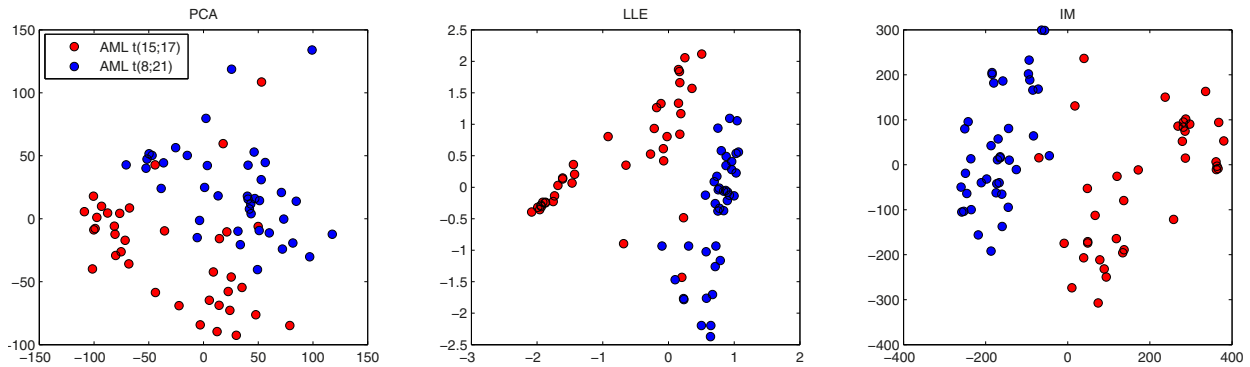


Figure 4.5: **Visualisation of a Leukaemia dataset with PCA, manifold LLE and manifold Isomap** [BKR⁺10]

Another non-linear method for classification is Kernel PCA. It has been regularly used [RVO14, LCB05] since dimensionality reduction helps with the interpretability of the results. It does have an important limitation in terms of space complexity since it stores all the dot products of the training set and therefore the size of the matrix increases quadratically with the number of data points [LY09].

Neural methods, like Self Organizing Maps (SOMs) are also used for dimensionality reduction [Koh88]. SOMs or Kohonen maps create a lower dimensional mapping of an input by preserving its topological characteristics. They are composed of nodes or neurons and each node is associated with its own weight vector. SOMs training is considered to be “competitive” since when a training example is fed to the network its Euclidean distance with all nodes is calculated and it is assigned to the node with the smallest distance (Best Matching Unit (BMU)). The weight of that node along with its neighbouring nodes is adjusted to match the input. SOMs have been used as a method of dimensionality reduction for gene expression data [KNT⁺01, KSO02] but they were never broadly adopted for analysis because they need just the right amount of data to perform well. Insufficient or extraneous data can cause randomness to the clusters. Another neural network method for dimensionality reduction (and dimensionality expansion) uses autoencoders. Autoencoders are feed-forward neural networks which are trained to approximate a function by which data can be classified. For every training input the difference between the input and the output is measured (using square error) and it is back-propagated through the neural network to perform the weight updates to the different layers. In a paper that compares stacked autoencoders with PCA with Gaussian SVM on 13 gene expression datasets, it was shown that autoencoders perform better on the majority of datasets [RFAM13]. Autoencoders use fine tuning, a back-propagation method for adjusting their parameters. Without back-propagation the autoencoders suffer from very low accuracies. A general problem with autoencoders is that a large number of internal layers can easily “memorise” the training data and create a model with zero error which will over-fit the data and be unable to classify future test data.

Independent Component Analysis (ICA) is also used for analysing microarray data [LB03, EDMA10] in combination with a clustering method. ICA finds the correlation among the data, and decorrelates it by maximizing or minimizing the contrast information. This is called “whitening”. The whitened matrix is then rotated to minimise the Gaussianity of the projection and in effect retrieves statistically independent data. It can be applied in combination with PCA. It is said that ICA works better if the data has been preprocessed with PCA [CCC⁺03]. This could, however, merely be due to the decrease in computational

load caused by the reduced dimensionality.

$$\begin{aligned} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{N-1} \\ X_N \end{bmatrix} &\rightarrow \begin{bmatrix} X_i \\ \vdots \\ X_k \\ X_n \end{bmatrix} \\ \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{N-1} \\ X_N \end{bmatrix} &\rightarrow \begin{bmatrix} Y_1 \\ \vdots \\ Y_K \end{bmatrix} = f \left(\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{N-1} \\ X_N \end{bmatrix} \right) \end{aligned}$$

Figure 4.6: **Feature Selection And Feature Extraction:** Difference between feature selection (top) and feature extraction (bottom)

The advantages and disadvantages of feature extraction and feature selection are shown in table 4.3 and their differences in figure 4.6.

Method	Advantages	Disadvantages
Selection	Preserves data characteristics for interpretability	
	Lower training times	Discriminative power
	Reduced over-fitting	
Extraction	Higher discriminating power	Loss of Data interpretability
	Control of over-fitting when unsupervised	Transformation maybe expensive

Table 4.3: **Advantages and Disadvantages between Feature Selection And Feature Extraction Approaches**

4.4.3 Affinity Matrices

Many classification algorithms use feature extraction from an affinity matrix, which records the similarity in the behaviour of each pair of variables in the problem. In the case of cancer data analysis we intuitively think of the genes as being variables whose values is an activity level measured by one or more spots on a microarray, and the classes of interest

are cancer subtypes, or more simply the sets of subtypes that will go into remission and those that will not. In this interpretation the affinity matrix will have the dimension of the number of genes, or gene probes, and consequently will be very large. We refer to this as the gene-by-gene affinity matrix. The similarity value that we calculate for each gene pair is based on extracting a tuple from each patient case (or sample). The number of these data points is comparatively small. Similarity may be computed by any distance metric, but the most commonly used is linear correlation. An alternative way to address the problem is to consider that each patient case (sample) is a variable. In this case the dimension of the affinity matrix is the same as the number of samples, and consequently is usually small. It is referred to as the sample-by-sample affinity matrix. If PCA is used for feature extraction then there is a direct relationship, known as the Kohonen-Lowe transformation, between the features extracted from the two forms of the affinity matrix.

Let us suppose the data is arranged in a matrix U where the columns are the gene probe measurements and the rows are the samples (or patient cases). Let there be n columns and N rows. The data can be mean-centred by subtracting the average of each column from each of its entries. Using correlation as the similarity measure we have that:

$$\Sigma_g = U \times U^T / (N - 1)$$

$$\Sigma_s = U^T \times U / (N - 1)$$

where Σ_g is the gene-gene affinity matrix, Σ_s is the sample-sample affinity matrix and N is the number of samples. The features used in classification are the eigenvectors of the affinity matrix, so starting with the sample-sample affinity matrix we compute the set of features Φ_s using:

$$\Sigma_s \Phi_s = \Phi_s \Lambda$$

$$(U U^T) \Phi_s = \Phi_s \Lambda$$

If we multiply both sides by U^T we get:

$$\begin{aligned}
U^T(UU^T)\Phi_s &= U^T\Phi_s\Lambda \\
(U^TU)(U^T\Phi_s) &= (U^T\Phi_s)\Lambda
\end{aligned}$$

the features Φ_g extracted from the gene-gene affinity matrix are given by:

$$\begin{aligned}
\Sigma_g\Phi_g &= \Phi_g\Lambda \\
(U^TU)\Phi_g &= \Phi_g\Lambda
\end{aligned}$$

from which we can infer that:

$$\Phi_g = U^T\Phi_s$$

Although the number of eigenvectors extracted from the gene-gene matrix (n) is much larger than the number extracted from the sample-sample matrix (N), in both cases the number of eigenvectors with non-zero eigenvalues will be $N-1$ because, assuming the samples are all linearly independent, the rank of both affinity matrices is $N-1$. Computationally it is much faster to calculate $U^T\Phi_s$ than to calculate Φ_g directly, since eigenvector extraction scales with the square of the dimension of the data, and no information is lost in the process.

4.5 Prior Knowledge

Prior knowledge has previously been used in microarray studies with the objective of improving the classification accuracy. One early method for adding prior knowledge in a machine learning algorithm was introduced by Eran Segal [SKFJ05]. It first partitions the variables into modules, which are gene sets that have the same statistical behaviour (share the same parents in a probabilistic network) and then uses this information to learn patterns. The modules were constructed using Bayesian networks and a Bayesian Scoring function to decide how well a variable fits in a module. The parents for each module were restricted to only some hundreds of possible genes otherwise the model was becoming too complex. Regression Trees were used to learn the module networks. The gene expression data was

taken from yeast in order to investigate how it responds to different stress conditions. The results were then verified using the *Saccharomyces* Genome Database [CHA⁺12]. Adding prior knowledge reduces the complexity of the model and the number of parameters making analysis easier. A disadvantage however of this method is that it relies only on gene expression data, which is noisy. Many sources of external biological information are available and can be integrated with machine learning and/or dimensionality reduction methods. This helps overcome one of the limitations of machine learning classification methods which is that they do not provide the necessary biological connection with the output. Adding external information with microarray data can give an insight to the functional annotation of the genes and the role they play in a disease, such as cancer.

4.5.1 Gene Ontology

Gene Ontology (GO) terms are a popular source of prior knowledge since they describe known functions of genes. Protein information found in the genes' GO indices has been combined with their expressions in order to identify more meaningful relationships among the genes [CX04, HP06]. A study infused GO information in a dissimilarity matrix [KZ10] using Lin's similarity measure [Lin98] in order to make the clustering results more biologically relevant. GO terms were also used as a way of weighting the longest partial path shared by two genes [CCM⁺04]. A pair-wise similarity matrix of gene expressions and the weight derived from the GO data were added together in order to produce gene clusters. Clusters created using GO terms were denser and more biologically relevant (genes detected were related to the particular type of cancer) than without the GO terms. GO term information integrated with gene expression was used by Chen and Wang [CW09b]. They applied SPCA to find the most significant principal components and classify cancer survival .

Not all of these methods have been compared to other forms of dimensionality reduction such as PCA or manifold which is a serious shortcoming as to their actual performance. It is however the case that in all of those papers some important limitations of GO terms are described. Some genes do not belong in a functional group and therefore cannot be used.

GO terms tend to be very general when it comes to the functional categories and that leads to bigger gene clusters that are not necessarily relevant in microarray experiments.

4.5.2 Protein-Protein Interaction

Other studies have used protein-protein interaction (PPI) networks for the same purpose [CLL⁺07]. Subnetworks are identified using PPI information. Iteratively more interactions are added to each subnetwork and scored using mutual information between the expression information and the class label in order to find the most significant subnetwork. The initial study showed that there is potential for using PPI networks but there is a lot of work to be done in order to take advantage of the prior knowledge more effectively. Prior knowledge methods tend to use this knowledge in order to filter data out or even penalise features. These features are called outliers and normally are the ones that have a much higher variance than every other point. Su et al. [SYD10] overlaid PPI and gene expression data. Using the PPI information genes that belonged in the same PPI network were tested for correlation and high discriminatory power. The top scoring PPI networks were combined in a single network. The genes belonging to the final network were removed from the PPI dataset and the process was repeated. The results showed that this method could classify breast cancer metastasis accurately using the genes in the final network.

4.5.3 Gene Pathways

The most promising results have been shown when using pathway information as prior knowledge. Many databases containing information on networks of molecular interaction in different organisms exists (KEGG, Pathway Interaction Database, Reactome, etc). It is widely believed that these lower level interactions can be seen as the building blocks of genetic systems, and can be used to understand high-level functions of the biological systems. KEGG pathways have been quite popular in network constrained methods which use networks to identify gene relations to diseases. A network-based penalty function for vari-

able selection has been introduced [LL08]. The framework used penalised regression, after imposing a smoothness assumption on the regression coefficients based on their location in the gene network. The biological motivation for this penalty is that the genes that are linked on the networks are expected to have similar functions and therefore bigger coefficients. The weights are also penalised using the sum of squares of the scaled difference of the coefficients between neighbour vertices in the network in order to smooth the regression coefficients. The results were promising in terms of identifying networks and subnetworks of genes that are responsible for a disease. However the authors only used 33 networks and not the entire set of available networks. A similar theoretical approach also exists, which according to the authors can be applied to cancer microarray data but to date has not been explored [RZD⁺07]. The proposed method was based on Fourier transformation and spectral graph analysis. The gene expression profiles were reconstructed using prior knowledge from gene networks to modify the distance. They use the assumption that the information lies in the low frequency component of the gene expression while the high frequency component is mostly noise. Using spectral decomposition the smaller eigenvalues and corresponding eigenvectors are kept (the smaller the eigenvalue the smoother the graph). A linear classifier can be inferred by penalising the regression coefficients based on network information. The Biological Pathway-Based Feature Selection Algorithm (BPFS) [BKG⁺09] also utilizes pathway information for microarray classification. It uses SVMs to calculate the marginal classification power of the genes and puts those genes in a separate set. Then the Influence Factor for each of the genes in the second set is calculated. This is an indication of the interaction of every gene in the second set with the already selected genes. If the Influence factor is low the genes are added to the set of the selected genes. The Influence factor is the sum of the shortest pathway distance that connects the gene to be added with each other gene in the set.

4.6 Discussion

Feature selection methods that perform statistical analysis tend to be less computationally expensive than feature extraction methods. Most feature selection methods work using statistical significance tests. However these methods have trouble dealing with datasets where the number of samples is much smaller than the number of features. Due to improvements in sequencing techniques, microarray experiments can now capture information for all the genes in the human genome, which leads to datasets of hundreds of thousands of features. In addition most of these methods require the variances of the different classes to be identical. This is not always the case when dealing with biological data that can be extremely noisy and contain outliers. In addition, the normality assumption does not always hold, especially when the number of samples is small, which is a pre-requisite for proving significance for statistical analysis tests. Another problem is that even for multivariate methods, it is difficult to check all the possible combinations of genes. Unless prior information is used it is computationally expensive and therefore it is limited to some groups of genes that either have specific expression patterns or are identified using combinatorial searches in the dataset. Methods like SVM-RFE or random forests tend to be restricted to only one classifier and are not widely applicable to others.

Feature extraction methods face similar problems, due to the high dimensionality of the microarray datasets. Moreover there is no connection between the output and its biological meaning. Some of the genes that are identified by the model as important might not be related to the actual outcome and are just part of the model because they exhibit specific expression patterns, which could be due to the noisy nature of the microarray data. Some biological component needs to be added in order to make classification more accurate and more specific to the problem.

Adding prior biological knowledge in machine learning algorithms is gaining more popularity, since it is a way of dealing with the growing size of the datasets. Care must be taken when the prior knowledge is integrated in the model so it will not over-fit. Adding prior knowledge without integrating it with the dataset information, or using it as a way of pre-

processing it to filter out information, will result in models that reflect the prior knowledge instead of a combination between the prior knowledge and the microarray dataset. The prior knowledge used needs to come from a reliable, well curated source. Pathways are the most popular source now but most of the methods proposed only use a fraction of the available pathways.

Chapter 5

A priori Manifold Learning

5.1 Introduction

In this chapter a method which incorporates manifold learning along with a novel approach for estimating the k -Nearest Neighbours is presented.

A new way of constructing the manifold was investigated which makes use of prior knowledge. Prior knowledge has previously been used in microarray studies [CX04, KZ10, CCM⁺04] with the objective of improving the classification accuracy. Although several types of prior knowledge could have been used, the KEGG pathways database was deemed to be best for this purpose, since the KEGG pathway database describe actual interactions between genes. KEGG (Kyoto Encyclopedia of Genes and Genomes) [Kan97] is a collection of databases containing information on networks of molecular interactions in different organisms. It is believed that these lower level interactions can be seen as the building blocks of genetic systems, and can be used to understand high-level functions of biological systems. KEGG pathways have been a popular in network constrained methods which use networks to identify gene relations to diseases [LL08, RZD⁺07]. Other studies have used protein-to-protein interaction (PPI) networks for the same purpose [CLL⁺07]. Gene Ontology (GO) terms are a popular source of prior knowledge since they describe known functions of genes [CW09b, CX04, KZ10, CCM⁺04].

The proposed method of building the manifold is as follows. In common with all previous methods the affinity matrix is built first from a set of microarrays. A gene-by-gene affinity matrix is a square matrix whose dimension is the same as the number of gene probes in the microarray data. The matrix is symmetric and each entry is a similarity measure (for example covariance) of the expression levels of the two genes that index it. The information from the KEGG pathways was then fused with the affinity matrix by increasing the values for gene pairs with a strong relationship in KEGG. Next a conventional manifold learning method was applied to the fused affinity matrix to find the manifold. Having found the manifold of the gene probes the raw data is projected onto it so that the classification experiments can be carried out. This means that the KEGG pathway data is only involved in building the manifold. In contrast to previous data fusion approaches [TP07], the prior knowledge is only used to find a suitable space for representing the data. This ensures that the results are more specific to the biological content of the dataset under investigation. In contrast to other prior knowledge methods, *a priori* manifold learning does not use prior knowledge to filter out data from the original dataset, or just weight the affinity matrix.

The cluster validation and accuracy measurements, along with the original Isomap algorithm and PCA were built using the sklearn [PVG⁺11] package for Python. The method was developed by myself and MEng student George Trigeorgis. The original paper can be found in [HTG14].

5.2 The *A Priori* Manifold Learning Algorithm

Biological pathways are usually directed graphs with labelled nodes and edges representing associations of genes participating in a biological process. These interactions can help in understanding the underlying processes in different organisms as well as their contribution to diseases. Some of the interactions include regulation of gene expression, transmission of signals and metabolic processes. It is not yet completely clear as to why and how these interactions came to exist and what other, if any, external factors contribute to them. When it

comes to machine learning, information from the pathways can be used as prior knowledge for either feature selection or dimensionality reduction of the original data set. For this implementation, KEGG pathways are used as a way to weight the distance between the gene-to-gene interactions. Genes that share a greater number of common pathways should have a higher probability in being closer together when it comes to clustering, since it is already known that they interact with each other. The metric used in weighting the distances was based on a feature selection method [CSTK01]. This method works by assigning weights on the different features so that the more important ones play a greater role in the equation. By exploiting the use of these weights the classical k -nearest points algorithm using the weighted Mahalanobis distance was modified as shown in equation (5.4).

The algorithm to find the k -nearest points works as follows:

1. Given a pair of probes the Jaccard coefficient is used to evaluate the similarity of pathways they share together. This index, coined by Paul Jaccard [Jac01] and shown in equation 5.1, is a statistic commonly used for comparing similarity and diversity of sample sets.

$$R(i, j) = \frac{|\xi(i) \cap \xi(j)|}{|\xi(i) \cup \xi(j)|} \quad (5.1)$$

where $\xi \subset$ power-set of the KEGG Pathway.

2. The distance metric selected to calculate the gene-to-gene distance is the Mahalanobis distance. It is measured using the correlations between two gene feature sets.

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \quad (5.2)$$

where S is the covariance matrix.

3. The weights equation is shown in equation for genes i and j is shown in equation 5.3

$$w_{ij} = \exp(-\eta \times R(i, j)) \quad (5.3)$$

where η is a learning parameter and R is the Jaccard coefficient. The learning parameter η is a way of minimising or maximising the influence of any given feature in the dataset. When η is large the changes in the dataset are exponentially reflected on the weights. For example the way the parameter η affects endometrial cancer results is shown in figure 5.1. The η value selected for the embedding of the endometrial cancer was 19000. It is the value with the highest Dunn index as shown in the figure 5.1

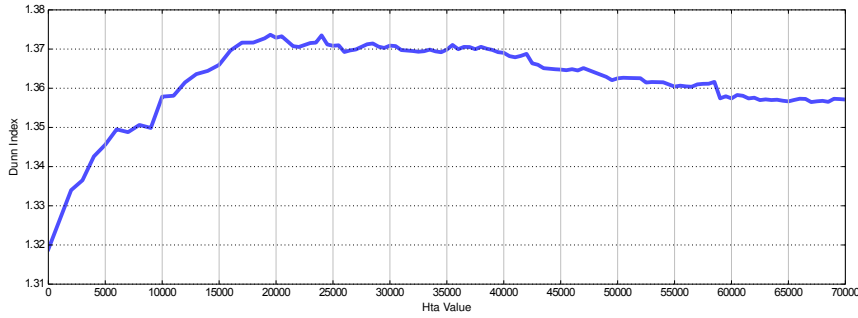


Figure 5.1: **Endometrium Cancer:** How the η value affects the value for the Dunn index

The η value was optimised using grid search [BB12]. Grid search works with cross-validation and is a simple, general purpose method that can always be applied. It involves computing the error at every point on a grid spanning the range of parameter values. In our case there was a predefined range of η values the algorithm had to run for, for example, from 0 to 10000 with a step of 100. Because that range was large and would take a lot of CPU time, a variation of the randomised grid search was used. Initially the algorithm run for all values as long as the accuracy was increasing. The algorithm was run using randomly sampled grid points (without replacement). The grid point with the largest η value was recorded. After sampling a specific number of points without improvement of the accuracy the algorithm was terminating.

4. The weights along with the Mahalanobis distance are expressed as:

$$D(i, j) = \sqrt{w_{i,j} \times d(\vec{i}, \vec{j})} \quad (5.4)$$

where i and j are two different probes.

This is shown in algorithm 5.1.

Algorithm 5.1. Calculation of the k -Nearest points of the manifold: First the Jaccard coefficient is calculated, then the Mahalanobis distances among the genes and the weights.

```

for each probe  $i$  in probes do
  for each probe  $j$  in probes do
     $R(i, j) = \frac{|\xi(i) \cap \xi(j)|}{|\xi(i) \cup \xi(j)|}$ 
  end for
end for
 $R = \frac{R}{\sum_{i,j} R(i,j)}$ 
for each probe  $i$  in probes do
  for each probe  $j$  in probes do
     $w_{ij} = \exp(-\eta \times R(i, j))$ 
     $distance(i, j) = \sqrt{w_{ij} \times \sum_{k=1}^n Mahalanobis(i_k, j_k)}$ 
  end for
  NearestNeighbours( $i$ ) = sorted(distances( $i$ ))
end for

```

The shortest paths Δ are found using either the Dijkstra [Dij59] or Floyd-Warshall algorithm [Flo62]. Dijkstra's algorithm is usually preferred since it is faster and the weights are non-negative. The Isomap mapping is performed by calculating the eigenvalues of Θ as shown in equation 3.5.

5.3 Datasets

To verify the effectiveness of *a priori* manifold learning algorithm the following datasets were used:

1. GEMLeR [SK10], which provides a collection of gene expression datasets that can be used for benchmarking gene expression oriented machine learning algorithms. Each of the gene expression samples in GEMLeR came from a large publicly available repository. GEMLeR was mainly preferred because:

- The processing procedure of tissue samples is consistent;
- The same Affymetrix microarray assay platform is used (Affymetrix GeneChip U133 Plus 2.0);

- There is a large number of samples for different tumour types;
- Additional information is available for combined genotype-phenotype studies.

2. Acute lymphoblastic leukaemia (ALL) dataset. ALL is a form of leukaemia characterised by excess lymphoblasts. There are two main types of acute leukaemia: T-cell ALL and B-cell ALL. T-Cell acute leukaemia is aggressive, progresses quickly and is more common in older children and teenagers. B-Cell ALL leukaemia [CSG09] is another type of ALL, originating in a single cell and characterised by the accumulation of blast cells that are phenomenologically reminiscent of normal stages of B-cell differentiation.

A summary of the datasets is shown in table 5.1. For the experiments in Chapter 5 and Chapter 6 the classification method was "one-vs-all". There was a total of 1545 samples in the dataset for 9 types of cancer (Breast, Colon, Kidney, Ovary, Lung, Uterus, Omentum, Prostate, Endometrium). The experiments done, were for distinguishing between the different types of. For example for breast cancer, the breast cancer samples were left intact and everything else was marked as "other". Therefore the classification was binary. The *a priori* manifold learning algorithm was used to devise a model that can separate breast cancer from every other cancer in the dataset. The Acute Lymphoblastic Leukaemia dataset was not used for classification but only for visualisation purposes for the three different types of leukaemia.

Type Of Cancer	Number Of Samples	Number Of Genes
Breast Cancer	344 cancer samples vs Other (1201 samples)	10935
Colon Cancer	286 cancer samples vs Other (1259 samples)	10935
Kidney Cancer	260 cancer samples vs Other (1285 samples)	10935
Ovary Cancer	198 cancer samples vs Other (1347 samples)	10935
Lung Cancer	126 cancer samples vs Other (1419 samples)	10935
Uterus Cancer	124 cancer samples vs Other (1421 samples)	10935
Omentum Cancer	77 cancer samples vs Other (1468 samples)	10935
Prostate Cancer	69 cancer samples vs Other (1476 samples)	10935
Endometrium Cancer	61 cancer samples vs Other (1484 samples)	10935
Acute Lymphoblastic Leukaemia	19 B-Cell vs 8 T-Cell vs 11 Normal	5000

Table 5.1: **Datasets Used**

5.4 Results

A priori manifold learning was tested against the original Isomap algorithm and PCA. The Dunn index was used as a metric for evaluating the density and the structure of the clusters in the embedding. The k -Nearest Neighbours (k -NN), Support Vector Machines (SVMs) and Linear Discriminant Analysis (LDA) classifiers were applied along with 10-fold cross validation to test the accuracy of the model. Nine different types of cancer were used to evaluate the methods (GEMLeR) and a smaller dataset (ALL) was used to visualise the results. The datasets are described in table 5.1. The evaluation scheme is shown in figure 5.2.

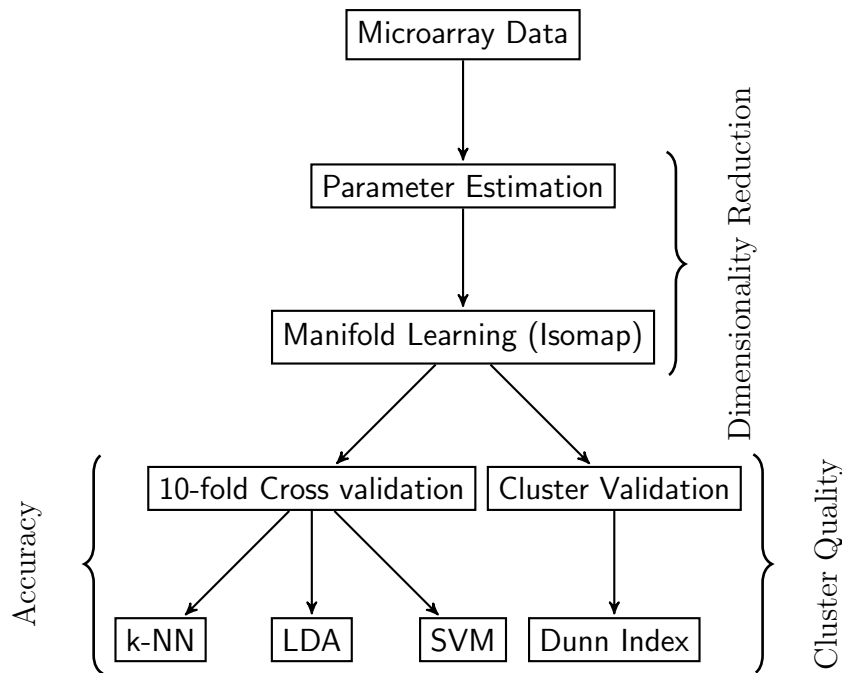


Figure 5.2: **Evaluation benchmark:** The η parameter is estimated and the resulting embedding is evaluated using cluster quality and cross validation metrics

5.4.1 Internal Evaluation

Dunn Index

The Dunn index was discussed in chapter 3. For these experiments the Dunn index can indicate how well the resulting embedding separates the samples according to their label, since it uses the labels of each sample as the cluster indicators. In practice manifold learning does not create any clusters but if the embedding is done in a successful way many points will end up being next to each other, since the embedding is just a mapping from the original dataset to a different space. This experiment was run for different dimensional embeddings (2 to 50 components) as the component that will end up being used in the embedding is heavily dependent on the complexity of the data.

A priori manifold learning is implemented using a gene-by-gene affinity matrix, since the prior knowledge involved is in the form of gene to gene similarities, and is applied to the

affinity matrix rather than to the original data. It is not possible to construct a corresponding sample-by-sample affinity matrix for this algorithm. However classification using the standard PCA and ISOMAP algorithms, which do not involve prior knowledge, can be computed using features drawn from either form of the affinity matrix. Figures 5.3 and 5.4 compare the *a priori* manifold learning algorithm results with both PCA and ISOMAP classifiers. In figure 5.3 the PCA and ISOMAP results used features drawn from the sample by sample affinity matrices. In figure 5.4 the PCA and ISOMAP results used features drawn from the gene-by-gene affinity matrices. The *a priori* manifold learning algorithm was the same in both figures.

The results for the Dunn index for sample-by-sample experiments in figure 5.3 and for gene-by-gene experiments in figure 5.4 show that *a priori* manifold learning creates denser clusters in all cases except colon, uterine and lung cancer. From the graph induced for the colon dataset for both sample-by-sample and gene-by-gene experiments and the uterine dataset in the gene-by-gene experiments it can be seen that *a priori* manifold learning outperforms PCA and Isomap for embeddings with lower dimensions. The goal is to create an embedding with as few components possible to represent the original high-dimensional data. For the lung dataset in the sample-by-sample experiments more samples are needed to create a more accurate embedding.

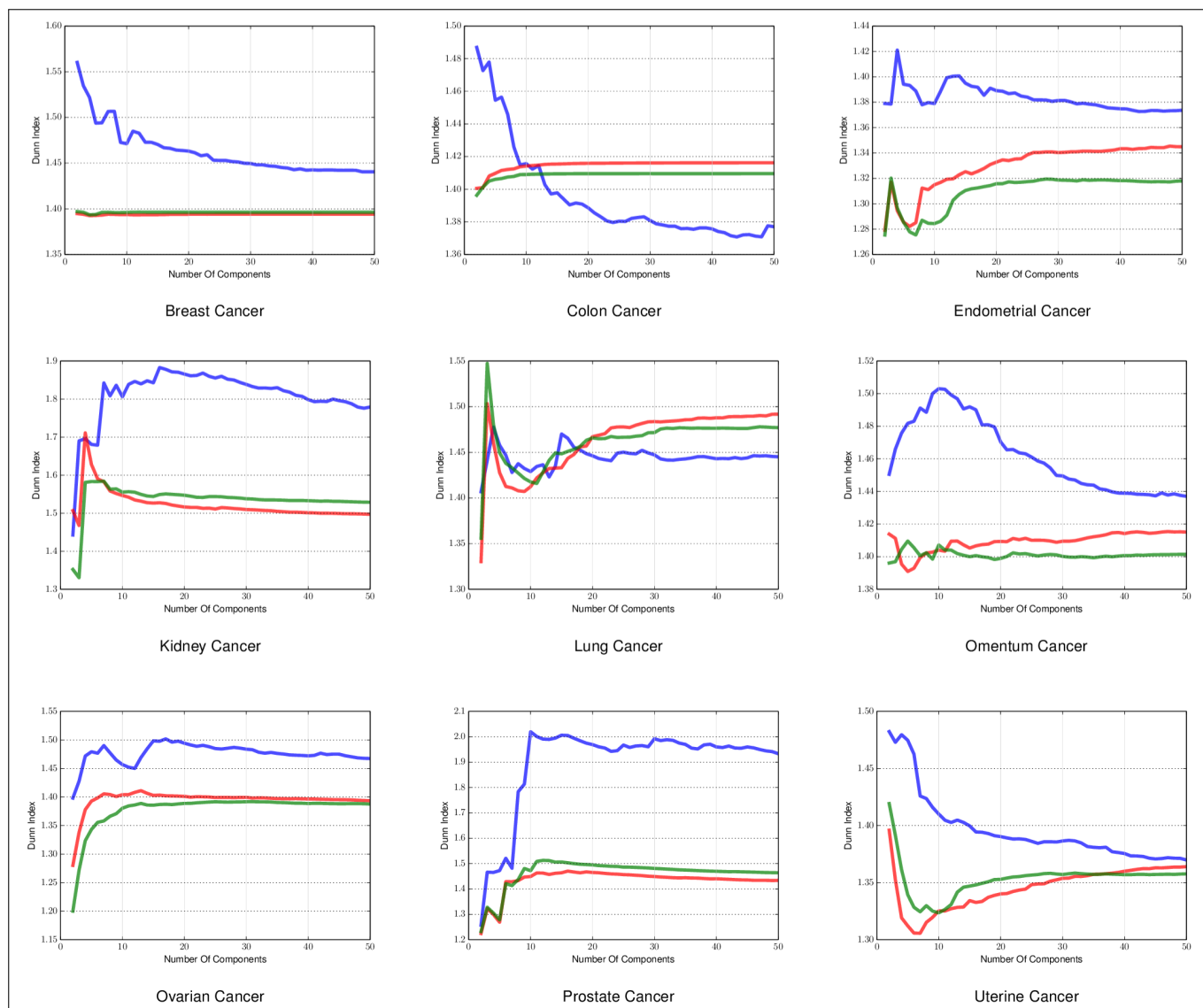


Figure 5.3: **Dunn Index for cancer classification results:** The Dunn Index found using *a priori* manifold learning (Blue) compared with PCA (Green) and Isomap (Red). PCA and ISOMAP were computed using features from the sample-by-sample affinity matrix.

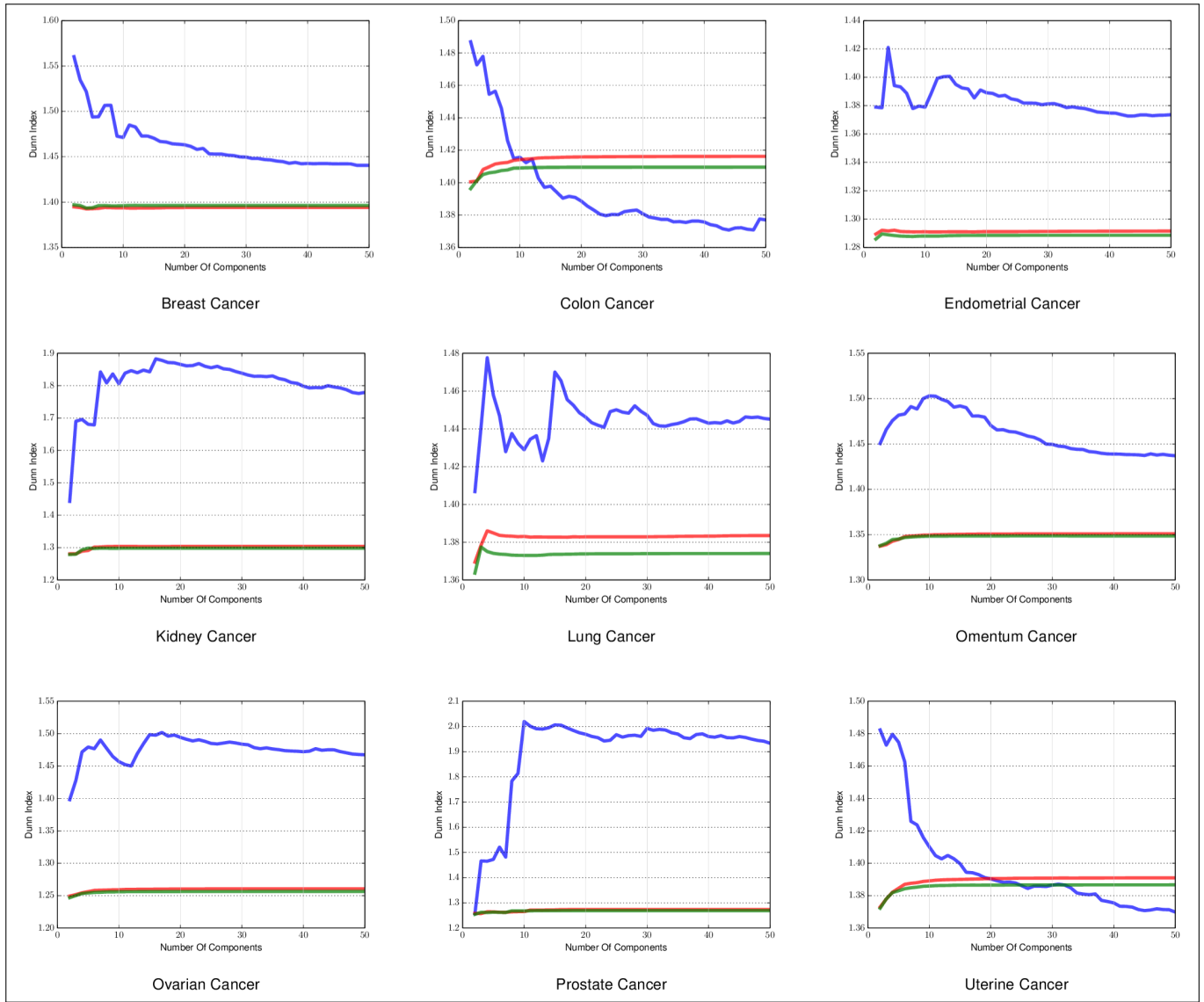


Figure 5.4: **Dunn Index for cancer classification results:** The Dunn Index found using *a priori* manifold learning (Blue) compared with PCA (Green) and Isomap (Red). PCA and ISOMAP were computed using features from the gene-by-gene affinity matrix

5.4.2 Ten Fold Cross-Validation

To evaluate the accuracy of the embeddings k -NN and LDA classifiers were used with ten fold cross-validation. In order to get the values, the trapezoidal rule which approximates the definite integral of the plots was used. Results are shown in table 5.2 for sample-by-sample experiments and in table 5.3 for gene-by-gene experiments using k -NN. The corresponding results for LDA are shown in table 5.4 for sample-by-sample and in table 5.5 for gene-by-gene experiments. The cases in which *a priori* manifold learning outperforms the rest of the

methods are emphasised in bold. It should be noted that the variance is small enough such that the individual accuracies of the experiments can be compared safely. The tables for the variance can be found in Appendix B.1. In the LDA results *a priori* manifold learning outperforms PCA and Isomap for 5 out of 9 datasets. These are the same datasets for both sample-by-sample and gene-by-gene experiments. For k -NN, the results for the gene-by-gene experiments show that *a priori* manifold learning outperforms the other two methods in all cases, while for the sample-by-sample experiments, *a priori* manifold learning does not perform so well.

When performing cross validation experiments both PCA and Isomap features can be computed using either the gene-by-gene affinity matrix or the sample-by-sample affinity matrix. The latter is a square matrix with dimension equal to the number of microarrays used in the experiment. Each entry represents the similarity (or distance) between the corresponding pair of microarrays. It is considerably smaller than the gene-by-gene matrix and consequently more robust to noise. *A priori* manifold learning can only be computed using the gene-by-gene affinity matrix. This is because the prior knowledge extracted from the KEGG database is in the form of similarities between gene pairs. The results show that both PCA and Isomap perform better using the sample-by-sample affinity matrix.

The sample-by-sample affinity matrix cannot be computed directly using *a priori* manifold learning since it needs the genes for constructing the affinity matrix, and therefore *a priori* manifold learning only operates on a gene-by-gene affinity matrix. For the GEMLeR dataset, the sample-by-sample affinity matrix has dimensions 1545 by 1545. This is the number of microarrays in the dataset. The gene-by-gene affinity matrix is 10935 by 10935 which is the number of gene probes in each microarray.

The error bars with one standard deviation of uncertainty for the 10-fold cross validation for the k -NN and the LDA classifier can be found in appendix B.2.

Type Of Cancer	A Priori Manifold Learning	Isomap	PCA
Breast cancer	0.806	0.863	0.879
Colon cancer	0.868	0.897	0.906
Kidney cancer	0.937	0.931	0.932
Ovary cancer	0.841	0.842	0.851
Lung cancer	0.902	0.911	0.917
Uterus cancer	0.891	0.890	0.891
Omentum cancer	0.914	0.912	0.912
Prostate cancer	0.955	0.954	0.954
Endometrium cancer	0.923	0.924	0.926

Table 5.2: **Cancer classification accuracy measured using the k -Nearest Neighbours algorithm and Ten Fold Cross validation:** The results for PCA and ICA were computed using features from sample-by-sample affinity matrices. The *a priori* manifold learning method is either comparable with, or outperforms the others. The cases in which *a priori* manifold learning outperforms the other methods are shown in bold.

Type Of Cancer	<i>A priori</i> manifold learning	Isomap	PCA
Breast cancer	0.806	0.782	0.792
Colon cancer	0.868	0.834	0.834
Kidney cancer	0.937	0.900	0.903
Ovary cancer	0.841	0.834	0.838
Lung cancer	0.902	0.883	0.886
Uterus cancer	0.891	0.882	0.881
Omentum cancer	0.914	0.912	0.912
Prostate cancer	0.955	0.943	0.945
Endometrium cancer	0.923	0.922	0.922

Table 5.3: **Cancer classification accuracy measured using the k -Nearest Neighbours algorithm and Ten Fold Cross validation:** The results for PCA and ICA were computed using features from gene-by-gene affinity matrices. The *a priori* manifold learning method is either comparable with, or outperforms the others. The cases in which *a priori* manifold learning outperforms the other methods are shown in bold.

Type Of Cancer	<i>A priori</i> manifold learning	Isomap	PCA
Breast cancer	0.890	0.901	0.912
Colon cancer	0.906	0.914	0.925
Kidney cancer	0.956	0.952	0.953
Ovary cancer	0.871	0.867	0.870
Lung cancer	0.935	0.938	0.941
Uterus cancer	0.906	0.900	0.905
Omentum cancer	0.927	0.923	0.924
Prostate cancer	0.973	0.972	0.972
Endometrium cancer	0.937	0.934	0.930

Table 5.4: **Cancer classification accuracy measured using linear discriminant analysis and ten fold cross validation:** The results for PCA and ICA were computed using features from sample-by-sample affinity matrices. The *a priori* manifold learning method is either comparable with, or outperforms the others. The cases in which *a priori* manifold learning outperforms the other methods are shown in bold.

Type Of Cancer	<i>A priori</i> manifold learning	Isomap	PCA
Breast cancer	0.890	0.888	0.910
Colon cancer	0.906	0.914	0.924
Kidney cancer	0.956	0.911	0.954
Ovary cancer	0.871	0.945	0.870
Lung cancer	0.935	0.924	0.940
Uterus cancer	0.906	0.901	0.905
Omentum cancer	0.927	0.926	0.923
Prostate cancer	0.973	0.970	0.972
Endometrium cancer	0.937	0.932	0.930

Table 5.5: **Cancer classification accuracy measured using Linear Discriminant Analysis and Ten Fold Cross validation:** The results for PCA and ICA were computed using features from gene-by-gene affinity matrices. The *a priori* manifold learning manifold learning method is either comparable with, or outperforms the others. The cases in which *a priori* manifold learning outperforms the other methods are shown in bold.

Receiver Operating Characteristic Curves

In addition Receiver Operating Characteristic (ROC) curves were constructed to illustrate the ratio of true positive to false positive results. Three different classification methods were used to illustrate the effectiveness of *a priori* manifold learning .

k - Nearest Neighbours (*k*-NN)

For the *k*-NN classifier the results obtained for the ROC curves agree with the 10-fold cross validation results. *A priori* manifold learning performs better in all the gene-by-gene experiments as shown in figure 5.5, while in the sample-by-sample ones only performs better in one dataset as shown in figure 5.6

Support Vector Machines (SVMs)

Using SVMs *a priori* manifold learning performs better in 6 out of 9 datasets for the gene-by-gene experiments (figure 5.7) while in the sample-by-sample experiments (figure 5.8) it performs better in all datasets.

Linear Discriminant Analysis (LDA)

For the same purpose LDA was used. For the gene-by-gene experiments the results are shown in figure 5.9 and for sample-by-sample experiments the result are shown in figure 5.10. *A priori* manifold learning performs better in 5 out of 9 datasets.

Comparing the ROC curves of the three different classifiers shows that the *a priori* manifold learning gives consistent results for LDA and SVMs for both gene-by-gene and sample-by-sample experiments. However, the *k*-NN classifier seems to perform very well for the gene-by-gene experiments but not for the sample-by-sample ones. A possible explanation for this is that discriminant methods like SVMs and LDA use a data model computed from the whole dataset, and may therefore be more robust to noise and other artefacts. By contrast the *k*-NN classifier relies on the local distribution of the data, and could therefore be less effective particularly in small sample size problems.

The Acute Lymphoblastic Leukaemia (ALL) dataset for leukaemia was analysed to demonstrate how the different cells were clustered. The ALL dataset is simple enough to visualise

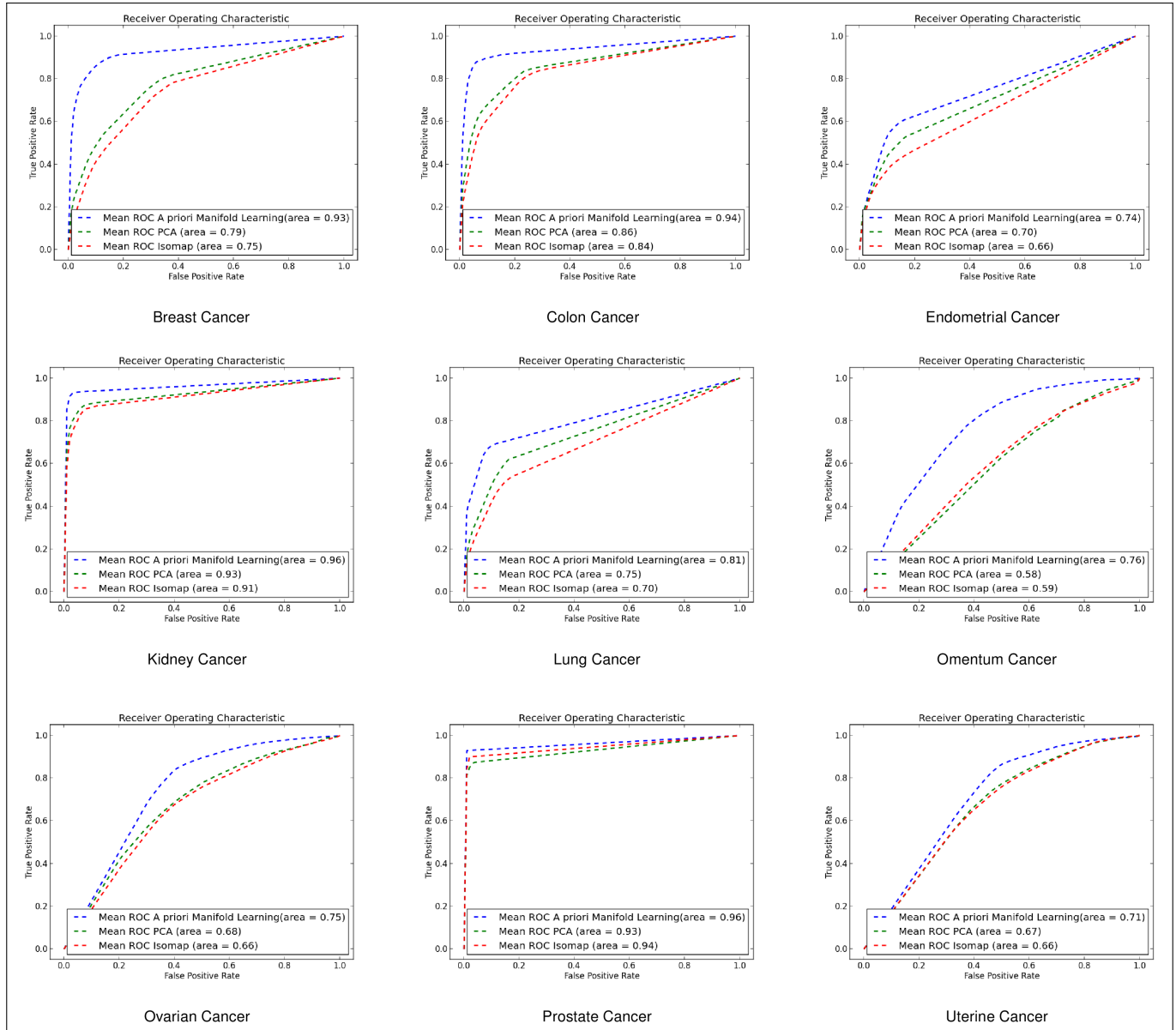


Figure 5.5: ROC curves for cancer classification using k -Nearest Neighbours: *A priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red). PCA and ISOMAP were computed using the gene-by-gene affinity matrix.

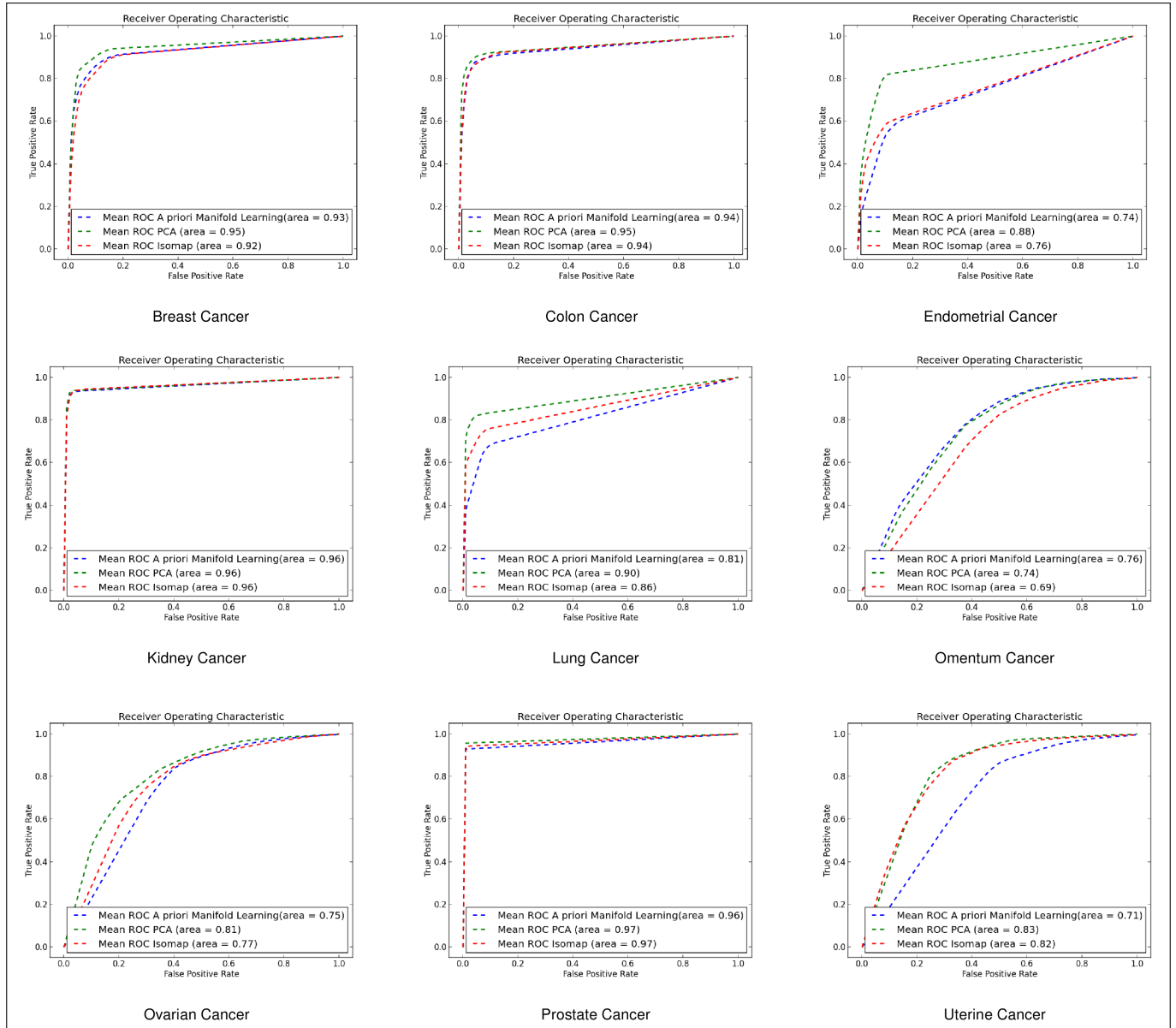


Figure 5.6: ROC curves for cancer classification using k-Nearest Neighbours: *A priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red). PCA and ISOMAP were computed using the sample-by-sample affinity matrix.

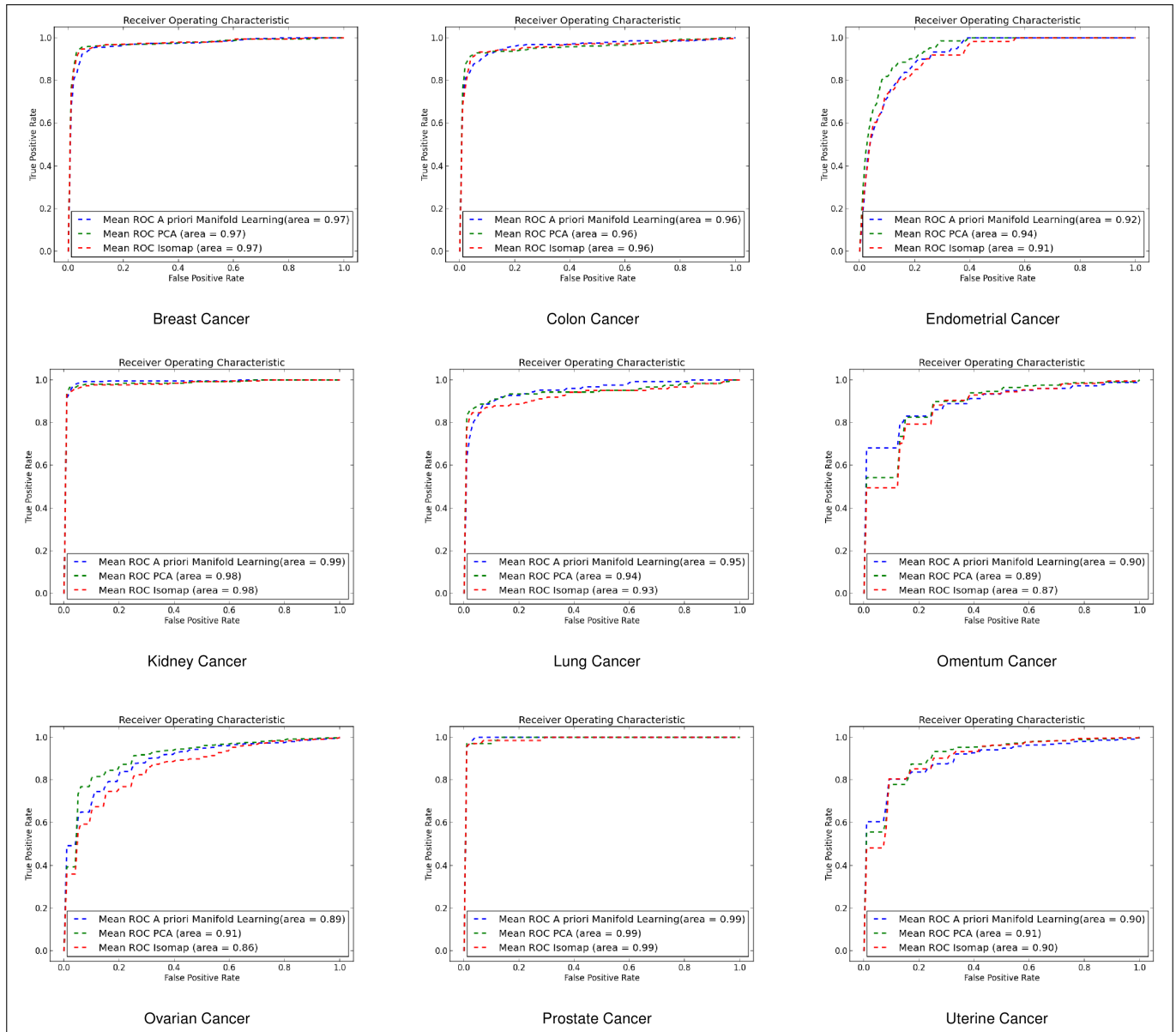


Figure 5.7: ROC curves for cancer classification using support vector machines: *A priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red). PCA and ISOMAP were computed using the gene-by-gene affinity matrix.

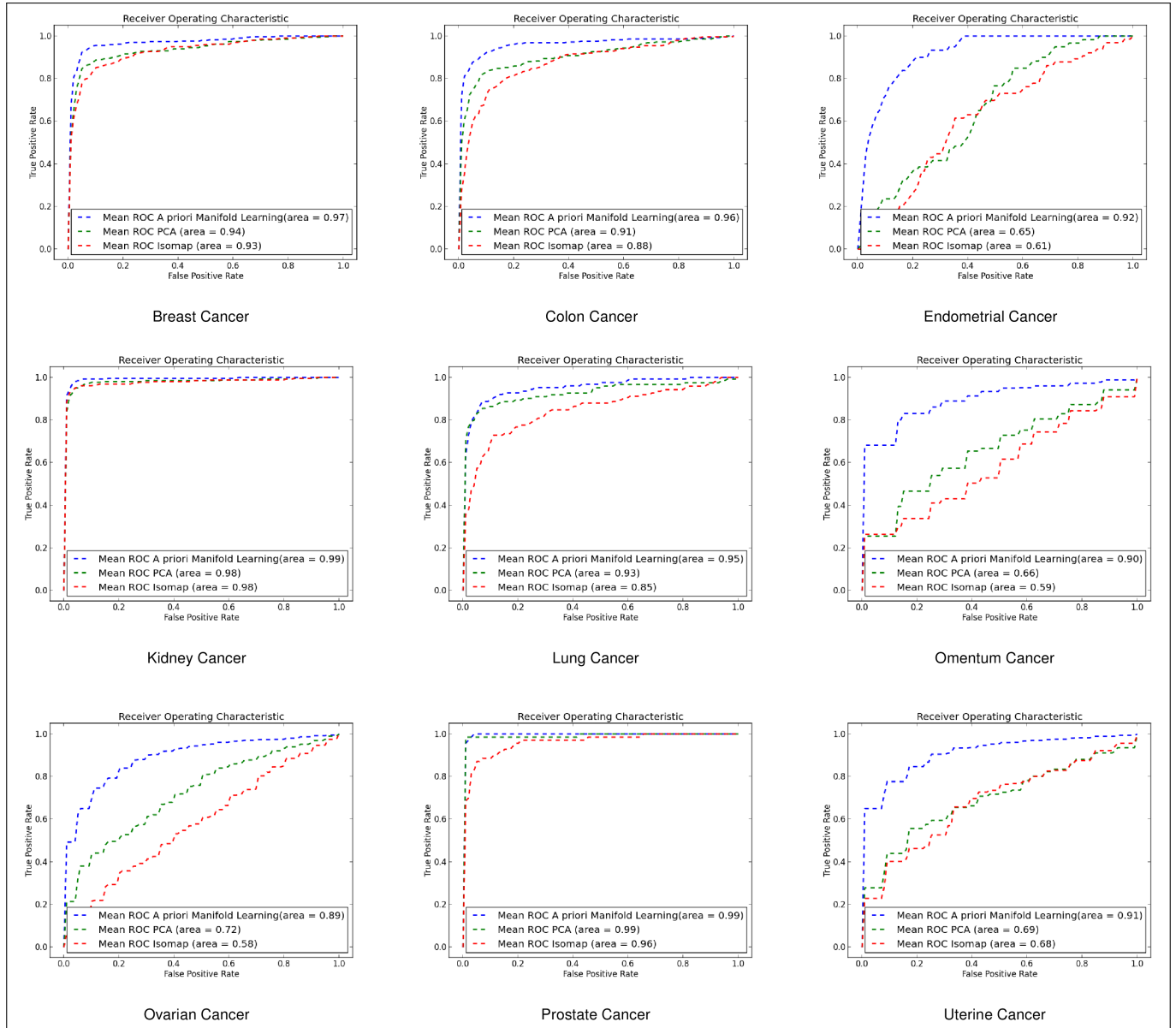


Figure 5.8: ROC curves for cancer classification using support vector machines: *A priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red). PCA and ISOMAP were computed using the sample-by-sample affinity matrix.

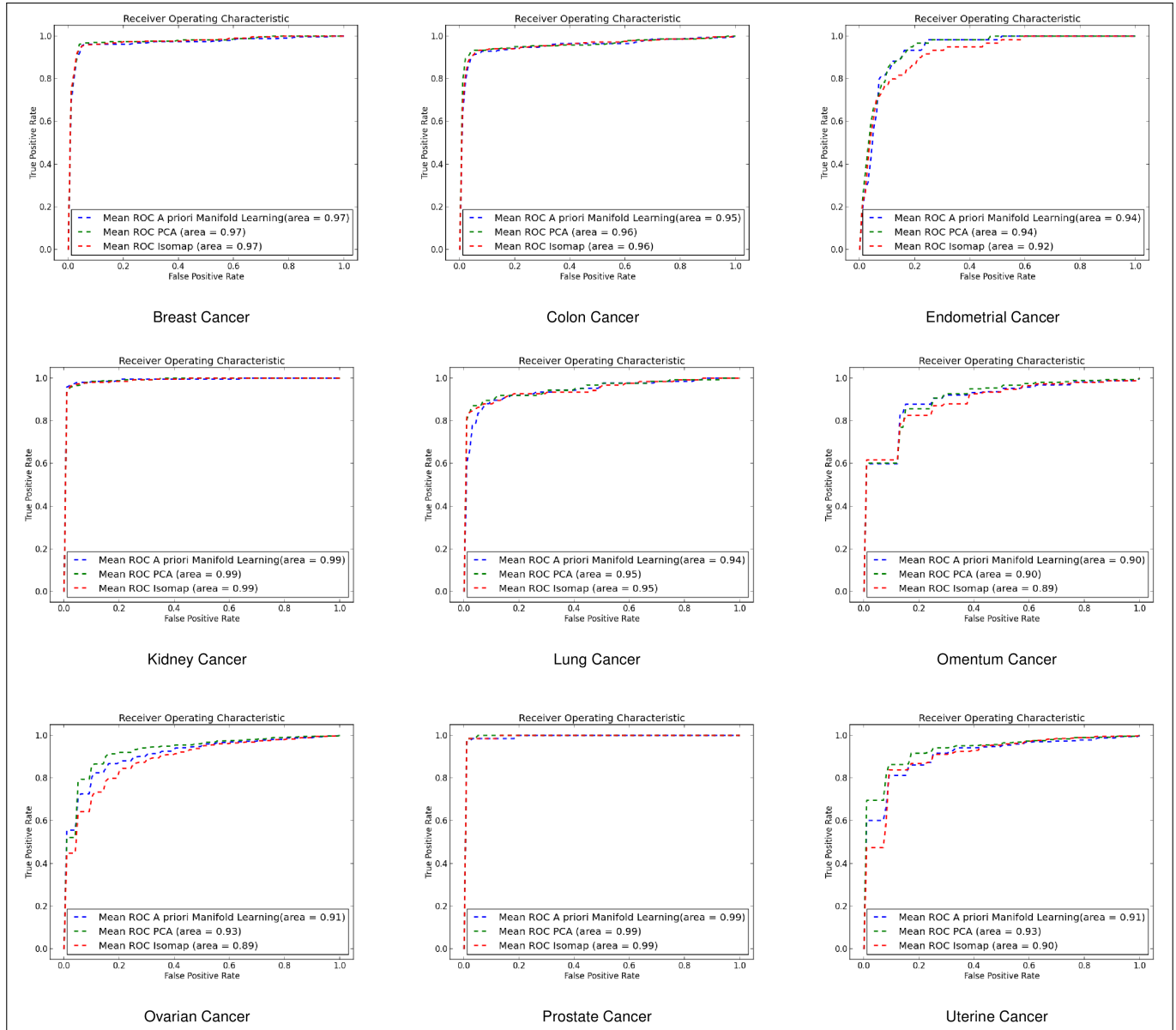


Figure 5.9: ROC curves for cancer classification using Linear Discriminant Analysis: A priori manifold learning (blue) compared with PCA (Green) and Isomap (Red). PCA and ISOMAP were computed using the gene-by-gene affinity matrix.

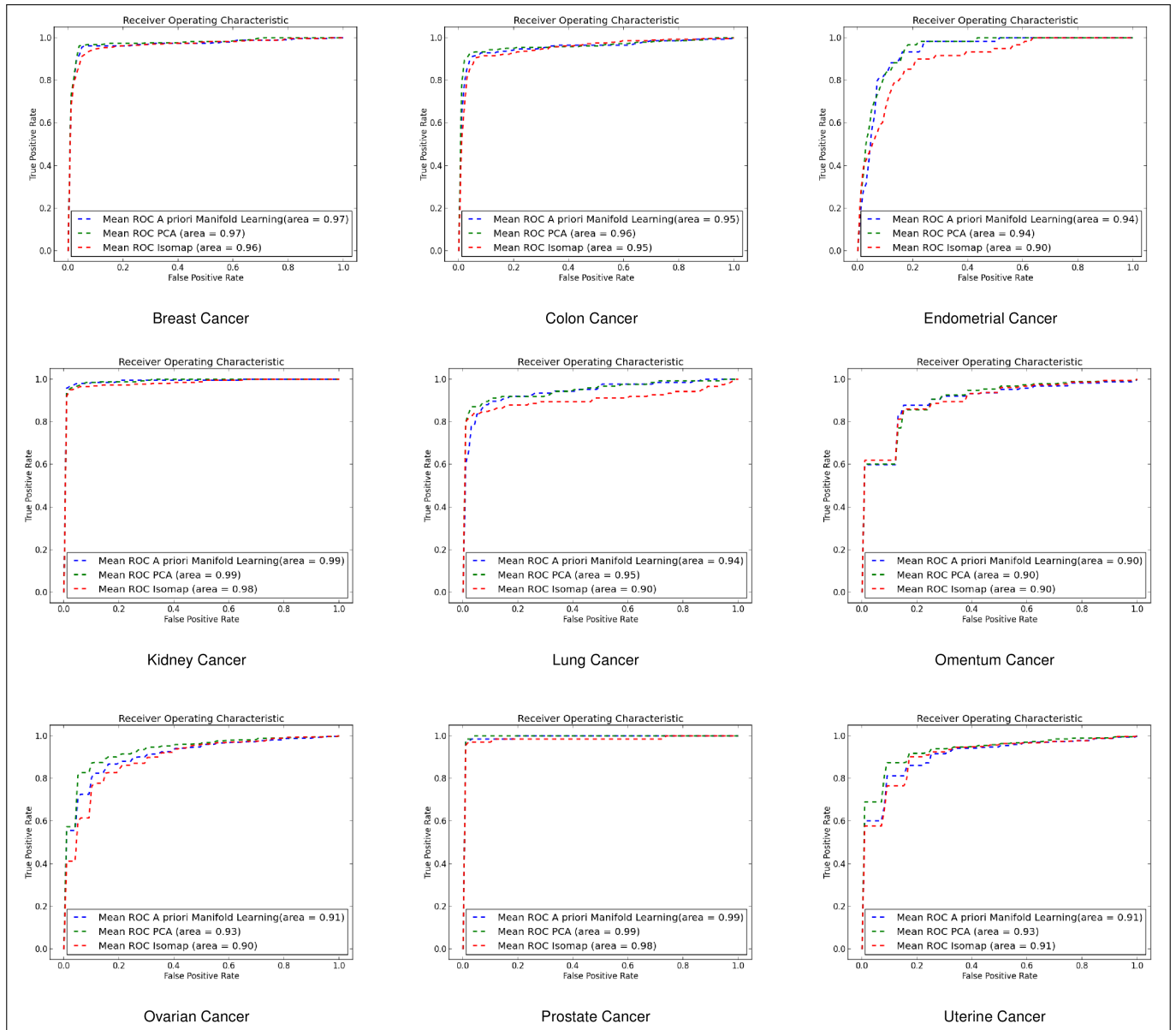


Figure 5.10: ROC curves for cancer classification using linear discriminant analysis: A priori manifold learning (blue) compared with PCA (Green) and Isomap (Red). PCA and ISOMAP were computed using the sample-by-sample affinity matrix.

and has been used before [BTGM04] to demonstrate the clustering of the different types of cells in two dimensions. The embedding with the samples annotated with their true labels is found in figure 5.11.

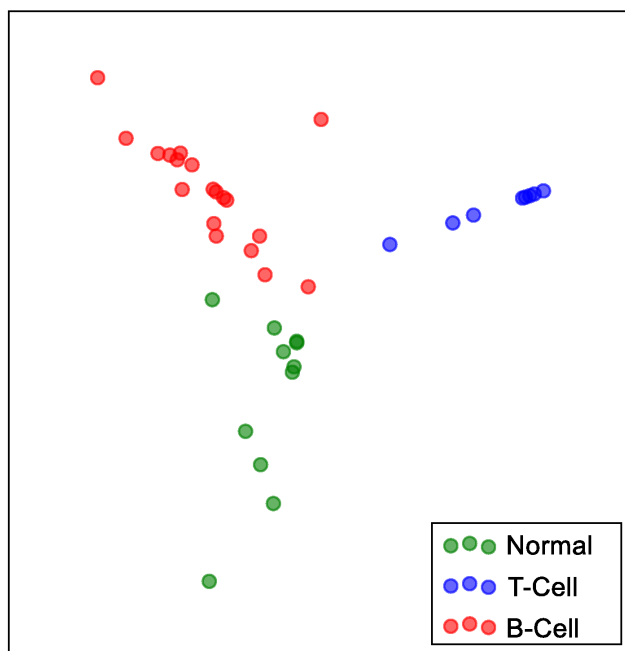


Figure 5.11: **Leukaemia cell:** Two dimensional manifold of the three different leukaemia cells. Clusters of the different cell types are formed and are easily distinguished in the lower dimensional space.

5.4.3 Pathway Robustness

The robustness and the effectiveness of using pathways was shown by removing pathways using a uniform distribution with different probabilities. By removing a percentage of the KEGG pathways in different runs of the algorithm it is shown how the number of pathways affects the performance. The results are shown in appendix B.3. In most cases removing pathways results in reducing the accuracy as shown in the ROC curves and the Dunn index graphs.

5.5 Discussion

Conventional manifold learning algorithms, such as Isomap, aim to project the microarray data to a lower dimensional space in which functionally different clusters are better separated. The lower dimensional space is a manifold (hypersurface) contained in the original data space and found from the local distribution of the data. A large representative dataset is used to compute the manifold. This method provides a way of improving the way Isomap finds the k -nearest points and creates the neighbouring graph by utilising KEGG pathway information. The KEGG data is a form of prior knowledge which is better curated and more reliable than the microarray data. Once the manifold has been constructed the raw microarray data is projected onto it and clustering and classification can take place. The method was called *a priori* manifold learning and was compared with the original Isomap and the PCA algorithm, since PCA is the most commonly used method for dimensionality reduction. By incorporating prior knowledge it is argued that it can assist in having less variable and more biologically significant clusters. The clusters were checked against the DAVID database [HSL09b, HSL09a] and the majority of genes in the clusters were related to the cancer to be classified. Information taken from KEGG pathways is a way of decreasing the noise in the microarray experiments. Results were produced using ten different datasets of cancer data, where the goal was to distinguish between different types of cancers. Nine out of the ten datasets are considered to be high dimensional (10935 features).

The results were similar across the different datasets. In the first set of results, it was shown, using the Dunn index, that *a priori* manifold learning is able to create denser clusters with objects that lie closer to the mean of the cluster with a small variance. *A priori* manifold learning produces more compact, well-separated clusters when compared with PCA and the original Isomap. In some cases *a priori* manifold learning performs better only for embeddings with a smaller number of components which is still useful since the embeddings with a lower number of dimensions are considered to be better. There were also cases where the samples and the KEGG signatures were not enough for *a priori* manifold learning to perform better than PCA and Isomap.

A priori manifold learning on average performs better in many cases when using the LDA and SVM classifiers. It does not do so well in classification experiments where PCA and Isomap are computed using the sample-by-sample affinity matrix using the k -NN classifier. In this case there is no significant difference between the three formulations. A possible reason for this is that both LDA and SVM classifiers create a model of the underlying classes, but k -NN is a parametric method which depends on the local distribution of the data, and consequently may be more susceptible to noise.

Overall it can be seen that *a priori* manifold learning produces better formed clusters than either PCA or Isomap, and also performs better in classification experiments using either SVM or LDA methods in most cases. One of the drawbacks of the method is that it has only been formulated using the gene-by-gene affinity matrix, and this makes it more prone to noise than methods that can be computed directly on a sample-by-sample affinity matrix.

Incorporating prior knowledge using KEGG pathways is not only limited to cancer data but it can be applied to a number of diseases that have KEGG signatures. This, along with the fact that the method does not require any other information, makes it easy to adapt to any kind of biological problem. Other studies [CX04, KZ10, CCM⁺04] have used Gene Ontology (GO) terms instead of KEGG pathways. KEGG pathways carry more information when it comes to diseases rather than GO terms since GO terms mostly give information about the function of a gene and the annotation of the different gene products. GO terms can be structured as graphs that show relationships between the different GO terms which are separated in three different categories: (i) biological processes, (ii) functions and (iii) cellular components. None of these provides any information of the gene interactions. They only provide information on the different gene product properties. An implementation of *a priori* manifold learning using GO terms is discussed in the next chapter.

Chapter 6

A Priori Manifold Learning Using GO Terms

6.1 Introduction

A priori manifold learning was also implemented using Gene Ontology (GO) terms. Gene Ontology provides a common language to describe the various gene products and allows comparisons between products of different species. It provides a mark-up language for i) maintaining a controlled vocabulary of gene and gene product attributes; ii) annotating genes and gene products; and iii) providing tools to access the data in the Gene Ontology database and enable functional interpretation of experimental data.

The gene products are described in terms of their associated biological processes, cellular components and molecular functions. Biological processes are essential for a living organism (metabolism), cellular components are substances of which cells are composed (membranes, organelles, proteins) and molecular functions describe activities that occur at the molecular level (catalytic and binding activities). A gene product and its GO terms are shown in figure 6.1.

A table showing gene HAAO with its associated GO Terms is shown in table 6.1. The GO

Gene product: Actin, alpha cardiac muscle 1
GO term: heart contraction ; GO:0060047 (biological process)
Evidence code: Inferred from Mutant Phenotype (IMP)
Reference: PMID 17611253
Assigned by: UniProtKB, June 6, 2008

Figure 6.1: Gene Product Association with GO Term.

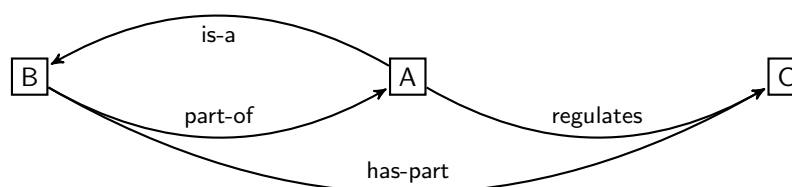


Figure 6.2: **GO Terms Relationships:** GO terms relationships can be: is-a (is a subtype of); part-of; has-part; regulates (negatively, positively).

terms are structured as directed acyclic graphs, and each term has defined relationships to one or more other terms in the same domain, and sometimes to other domains. The relationships are shown in figure 6.2 and are described as:

- **is-a:** A is a subtype of node B; e.g. Mitotic cell cycle is-a cell cycle
- **part-of:** if B exists it is always a part of A; e.g. Replication fork is necessarily part-of a chromosome
- **has-part:** B always has C as a part, if B exists, C will always exist; e.g. Nucleus necessarily has-part chromosome
- **regulates:** A affects C; e.g. Regulation of mitotic spindle organization regulates mitotic spindle organization

Gene Name / Product	Biological Process	Cellular Component	Molecular Function
HAAO 3-hydroxyanthranilate 3,4-dioxygenase	tryptophan catabolic process (GO:0006569)	cytosol (GO:0005829) extracellular exosome (GO:0070062)	3-hydroxyanthranilate 3,4-dioxygenase activity (GO:0000334)
	response to zinc ion (GO:0010043)		protein binding (GO:0005515)
	quinolinate biosynthetic process(GO:0019805)		ferrous iron binding (GO:0008198)
			electron carrier activity (GO:0009055)

Table 6.1: **Gene HAAO Mapped to GO Terms**

6.2 GO Terms Similarity Measures

The *a priori* manifold learning was implemented using the Jaccard coefficient as a similarity measure. For GO terms however, there are alternative similarity measures that have been proposed to make the comparison of GO terms more accurate.

Gene Ontology (GO) is widely used as the basis for measuring the functional similarity of genes. Different similarity measures are split in two categories: Information content-based methods and Graph-based methods.

6.2.1 Information Content-Based Methods

These methods depend on the frequencies of the GO terms being compared and the frequencies of their closest common ancestor term in a specific corpus of GO annotations (biological processes, cellular components, molecular functions).

Resnik [Res95], Jiang and Conrath [JC97], Lin [Lin98] and Schlicker [SDRL06] similarities are based on information theory. The relative frequency of a GO term t is defined as the sum of all the times t is found in the set of GO terms over the total number of terms. Since the probability of a GO term appearing depends on all the concepts it includes, it means that the closer it is to the root, the more frequent it is. As frequency increases the informativeness decreases. Therefore:

$$IC = -\log(p(t)) \quad (6.1)$$

where IC is the Information Content and $p(t)$ is the frequency of a GO term t as shown in equation 6.2.

$$p(t) = \frac{n_t}{N} \quad (6.2)$$

where n_t is the number of occurrences of t and its children in the GO term corpus with N elements. IC can take values between 0 to 1. 0 means closer to the root therefore it is uninformative and 1 means it is on a distant leaf.

Resnik defined similarity as:

$$sim_{Resnik}(t_i, t_j) = IC(MICA) \quad (6.3)$$

where MICA is their most informative common ancestor. Resnik's measure does not consider how distant the terms are from their common ancestor. To take that distance into account, Lin's, Schlicker's and Jiang and Conrath's measures relate the IC of the MICA to the IC of the terms being compared.

Lin's similarity measure is a normalised version of Resnik's similarity that takes into consideration the Information Content of both terms:

$$sim_{Lin}(t_i, t_j) = \frac{2 \times sim_{Resnik}}{IC(t_i) + IC(t_j)} \quad (6.4)$$

Jiang and Conrath's method is a distance metric (semantic distance) between two GO terms.

$$sim_{Jiang}(t_i, t_j) = \frac{1}{IC(t_i) + IC(t_j) - 2 \times IC(MICA) + 1} \quad (6.5)$$

The similarity measure derived from the distance metric is shown below:

$$sim_{Jiang}(t_i, t_j) = 1 - \min(1, IC(t_i) + IC(t_j) - 2 \times IC(MICA)) \quad (6.6)$$

Lin and Jiang and Conrath measures are displaced from the graph, which means that these measures are proportional to the IC differences between the terms and their MICA, independently of the absolute IC of the ancestor. To overcome this limitation, Schlicker's measure was proposed, based on Lin's similarity, which uses the probability of selecting the MICA,

as a weighting factor to provide graph placement.

Schlicker's (or Relevance) similarity measure is a combination of Lin's and Resnik's measures:

$$sim_{Rel}(t_i, t_j) = \frac{2 \times IC(MICA) \times (1 - p(MICA))}{IC(t_i) + IC(t_j)} \quad (6.7)$$

6.2.2 Graph-Based Method

Graph-based methods use the topology of the directed acyclic graph (DAG) of the GO terms in order to calculate the semantic similarity. The DAG of a GO term t_i can be represented as $DAG_{t_i} = (t_i, T_{t_i}, E_{t_i})$ where T_{t_i} is the set of GO terms in DAG_{t_i} including t_i and its ancestors and E_{t_i} are the edges in DAG_{t_i} .

Wang [WABD04] similarity measure works by first defining the semantic value of term t_i as the combined contribution of all terms in DAG_{t_i} with terms closer to t_i contributing more to its semantics. For any term in DAG_{t_i} its contribution (S-value) is defined as $S_{t_i}(t_k)$ where $t_k \in DAG_{t_i}$ is defined as:

$$\begin{cases} S_{t_i}(t_i) = 1 \\ S_{t_i}(t_k) = \max(w_e \times S_{t_i}(t') | t' \in children(t_k) \text{ and } t_k \neq t_i) \end{cases}$$

where w_e is the semantic factor of the edge $e \in E_{t_i}$ between the term t_k and its child t' . w_e can be the type of GO relationship the two terms have (e.g is-a relationship or part-of relationship) which means that w_e can have different weights based on the relationship type. The sum of all terms in S_{t_i} in DAG_{t_i} is defined as:

$$SV_{t_i} = \sum_{t \in T_{t_i}} S_{t_i}(t) \quad (6.8)$$

Therefore for two terms t_i and t_j the Wang similarity is calculated as:

$$S_{t_i, t_j} = \frac{\sum_{t \in T_{t_i} \cap T_{t_j}} (S_{t_i}(t) + S_{t_j}(t))}{SV_{t_i} + SV_{t_j}} \quad (6.9)$$

where $S_{t_i}(t)$ is the S-value of GO term t related to term t_i and $S_{t_j}(t)$ is the S-value of GO term t related to term t_j . Wang's similarity, measures the semantic similarity of two GO terms based on their locations on the GO graph and their relations with their ancestors.

6.3 Results Using The Jaccard Coefficient

Following the same procedure as before *a priori* manifold learning was applied on the same cancer datasets. The Dunn index was used evaluating the density and the structure of the clusters. *K*-Nearest Neighbours (*k*-NN), Support Vector Machines (SVMs) and Linear Discriminant Analysis (LDA) classifiers were applied along with 10-fold cross validation to test the accuracy of the model.

6.3.1 Internal Evaluation

Dunn Index

For the Dunn index the graphs compared to PCA and Isomap are shown in figures 6.3 for the sample-by-sample transformation and for the gene-by-gene transformation in figure 6.4. The Dunn index of the clusters is higher than PCA and Isomap in four occasions for the sample-by-sample transformations (Endometrial, Kidney, Prostate and Uterine) and seven occasions for the gene-by-gene transformations (all but Breast and Colon).

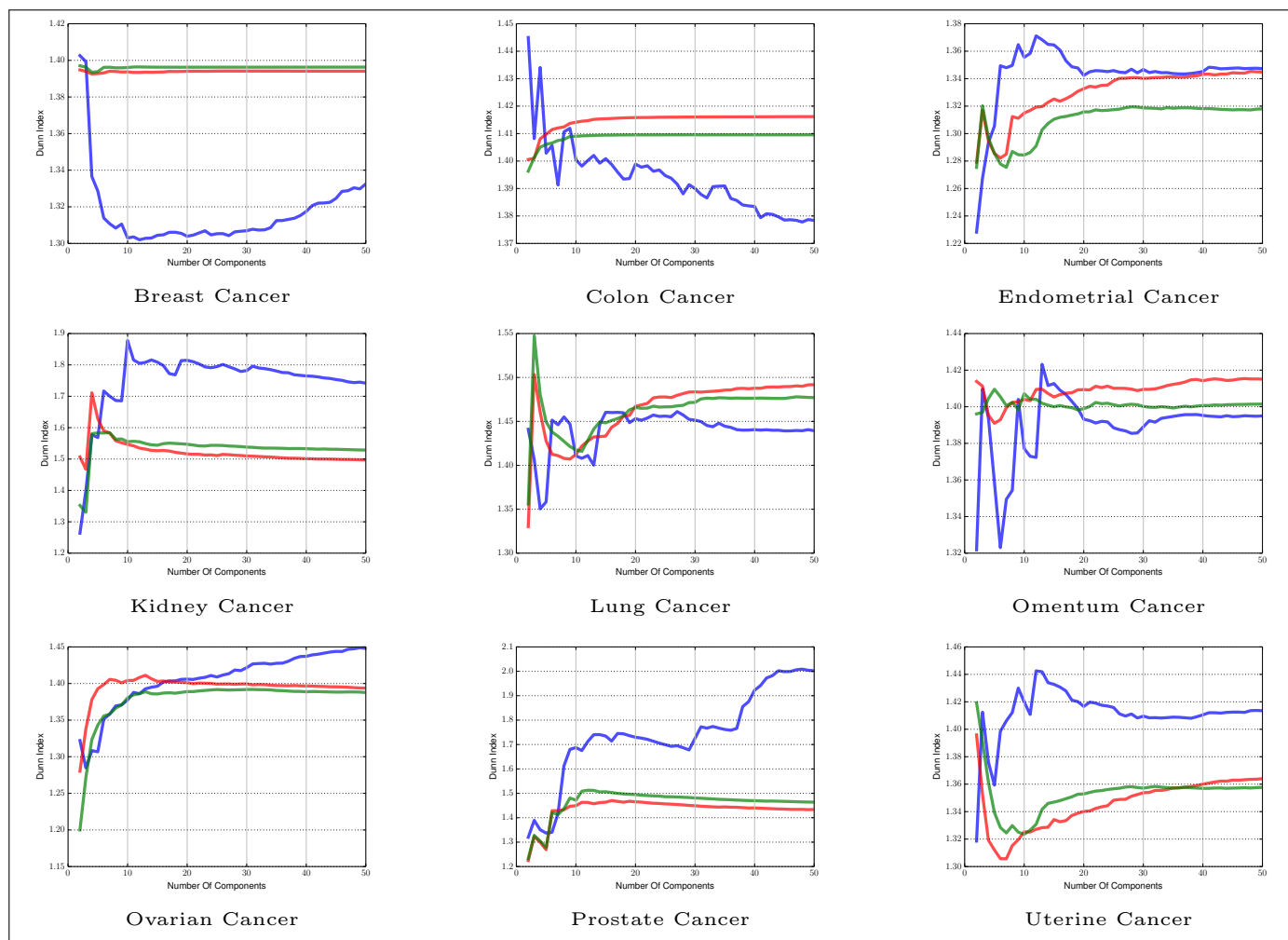


Figure 6.3: **Dunn Index for cancer classification results:** The Dunn Index found using *a priori* manifold learning with GO terms (Blue) compared with PCA (Green) and Isomap (Red). PCA and ISOMAP were computed using features from the sample-by-sample affinity matrix.

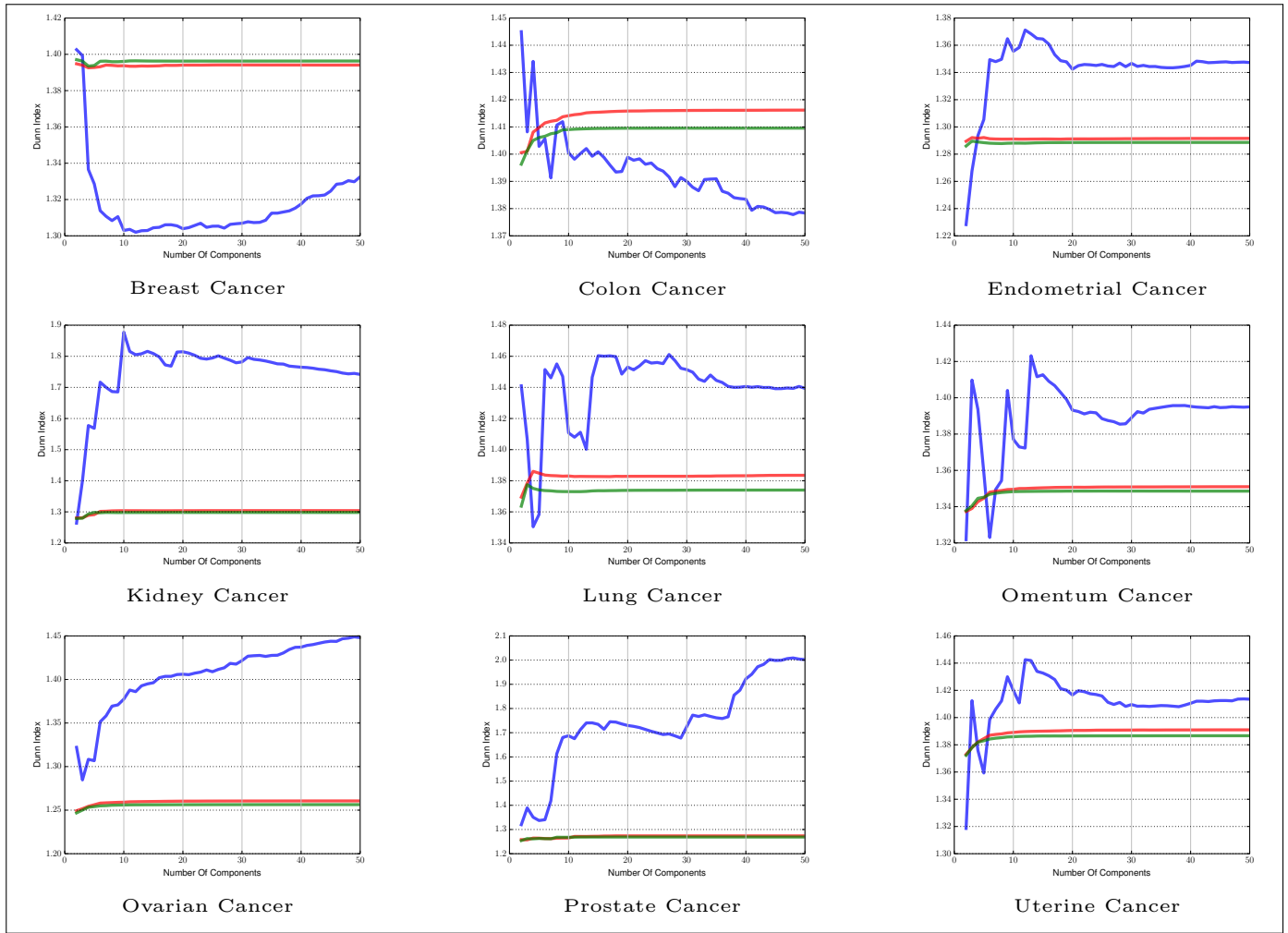


Figure 6.4: **Dunn Index for cancer classification results:** The Dunn Index found using *a priori* manifold learning with GO terms (Blue) compared with PCA (Green) and Isomap (Red). PCA and ISOMAP were computed using features from the gene-by-gene affinity matrix

6.3.2 Ten Fold Cross-Validation

When applying cross-validation using k -NN, SVMs and LDA it was observed that the models for some of the cancers over-fit, since the cross-validation variance was too high. This is shown in table 6.2 for k -NN, in table 6.5 for SVMs and in table 6.6 for LDA. The results when compared with PCA and Isomap using k -NN are shown in table 6.3 and table 6.4 for genes-by-gene and sample-by-sample transformations respectively. For LDA the results are shown in table 6.7 for genes-by-gene and table 6.8 for sample-by-sample.

Type Of Cancer	A Priori Manifold Learning (GO)	Variance
Breast cancer	0.892	0.0008
Colon cancer	0.937	0.0001
Kidney cancer	0.971	0.0001
Ovary cancer	0.873	2.501e-05
Lung cancer	0.924	3.419e-05
Uterus cancer	0.919	8.136e-06
Omentum cancer	0.950	7.466e-06
Prostate cancer	0.995	4.034e-05
Endometrium cancer	0.961	3.247e-06

Table 6.2: **10 Fold Cross Validation Accuracy using k -Nearest Neighbours:** The results of the 10-fold cross-validation on the dataset show that some of the models over-fit (shown in bold).

Type Of Cancer	<i>A priori</i> manifold learning (GO)	Isomap	PCA	A Priori Manifold Learning (KEGG)
Breast cancer	0.892	0.782	0.792	0.806
Colon cancer	0.937	0.834	0.834	0.868
Kidney cancer	0.971	0.900	0.903	0.937
Ovary cancer	0.873	0.834	0.838	0.841
Lung cancer	0.924	0.883	0.886	0.902
Uterus cancer	0.919	0.882	0.881	0.891
Omentum cancer	0.950	0.912	0.912	0.914
Prostate cancer	0.995	0.943	0.945	0.955
Endometrium cancer	0.961	0.922	0.922	0.923

Table 6.3: **Cancer classification accuracy measured using the k -Nearest Neighbours algorithm and Ten Fold Cross validation:** The results for PCA and ICA were computed using features from gene-by-gene affinity matrices. The *a priori* manifold learning algorithm using GO terms outperforms the other methods in most cases (shown in bold).

Type Of Cancer	A Priori Manifold Learning (GO)	Isomap	PCA	A Priori Manifold Learning (KEGG)
Breast cancer	0.892	0.863	0.879	0.806
Colon cancer	0.937	0.897	0.906	0.868
Kidney cancer	0.971	0.931	0.932	0.937
Ovary cancer	0.873	0.842	0.851	0.841
Lung cancer	0.924	0.911	0.917	0.902
Uterus cancer	0.919	0.890	0.891	0.891
Omentum cancer	0.950	0.912	0.912	0.914
Prostate cancer	0.995	0.954	0.954	0.955
Endometrium cancer	0.961	0.924	0.926	0.923

Table 6.4: **Cancer classification accuracy measured using the k -Nearest Neighbours algorithm and Ten Fold Cross validation:** The results for PCA and ICA were computed using features from sample-by-sample affinity matrices. The a priori manifold learning algorithm using GO terms outperforms the other methods in most cases (shown in bold).

Type Of Cancer	A Priori Manifold Learning (GO)	Variance
Breast cancer	0.908	9.034e-05
Colon cancer	0.867	0.001
Kidney cancer	0.964	0.0001
Ovary cancer	0.836	0.002
Lung cancer	0.926	0.0003
Uterus cancer	0.746	0.049
Omentum cancer	0.896	0.001
Prostate cancer	0.988	4.106e-05
Endometrium cancer	0.905	0.001

Table 6.5: **10 Fold Cross Validation Accuracy using Support Vector Machines:** The results of the 10-fold cross-validation on the dataset show that some of the models over-fit (shown in bold).

Type Of Cancer	A Priori Manifold Learning (GO)	Variance
Breast cancer	0.955	5.709e-05
Colon cancer	0.963	7.840e-05
Kidney cancer	0.982	2.566e-05
Ovary cancer	0.885	1.005e-05
Lung cancer	0.967	7.038e-05
Uterus cancer	0.913	0.0001
Omentum cancer	0.930	7.164e-05
Prostate cancer	0.997	8.379e-07
Endometrium cancer	0.916	0.0004

Table 6.6: **10 Fold Cross Validation Accuracy using Linear Discriminant Analysis:** The results of the 10-fold cross-validation on the dataset show that some of the models over-fit (shown in bold).

Type Of Cancer	<i>A priori</i> manifold learning (GO)	Isomap	PCA	A Priori Manifold Learning (KEGG)
Breast cancer	0.955	0.888	0.910	0.890
Colon cancer	0.963	0.914	0.924	0.906
Kidney cancer	0.982	0.911	0.954	0.956
Ovary cancer	0.885	0.945	0.870	0.871
Lung cancer	0.967	0.924	0.940	0.935
Uterus cancer	0.913	0.901	0.905	0.906
Omentum cancer	0.930	0.926	0.923	0.927
Prostate cancer	0.997	0.970	0.972	0.973
Endometrium cancer	0.916	0.932	0.930	0.937

Table 6.7: **Cancer classification accuracy measured using Linear Discriminant Analysis and ten fold cross validation:** The results for PCA and ICA were computed using features from gene-by-gene affinity matrices. *A priori* manifold learning using GO terms outperforms the other methods in most cases (shown in bold).

Type Of Cancer	<i>A priori</i> manifold learning (GO)	Isomap	PCA	A Priori Manifold Learning (KEGG)
Breast cancer	0.955	0.901	0.912	0.890
Colon cancer	0.963	0.914	0.925	0.906
Kidney cancer	0.982	0.952	0.953	0.956
Ovary cancer	0.885	0.867	0.870	0.871
Lung cancer	0.967	0.938	0.941	0.935
Uterus cancer	0.913	0.900	0.905	0.906
Omentum cancer	0.930	0.923	0.924	0.927
Prostate cancer	0.997	0.972	0.972	0.973
Endometrium cancer	0.916	0.934	0.930	0.937

Table 6.8: **Cancer classification accuracy measured using Linear Discriminant Analysis and ten fold cross validation:** The results for PCA and ICA were computed using features from sample-by-sample affinity matrices. *A priori* manifold learning using GO terms outperforms the other methods in most cases (shown in bold).

Receiver Operating Characteristic Curves

The ROC curves are shown in the appendix C, even though the models are not considered reliable since they over-fit. A lot of models seem to over-fit for all the classifiers used. This makes the cross-validation accuracy and the ROC curves (true positives and false negatives) inconsistent since the variance was very high. The accuracy for the models that do not over-fit was higher than all the scores obtained for PCA and Isomap by both gene-by-gene and sample-by-sample transformations. Only one cancer model does not over-fit by any classifier and that is Prostate cancer.

Over-fitting

It is clear from the information obtained from the cross-validation that the test error is much higher than training error for most of the datasets. This is a way of knowing that the models do over-fit and they cannot be trusted as an accurate representation of the data [GS03]. Since the algorithm is using the entire dataset it is expected that the variance of the dataset should be similar to the variance of the model which is not the case here. The

high variance observed shows the fluctuation in the predictions of the different folds during cross-validation.

6.4 Results Using Wang Similarity

The Jaccard coefficient can be replaced with the Wang or the Lin similarity for GO terms which can make the similarity score a lot more accurate. Unfortunately, the running time for the datasets was far too high, and it was impossible to get any results in a reasonable time.

When calculating the distance matrices and the GO term similarities there were more than 100 million comparisons to be made due to the size of the dataset (10935×10935). Every gene can be associated with a number of GO terms and both Wang and Lin work by pairwise comparisons of those genes. That means that the number of comparisons per gene (k) combination is $gComb = k \times k$ (worst case). These algorithms work by finding the ancestors of the two terms and calculating the information content which is $O(n^2)$ (average case) in terms of computational complexity. The complexity is super-linear and it grows as the size of the problem grows (if the size doubles, the time for solving the problem more than doubles). The total average complexity is $\approx O(gComb \times n^2)$.

In addition the Wang similarity creates a DAG which it then traverses in order to find the ancestors and the children of each ancestor. It constructs the DAG, finds the common ancestors between two genes and calculates the semantic contribution for all ancestors. If genes A and B have k ancestors in common then the worse case complexity will be $2 \times k \times n^2$ since for genes A and B two different DAGs are constructed and the computation for each takes place separately. This is again done in pairwise combinations $\approx O(gComb \times k \times n^2)$

It takes about 1 minute per comparison to get the similarity score for each pair of GO terms when using Wang. It will take approximately 83000 days to finish the calculation. The Lin similarity measure is slightly faster than the Wang similarity; it takes about 45 seconds per comparison which will take about 62000 days to finish.

Even when only one GO term per gene is selected at random, the algorithms are still too slow - using Lin similarity will take approximately 208 days to finish while Wang will need 1383.

6.4.1 Comparison With A Priori Manifold Learning Using Pathways

The models created with *a priori* manifold learning do not show any signs of over-fitting. That makes the pathway results more reliable than the GO terms results. In addition, for two of the models created using GO terms the η value was zero or near zero which means that the GO information was not taken into account significantly or not at all. This shows that for Prostate and Uterine cancer the models generated were just like the original Isomap algorithm; with the only difference being that instead of Euclidean distance, Mahalanobis distance is used. This explains the improvement in accuracy seen in some of the cases. The Dunn index for the pathways is much higher than the GO terms' for all cancers, which implies that the quality of the clusters is better when using pathways as a source of prior knowledge.

6.5 Discussion

In this chapter a modification to *a priori* manifold learning was presented in order to evaluate the use of GO terms as prior knowledge. Two different ways were shown for the calculation of the similarity measure for each pair of GO terms. The first one uses Jaccard coefficient as before and the other uses the Wang and Lin GO term similarity measures. The second way however was infeasible due to computational limitations even though it would be a more accurate measure of GO terms similarity. Using the Jaccard coefficient the results obtained were not reliable since the models for the different cancers over-fit. All cancers (except Prostate) over-fit (as shown by the variance of the models using LDA, SVMs and k -NNs) and therefore the results were not accurate. This is due to the abstract nature of GO terms, which are used as a way of describing gene products rather than providing infor-

mation on relationships between genes, which pathway data provide. A gene is associated with a lot more GO terms than pathways. GO terms have 3 different categories: biological processes, cellular components and molecular functions; and different types of relationships: is-a, part-of, has-part, and regulates; which is information about the annotation of the different gene products; while pathway information describes interactions among those genes. The same annotation i.e. tryptophan catabolic process; does not imply any sort of interaction, even though a number of genes can end up having the same annotation. For this reason, it seems that even genes that are not strongly related tend to have very high Jaccard coefficients which leads to a wrong distance matrix. Getting a higher coefficient for genes with the same annotation could mean that the model captures the noise instead of the actual information of the dataset.

When constructing the *a priori* manifold learning embeddings it was apparent that the best model for prostate cancer was the one where the GO terms were not taken into account at all and therefore the η value was 0. For uterine cancer the model's η value was only 800 which means that the GO terms were taken into account only slightly. The difference in the accuracy between an η of 800 and an η value of 0 for uterine cancer is only 0.01. In addition when using different models, created by *a priori* manifold learning using GO terms (with a different η value), it was observed that the variance could be much lower but the accuracy was also very low (≈ 0.7) for all the cancers. There was no good balance between variance and accuracy. This is known as the bias–variance trade-off. For the models where the variance was low the η value was near 0 for all cases.

Chapter 7

Identifying Significant Features In Cancer Methylation Data Using Gene Pathway Segmentation

7.1 Introduction

The latest worldwide cancer statistics, provided by GLOBOCAN 2012, have shown that approximately 14.1 million people suffered from cancer in 2012. The number is expected to rise to 24 million in 20 years time¹. Some advances have been made in the identification of genes related to the cancer aetiology. All these have led to the expansion of our understanding of the genetic mechanisms that are driving cancer progression.

Recent improvements in molecular biology technology have allowed the measurement and profiling of DNA methylation sites in large genomic samples [SKK⁺06]. DNA methylation is believed to be closely related to gene expression [ASH13, Boc12] and DNA methylation sites have been increasingly found to be related to the processes of cancer [Lau08, LM12]. Methylation biomarkers have also been associated with the response of a patient to particular treatment of cancer as shown in some clinical studies [MDH⁺05, Bay05]. Machine

¹<http://www.cancerresearchuk.org>

learning has been widely used on biological data with increasing success [WTH⁺05, OS10, LSC⁺09]. Methylation data however has only recently been analysed using machine learning [RJG⁺12].

Due to the high dimensionality of the methylation datasets, direct use of many machine learning methods is computationally intractable, and it is necessary to find a method of selecting significant features to analyse. Current methylation data may have in excess of 450,000 probes (n), and perhaps 100 patient cases (D). The computational complexity of PCA – the simplest and most fundamental method in multivariate data analysis is at least $n \times n \times D$, which can make the analysis infeasible. To overcome this, a new feature selection approach is proposed in which methylation data is combined with prior knowledge taken from biological pathways. Prior knowledge has been used before for the classification of microarray expression [BGL⁺00, GHHZ09, HTG14, CX04, KZ10, CCM⁺04, CW09b]. Little has been done in terms of predicting response to cancer treatment using methylation data and prior knowledge. For this experiment pathway information from the ConsensusPath database [KSLH13, KPG⁺11, KWLH09, POH⁺10] was used. The dataset was split into pathway sets and then each pathway set was analysed individually in order to see which sets can provide accurate classification. The ConsensusPath database integrates different types of information including: protein interactions, genetic interactions signalling, metabolism, gene regulation and drug target interactions in humans. These are taken from a number of databases including Reactome, KEGG, HumanCyc, PID and BioCarta. AdaBoost with decision trees as weak classifiers was applied on the sets inferred from the ConsensusPath database in order to classify response to treatment. Boosting techniques can help with reducing the bias in supervised learning by being less susceptible to the over-fitting problem than other learning algorithms [Kea98].

7.2 Feature Selection Over Pathway Segmentation Algorithm

In order to reduce the computation time and improve the accuracy each dataset was split into a number of subsets using information in the ConsensusPath database [KSLH13, KPG⁺11, KWLH09, POH⁺10]. Genes that belong in the same pathway were put in the same gene set. As a result the datasets were divided into 3213 smaller sets mostly between 100 and 2000 genes each. A gene can belong to more than one pathway. The method is shown in figure 7.1 and works as follows:

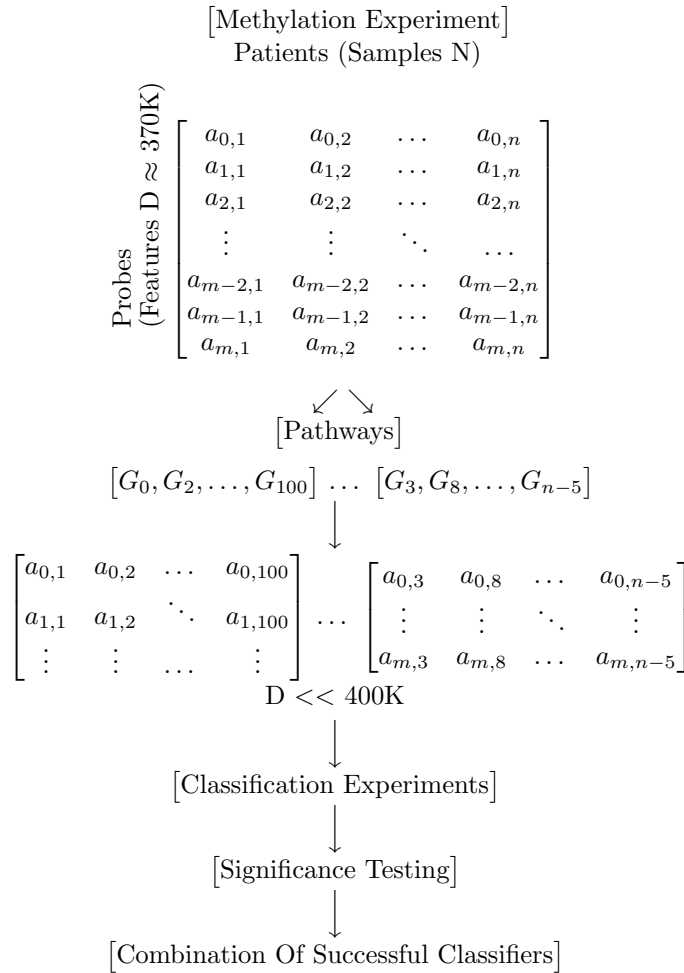


Figure 7.1: **Pathway Algorithm:** As a first step the original methylation dataset, is split into smaller sets based on pathway information taken from the ConsensusPath database. AdaBoost is applied on the smaller sets using stratified cross-validation (to account for unbalanced classes), and the accuracy score of each pathway is calculated. A number of random sets are created so that the pathway sets with the highest score could be verified using z-score and p-values.

- Split the dataset into subsets defined by pathways;
- Apply AdaBoost for classifying the response to treatment in each subset;
- Find the subsets that can predict progression accurately using z-scores and p-values;
- Recombine the successful pathway sets using the accuracy of the tree and the Gini importance of its features to construct a set of features that can accurately classify the dataset.

The relation between the different probe locations relative to each gene (island, shore, shelf) is not taken into account, instead the focus was on gene pathway sets that can influence response. For the probes to gene mapping two different methods were attempted:

1. Normalising the probes that belong to the same gene by dividing their methylation values by their mean and picking the probe with the highest intensity;
2. Using all the probes in the classifier.

Both methods identified the same sets of genes since probes that match the same gene are correlated [SH07] and decision trees use correlated features interchangeably. If the tree becomes repetitive it gets pruned [TL11] so leaving all the probes in can be much faster.

7.3 Datasets

For our feature selection over pathway segmentation algorithm two methylation dataset (Chronic myeloid leukaemia (CML) and TCGA Lower Grade Glioma (LGG)) were used. In the CML methylation dataset there were 429 231 probes with 91 samples, 60 of them were responsive to blood cancer treatment while 31 were not and in the LGG (<http://cancergenome.nih.gov/>) there were 370 203 probes with 82 samples, 57 of them were not responsive to treatment (Progressive Disease) while 25 were (Complete Remission/Response).

Information on the contents of the datasets is shown in table 7.1.

Type Of Cancer	Number Of Samples	Number Of Genes
Chronic Myeloid Leukaemia	60 responsive vs 31 non-responsive	429231
TCGA Lower Grade Glioma	25 responsive vs 57 non-responsive	370203

Table 7.1: **Datasets Used**

Lower Grade Glioma

Primary brain tumours emerge from various cells that make up the brain and central nervous system and are named after the cell they originate from. Approximately 42% patients survive for one year. This rate however drops to 14% $\approx$ 16% after five years and 9% after ten years. There are many types of brain cancer (meningioma, medulloblastoma, gliomas) and different types of gliomas: Astrocytomas (originating from astrocyte cells) which include: low-grade astrocytomas, anaplastic astrocytoma and glioblastoma multiforme; oligodendrogliomas (originating from oligodendrocytes); ependymoma (originating from ependymal cells) [BG14].

Chronic Myeloid Leukaemia

Leukaemia is the 11th most common cancer for both sexes and it accounts for 2.5% of all cancers with $\approx$ 352 000 new cases worldwide every year². The exact cause of leukaemia is not yet known but it is believed that is related to both environmental and inherited factors. This type of cancer originates in the bone marrow and results in a high number of leukaemia cells (abnormal white blood cells which are not fully developed) [Nat13]. Approximately 63% $\approx$ 64% patients survive for one year. This rate however drops to 44% after five years and 32% $\approx$ 33% after ten years. There are four major types of leukaemia:

1. Acute Lymphoblastic Leukaemia (ALL): Starts in abnormal lymphoid stem cells and progresses very quickly [Wei14].

²<http://www.cancerresearchuk.org/>

2. Acute Myelogenous Leukaemia (AML): Starts in abnormal myeloid stem cells and develops quickly [Sei14].
3. Chronic Lymphocytic Leukaemia (CLL): Starts in abnormal lymphoid stem cells and take months or even years to develop [Cle79].
4. Chronic Myelogenous Leukaemia (CML): Starts in abnormal myeloid stem cells and develops slowly [Bes14].

AML, CLL and CML are mainly adult cancers. They are very rarely encountered in children. ALL however is a cancer that is very common in children. Acute leukaemia usually develops quickly and worsens in some weeks unless treated, in contrast to chronic forms of leukaemia that progress very slowly and can be left untreated for months or years. Even though a lot of progress has been made over the years for the treatment of CML leukaemia, 30% to 35% of patients do not respond to treatment [CBD⁺13]. A drug called Imatinib, a tyrosine-kinase inhibitor, is the first line of treatment for CML and was introduced in 1988 [COJT⁺97, SS11]. The Chronic Myelogenous Leukaemia Methylation Dataset is unpublished.

7.4 Experiments Using The Pathway Algorithm

7.4.1 Baseline Experiments

Initially the complete datasets for LGG and CML were analysed using principal components analysis (PCA) [Pea01] for linear dimensionality reduction and manifold Isomap [TdSL00] for non-linear feature extraction. Sample-by-sample affinity matrices were used since gene-by-gene matrices were very large and even after several hours of computation did not yield any result. The results obtained were not significant in terms of accuracy and are shown in table 7.2. The ROC curves are as shown in figure 7.2 for LGG (left) and CML (right).

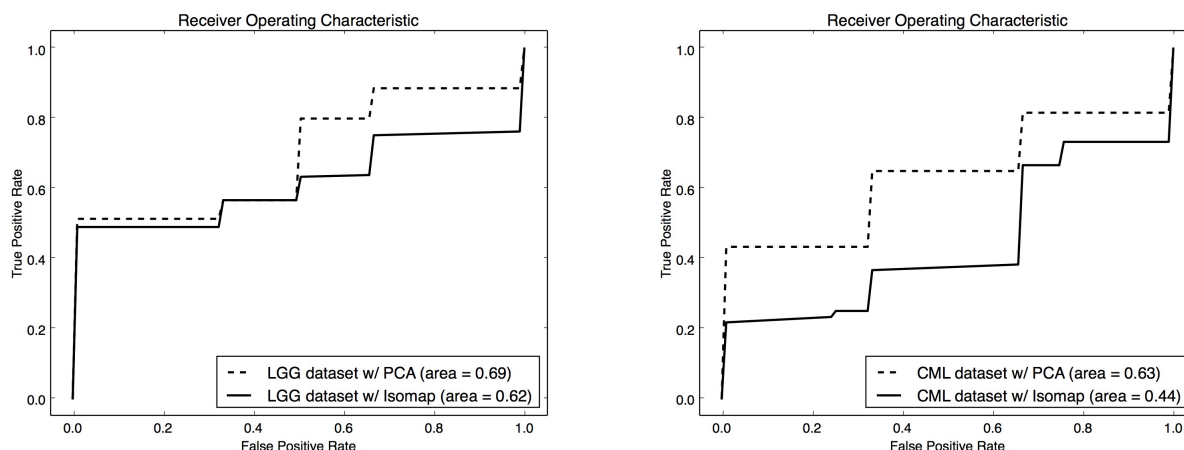


Figure 7.2: **ROC Curve for the original LGG (left) and CML (right) datasets:** Dimensionality reduction was performed using both PCA and Isomap, and classification was made using AdaBoost.

Dataset	Accuracy	Variance
CML with PCA	0.6044	0.0222
CML with Isomap	0.5155	0.0159
LGG with PCA	0.7083	0.0177
LGG with Isomap	0.6347	0.0289

Table 7.2: **Accuracy of the original datasets:** Results for the original datasets using linear (PCA) and non-linear (Isomap) forms of dimensionality reduction and AdaBoost. The results are not significant in terms of accuracy.

7.4.2 TCGA Lower Grade Glioma

Pathways

Classification for LGG treatment response was performed on all pathway sets with a number of genes greater than 100. Four gene pathway sets had accuracy between 0.88 ~ 0.90. To estimate the accuracy 10 fold stratified cross-validation was used, as shown in table 7.3. The p-values show that the random null hypothesis can be rejected and therefore the results are significant in the 99% confidence interval. The Receiver operating characteristic (ROC) curves were plotted for the identified gene pathway sets (figure 7.3). Dimensionality

reduction on those gene pathway sets worsens the results as shown in figure 7.4. The ROC curves for two other gene pathway sets that do not perform so well were compared to the Pantothenate and CoA biosynthesis set (figure 7.5). The accuracy for the gene pathway sets when logistic regression is applied instead of AdaBoost is shown in table 7.4. The results are not significant in terms of accuracy. The AdaBoost method performs better since the decision trees split the dataset several times instead of just once and because boosting methods tend to remove bias from the results by re-sampling.

Pathway Name	Accuracy	Variance	Z-Score	P-Value
Pantothenate and CoA biosynthesis - Homo sapiens (human)	0.904	0.0191	3.9713	0.000036
Transcription factor creb and its extracellular signals	0.891	0.0163	3.68249905	0.000116
Pyrimidine metabolism	0.890	0.0070	3.65034371	0.000131
IL2	0.879	0.0056	3.39268741	0.000346

Table 7.3: **Gene pathway sets with the highest scores for LGG:** Gene pathway sets for LGG and their accuracies and variances after 10 fold Stratified Cross-Validation

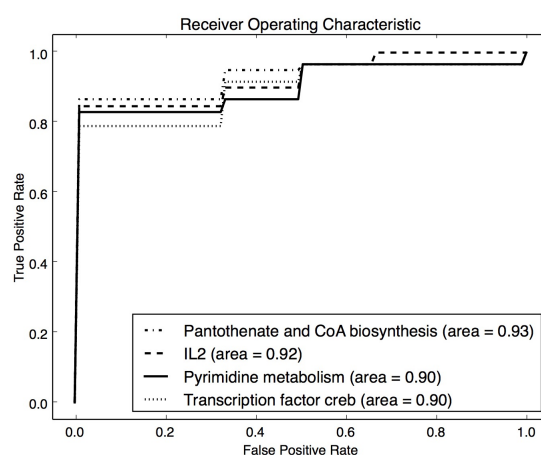


Figure 7.3: **ROC curves for the LGG Pathway Sets:** ROC curves for the four gene pathway sets with the highest accuracy

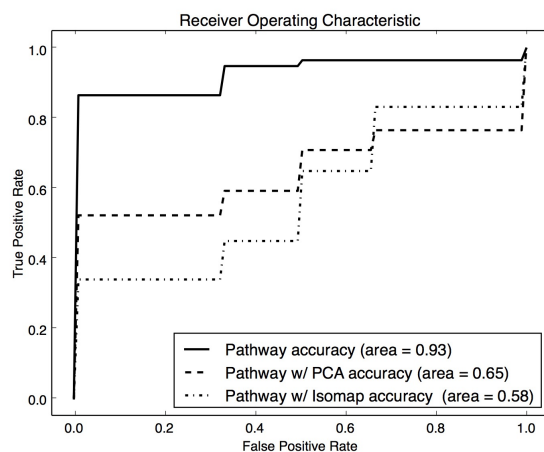


Figure 7.4: **Pantothenate and CoA biosynthesis and Retinoate Biosynthesis II pathway set:** Comparison of the pathway set, with and without dimensionality reduction (PCA and Isomap)

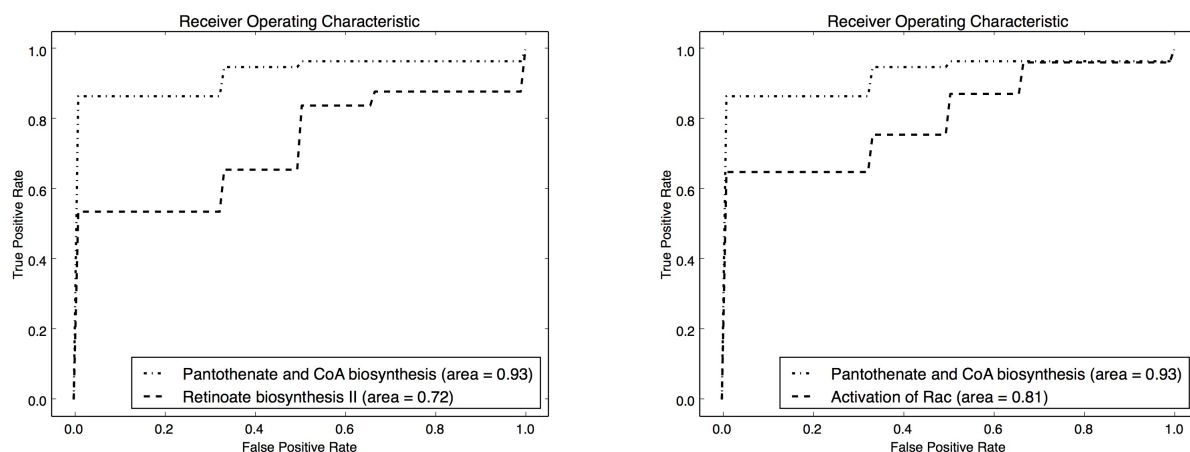


Figure 7.5: **Comparison between Pantothenate and CoA biosynthesis and Retinoate Biosynthesis II (left) and Activation of Rac (right) pathway sets**

Pathway Name	Logistic Regression
Pantothenate and CoA biosynthesis - Homo sapiens (human)	0.708
Transcription factor creb and its extracellular signals	0.697
Pyrimidine metabolism	0.650
IL2	0.674

Table 7.4: **Logistic Regression applied on LGG:** Results for logistic regression on LGG gene pathway sets

Each gene was removed from the four gene pathway sets to see the effect it had on the accuracy. It is shown that some genes have more effect than others and removing them

affects the accuracy negatively. This is shown figures 7.6, 7.7, 7.8. In addition random gene sets were created to verify the effectiveness of the pathways. The class distribution was the same in all random sets since the samples were left intact. The number of genes was different in every set, but the sets generated followed the distribution of the number of genes in the pathway sets. The graphs for the highest-scoring gene pathway sets against random gene sets were constructed. These results are shown in figure 7.9. 980 random sets were used to check the significance of the result. The z-score was used to calculate the difference from the mean of the accuracy for the random sets and also the highest scoring gene pathway sets. The p-values were calculated using the z-score obtained. Comparing the accuracy of all the random sets, after 10 different runs it was observed that the accuracies vary between 0.4 and 0.71. Their variance is either equal or much greater than the variance of the pathway sets. The variance of the random sets is between 0.018 and 0.0287. This shows that the genes belonging to the pathway sets used in this method can play an important role in classifying cancer.

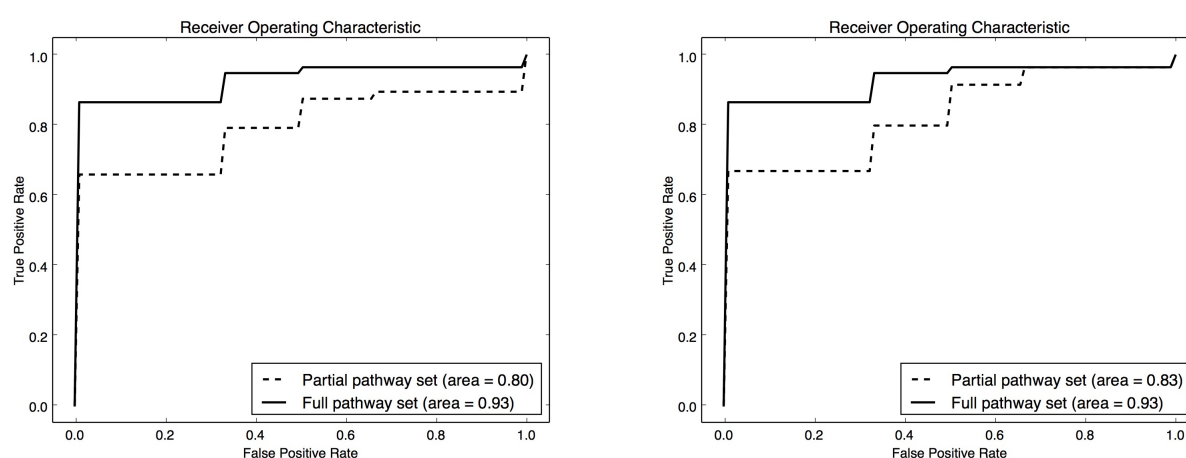


Figure 7.6: 50% of the genes were removed from the IL2 (left) and Pantothenate and CoA biosynthesis (right) pathway sets

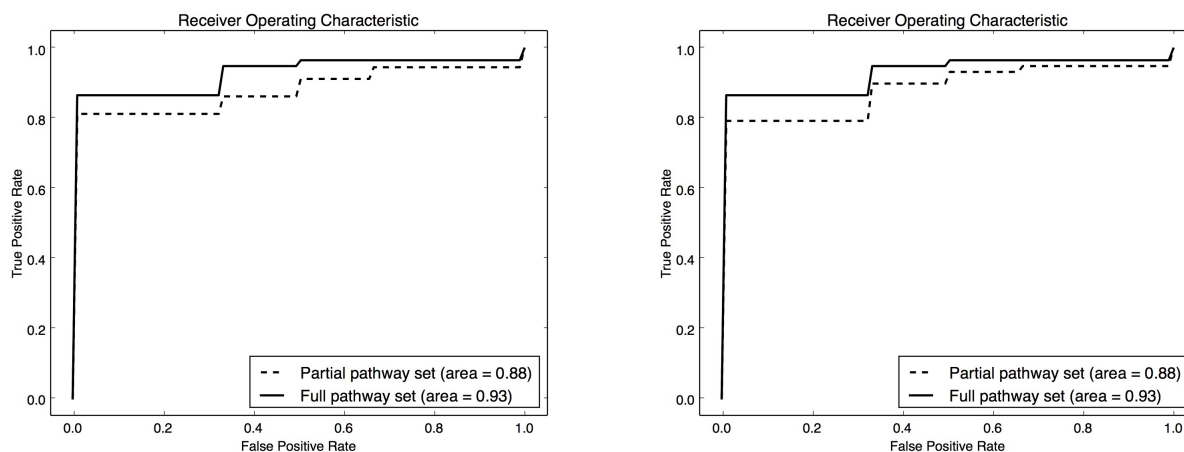


Figure 7.7: 20% of the genes were removed from the IL2 (left) and the Pantothenate and CoA biosynthesis (right) pathway sets

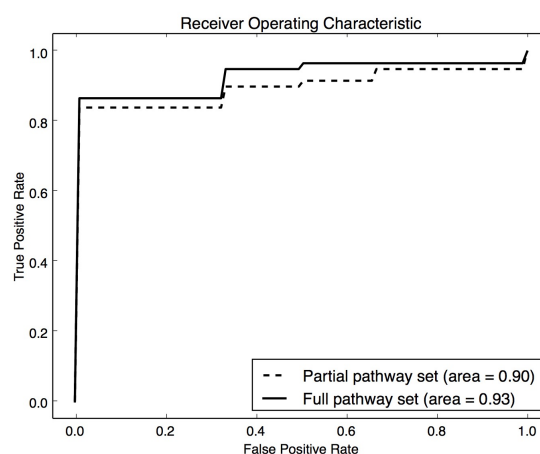


Figure 7.8: 50% of the genes were removed from the IL2 pathway set

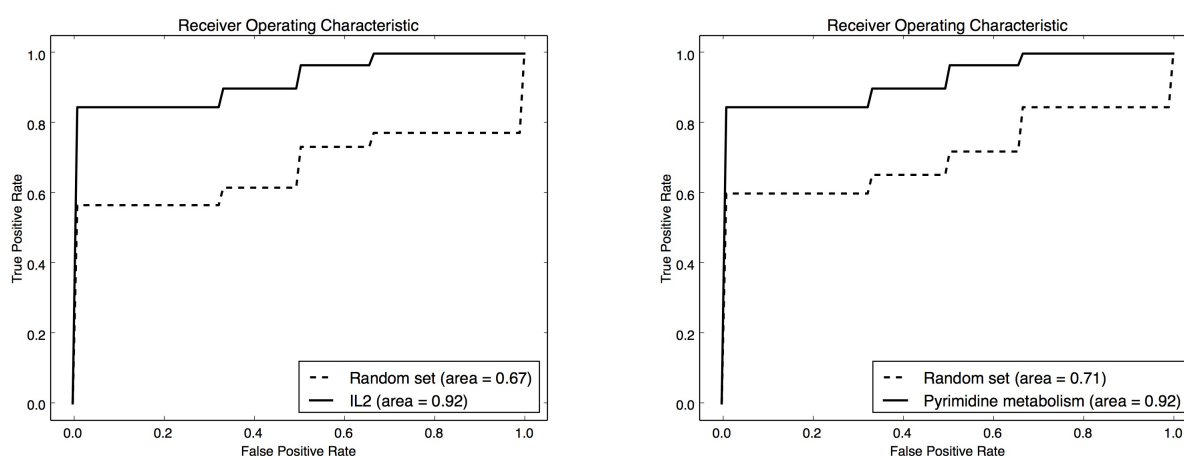


Figure 7.9: Comparison between IL2 and a random gene pathway set of 1170 genes (left) and between Pyrimidine metabolism and a random gene pathway set of 644 genes (right)

Gene Set

A more comprehensive set of genes with discriminative properties was then composed by combining several successful pathway sets that were identified by the algorithm shown in algorithm 7.1. To obtain this set AdaBoost was applied on all gene pathway sets. The gene pathway sets with the highest accuracy were selected. Looking closely at how AdaBoost builds the decision classifiers the features (genes) that were important when building the decision tree were identified and the ones that were not as important were filtered out. An important feature for a decision tree is one for which the weights are higher. It indicates how well the nodes of the decision tree are partitioned by that feature. The importance of a feature is related to its height in the tree with the root being the most important. This is also known as the Gini importance.

Several classifiers with different combinations of threshold values were constructed and the combination with the highest accuracy score was chosen. The algorithm is shown in algorithm 7.1. *AccuracyThreshold* had values between 0.7 – 0.9 and *GiniImportanceThreshold* had different values ranging from 0.003 to 0.5. Table 7.5 shows the resulting set of genes which can classify progression with an accuracy of 99%. Genes that have been previously associated with gliomas appear in bold. The p-value of this set is 0.00140625 showing that it is significant in the 99% confidence interval.

Algorithm 7.1. Gene Selection Algorithm based on accuracy thresholds and how important each feature is when constructing the decision tree³

Data: Methylation Data

for $p \in \text{PathwaySet}$ **do**

if $\text{accuracy}(p) \geq \text{AccuracyThreshold}$ **then**

for $\text{feature} \in \text{DecisionTree}(p)$ **do**

if $\text{importance}(\text{feature}) \geq \text{GiniImportanceThreshold}$ **then**

$\text{GeneSet} \leftarrow \text{feature}$

end if

end for

end if

end for

Gene Name	Functional Annotation
DDOST	Dolichyl-Diphosphooligosaccharide-Protein Glycosyltransferase Subunit
PRKAR2B	protein kinase, cAMP-dependent
PDPK1	3-phosphoinositide dependent protein kinase 1
PIK3CD	phosphatidylinositol-4,5-bisphosphate 3-kinase
CDC16	cell division cycle 16
OAT	ornithine aminotransferase
KRAS	Kirsten Rat Sarcoma Viral Oncogene Homolog
NTRK1	neurotrophic tyrosine kinase, receptor, type 1
NF1	neurofibromin 1
BTRC	beta-transducin repeat containing E3 ubiquitin protein ligase
PIK3R3	phosphoinositide-3-kinase, regulatory subunit 3 (gamma)
KCNMB4	potassium large conductance calcium-activated channel, subfamily M, beta member 4
IFNGR1	interferon gamma receptor 1
SC5DL	sterol-C5-desaturase
ATF2	activating transcription factor 2
GABRB2	gamma-aminobutyric acid (GABA) A receptor, beta 2
STX1A	syntaxin 1A (brain)
GPX4	glutathione peroxidase 4
GAB2	GRB2-associated binding protein 2
EIF2AK1	eukaryotic translation initiation factor 2-alpha kinase 1
SOS1	son of sevenless homolog 1 (Drosophila)
EXOC6	exocyst complex component 6
IRS1	insulin receptor substrate 1
ANK1	ankyrin 1, erythrocytic 2
IL6R	interleukin 6 receptor
NRCAM	neuronal cell adhesion molecule
SLC22A2	solute carrier family 22 (organic cation transporter), member 2
PPCDC	phosphopantothencysteine decarboxylase
UPB1	ureidopropionase, beta
PTK2B	protein tyrosine kinase 2 beta
ITGA2	integrin, alpha 2 (CD49B, alpha 2 subunit of VLA-2 receptor)
STAT3	signal transducer and activator of transcription 3 (acute-phase response factor)
SLCO4A1	solute carrier organic anion transporter family, member 4A1
SLCO2A1	solute carrier organic anion transporter family, member 2A1

Table 7.5: **Gene List for LGG:** Combined Gene List of the most influential genes for LGG

7.4.3 Chronic Myeloid Leukaemia

Pathways

The gene pathway sets with the highest scores for the CML data set are shown in table 7.6. The p-values show that the result is significant in the 99% confidence interval. Classification was performed on all gene pathway sets in order to see how well each of them could classify CML progression. Two gene pathway sets were isolated that had an accuracy of 0.9888. The accuracy of the Regulation of the KIT signalling pathway set was also calculated using two different methods of dimensionality reduction. Dimensionality reduction once again worsens the results as shown figure 7.11.

Pathway Name	Accuracy	Variance	Z-Score	P-Value
Regulation of KIT signaling	0.9888	0.0011	6.44028444	<0.00001
Signaling events mediated by Stem cell factor receptor (c-Kit)	0.9888	0.0011	6.44028444	<0.00001
Superpathway of D-myo-inositol(1,4,5)-trisphosphate metabolism	0.8244	0.0176	2.11346295	0.0176

Table 7.6: **Gene pathway sets with the highest scores for CML:** Gene pathway sets for CML and their accuracies and variances after 10 Fold Stratified Cross-Validation

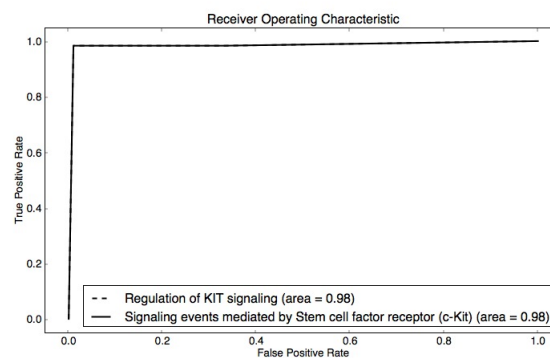


Figure 7.10: **ROC curves for the CML Pathway Sets:** ROC curves for the CML Pathway Sets for the two gene pathway sets with the highest accuracy

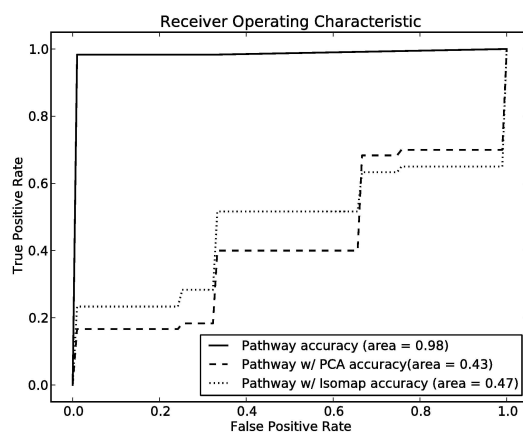


Figure 7.11: **Regulation of KIT signalling pathway set:** Comparison of the pathway set, with and without dimensionality reduction (PCA and Isomap)

The Receiver Operating Characteristic (ROC) curves for the two gene pathway sets with the highest accuracies were plotted (figure 7.10). The ROC curves for two other gene pathway sets that do not perform so well compared with the Regulation of KIT signalling set are shown (figures 7.12 and 7.13). The results for logistic regression are shown in table 7.7. Genes were again removed from the gene pathway sets to see the effect it had on the accuracy. Again some genes have more effect than others and removing them affects the accuracy negatively. This is shown in figures 7.14 and 7.15.

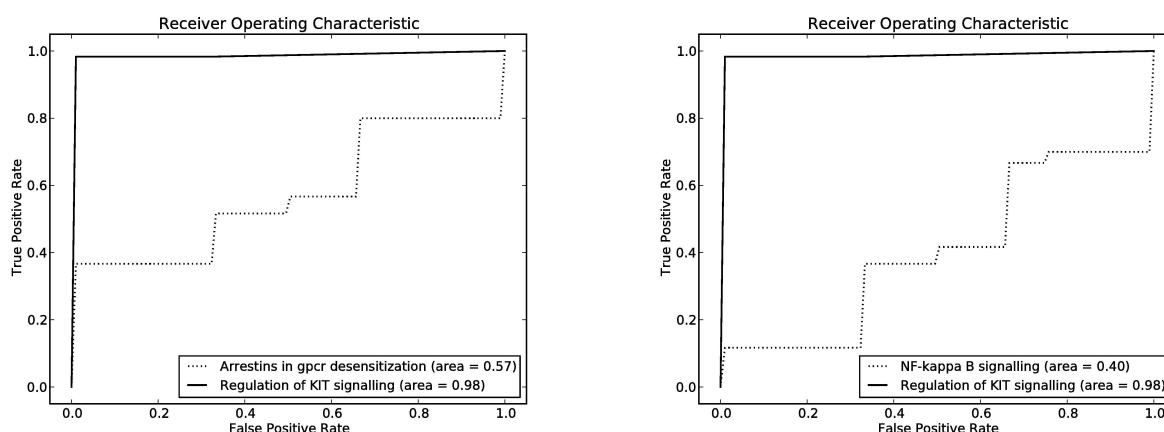


Figure 7.12: **Comparison between Regulation of KIT signalling and Arrestins in gpcr desensitization (left) and NF-kappa B signalling pathway - Homo sapiens (right) pathway sets**

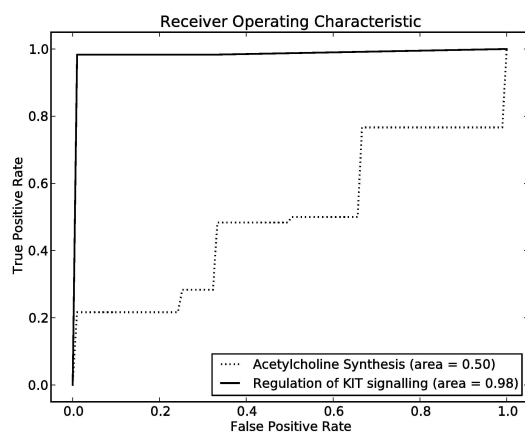


Figure 7.13: Comparison between Regulation of KIT signalling and Acetylcholine Synthesis

Pathway Name	Logistic Regression
Regulation of KIT signalling	0.703
Signaling events mediated by Stem cell factor receptor (c-Kit)	0.693
Superpathway of D-myo-inositol(1,4,5)-trisphosphate metabolism	0.682

Table 7.7: **Logistic Regression applied on CML:** Results for logistic regression on CML gene pathway sets

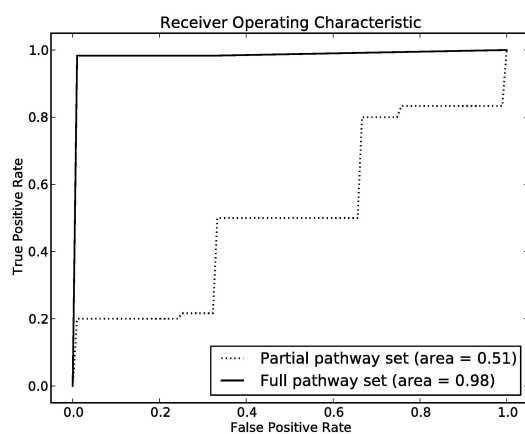


Figure 7.14: 60% of the genes were removed from the Regulation of KIT signalling pathway

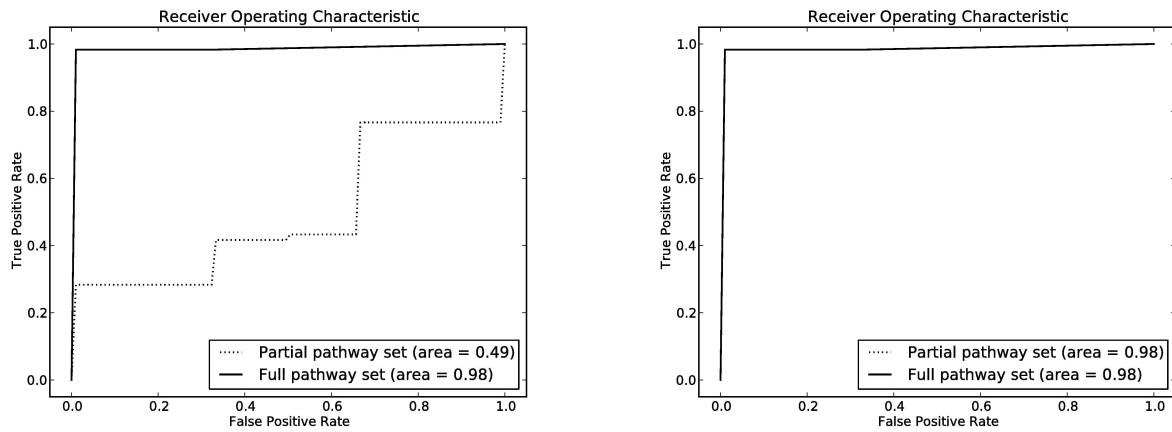


Figure 7.15: 80% of the genes were removed from the Regulation of KIT signalling pathway

Comparing the accuracy of all the random sets (shown in figures 7.16, 7.17), after 10 different runs it is observed that the accuracies vary between 0.4 and 0.7. Their variance is either equal to or much greater than the variance of the pathway sets. The variance of the two gene pathway sets is 0.0011 while for the random sets variance is between 0.0165 and 0.0260. The random gene pathway sets were used to calculate the z-score of the accuracy and the p-values of the random sets compared to the two highest scoring gene pathway sets.

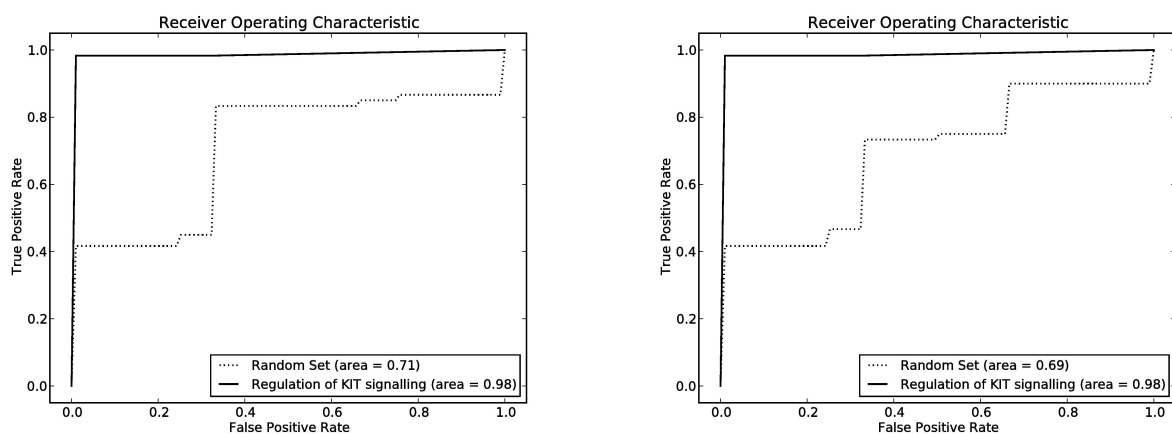


Figure 7.16: Comparison between Regulation of KIT signalling and a random pathway set of 644 genes (left) and a random pathway set of 1170 genes (right)

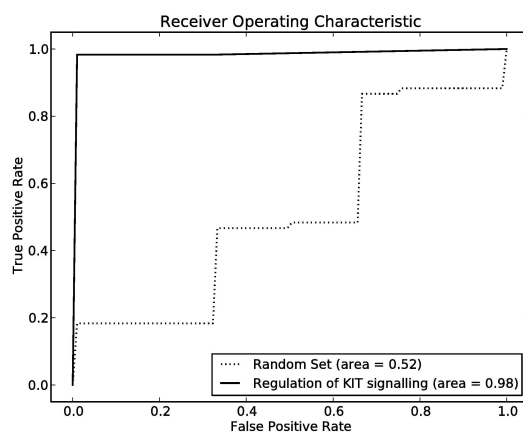


Figure 7.17: **Comparison between Regulation of KIT signalling and a random pathway set of 728 genes**

Gene SH2B3

Even though the purpose of this study was not to identify single genes, a single gene seemed to stand out. By studying how the AdaBoost was constructing the classifier models for the data, it was apparent that the CpG island cg00056489, which translates to gene SH2B3 or SH2B adaptor protein 3, was the gene that most of the modelling was based on. In fact removing this gene only from the pathway set reduced the classification accuracy to random (≈ 0.5) from the initial 0.99 before removal.

Gene Set

Another set of genes that is important to classification was constructed. The accuracy of this set was 0.94. The list of genes is shown in table 7.8 with the genes previously associated with CML shown in bold. SH2B3 was deliberately excluded from the list so it will not bias the result.

Gene Name	Functional Annotation
INPP5A	polyphosphate-5-phosphatase, 40kDa
INPP5B	inositol polyphosphate-5-phosphatase, 75kDa
IMPAD1	inositol monophosphatase domain containing protein 1
INPP1	inositol polyphosphate-1-phosphatase
INPP5J	inositol polyphosphate-5-phosphatase J
ITPKB	inositol 1,4,5-trisphosphate 3-kinase B
SYNJ2	synaptojanin 2

Table 7.8: **Gene List for CML:** Combined Gene List of the most influential genes for CML

7.5 Discussion

In this chapter a novel way of analysing big datasets by segmenting them based on prior pathway information is presented. Analysing the genes belonging to a pathway separately can give some information as to how a disease is related to that pathway and which biological mechanisms are involved. It also reduces the number of the variables to be analysed significantly, since now we are dealing with small pathway sets instead of the whole methylation dataset, and makes the computation tractable as shown in table 7.9. There were four pathway sets for LGG and two for CML and two lists of genes that show statistical association with response to treatment. Further experimentation, analysis and clinical significance testing must be performed to determine whether these results can be used to define an effective biomarker in a clinical setting in the battle against LGG and CML.

Pathway Size	Computational Time
≈ 100	≈ 3 seconds
≈ 500	≈ 7 seconds
≈ 1000	≈ 14 seconds
≈ 1000	≈ 17 seconds
≈ 2000	≈ 21 seconds

Table 7.9: **Computational Complexity Using Pathway Segmentation**

Four pathway sets were found that can classify progression and response to treatment accurately for LGG. All of them have previously been associated with gliomas and brain cancer. Pantothenate and CoA biosynthesis is related to brain neurodegeneration and iron accumulation in the brain [MC11]. Studies have shown that the ratio between iron and zinc affects the malignancy of the tumour [WCW⁺13]. IL2 (Interleukin 2) was first used in the treatment of glioma in 1986 as part of immunotherapy treatments [OKZ⁺09]. Moreover Pyrimidine metabolism in human gliomas is increased by comparison to normal brain [VBL94]. The last pathway is related to the transcription factor creb. Transcription factor creb is shown to be over-expressed in gliomas [TWZ⁺12] and it is related to their proliferation [DFB⁺14]. In addition a pathway set that had an accuracy of 0.85 is worth noting. This is the Renal cell carcinoma pathway. It was shown that Renal cell carcinoma (RCC) is one of the most common sources of brain metastases [RLM12]. From the optimal set of genes that was found response can be predicted very accurately (0.99). Some of the genes in the set have already been associated with gliomas. In particular, signal transducer and activator of transcription 3 has been associated with inducing the progression and formation of gliomas [KPR⁺14] ; Gab2 is related to glioma tumour invasion [SSZ⁺13]; NTRK1 has been associated with the different grades of glioma [PAV14] and KRas can initiate the growth of gliomas [HW05].

For CML, Gene SH2B3 was shown to be statistically related to the response to CML leukaemia treatment. SH2B adapter protein 3 is a protein that in humans is encoded by the SH2B3 gene [MMRK96, HTMZ⁺97]. Its role is to be involved in a range of signalling activities by growth factor and cytokine receptors. It is a member of the family of tyrosine kinase adapter proteins [ASK⁺99], the high-affinity cell surface receptors for many polypeptide growth factors, cytokines, and hormones [RWL00], which are shown to be involved with the progression of many types of cancer. The possibility of manipulating receptor tyrosine kinase signalling in order to prevent cancer or enhance cancer therapy was explored previously [ZBU01]. It is a key protein for the negative regulator of cytokine signalling and plays a critical role in hematopoiesis. This kind of cell is very much related with leukaemia [Sac96, Nat13]. More over SH2B3 has already been identified as a predisposition gene to Acute Lymphoblastic Leukaemia (ALL) [Wil13].

From the set of genes that can also predict response very accurately (0.94), inositol polyphosphate-5-phosphatase has already been associated with leukaemia in [MJN⁺94]. In addition it is associated with *SH2* since it encodes a protein in that domain. The protein is related to hematopoietic cells and its movement from the cytosol to the plasma membrane is mediated by tyrosine phosphorylation [LSJ⁺98]. Synaptojanin was also found in the set which belongs to the inositol-polyphosphate 5-phosphatase family that has previously been associated with hairy cell leukaemia, a chronic mature B-cell leukaemia characterized by malignant B cells that have typical hairy protrusions [SDVDVDM⁺03].

7.6 *A priori* Manifold Learning Over Pathway Segmentation

A priori manifold learning was applied on the pathway sets extracted from both the CML and the LGG datasets. The algorithm over-fits, as expected, since the datasets are already split into pathway sets and *a priori* manifold learning increases the importance of the features that exist in the same pathway. Given that they are already separated into pathway sets then all the features are classified as equally important. The accuracies were between 0.85 and 1 for all pathway sets. The results are obviously not reliable since the classifier needs to be trained on more general data.

Chapter 8

Conclusion

The aim of this thesis is to introduce some new methods for dimensionality reduction of cancer datasets using prior knowledge. The contribution is the development and testing of two novel dimensionality reduction algorithms, one of which is a feature extraction method and the other one is a feature selection method. Data from various biological databases that contain information about gene pathways were combined with already existing dimensionality reduction algorithms. In addition a number of machine learning methods were used for linear and non-linear dimensionality reduction and classification and were compared with the proposed methods. Both algorithms can work with other forms of microarray data and are not limited to cancer. In this chapter, the accomplishments presented in this thesis are summarised with some suggestions for future work. Part of the future work has already been undertaken as a master's project with my guidance.

8.1 Thesis Summary

Chapter 1: describes the motivation for this thesis and the contributions. It briefly introduces the *a priori* manifold learning and the feature selection over pathway segmentation algorithms.

Chapter 2: introduces the microarray technology, explains how microarrays are used and

what are the different types. It includes an explanation of how the analysis of microarray data works.

Chapter 3: gives a mathematical background of the dimensionality reduction and classification methods used throughout this thesis. Methods include: Principal Components Analysis, manifold - Isomap, Support Vector Machines, Linear Discriminant Analysis, k - Nearest Neighbours, AdaBoost with decision trees as weak classifiers, internal cluster validation and Cross Validation.

Chapter 4: provides a background on the current progress in machine learning applied on microarray data and introduces the main problems, which are the curse of dimensionality and the noise that can be found in this kind of data. It also explains the two ways of performing dimensionality reduction, feature selection and feature extraction, how they are different and how they have been used up to now.

Chapter 5: explains the *a priori* manifold learning algorithm and the way prior knowledge was used in constructing the manifold instead of filtering out data. *A priori* manifold learning was compared in terms of cluster density and accuracy using three different classification methods with principal components analysis and the conventional Isomap. *A priori* manifold learning performs significantly better than the other methods.

Chapter 6: implements a modification of the *a priori* manifold learning algorithm described in chapter 5 which incorporates information from the GO terms database. The results are not as good as with pathways since most of the models created over-fit and that makes the accuracy and the cluster quality unreliable. For the few models that do not over-fit the accuracy is higher than the accuracy obtained by *a priori* manifold learning using pathway information.

Chapter 7: describes the feature selection over pathway segmentation algorithm. The effectiveness was proven using a CML leukaemia and a Lower Grade Glioma dataset for which pathways were identified, and in the case of CML leukaemia a single gene that seems to be statistically related to the response to treatment. In addition lists of genes are shown, that can be equally effective in predicting response. The classification method used was AdaBoost using decision trees as weak classifiers. The results from this chapter can be used

to undertake further clinical research in order to investigate further the involvement of the genes in question to CML and LGG.

Chapter 8: concludes the thesis with a summary of the contributions and future work.

To conclude, in chapters 5, 6 and 7 two novel methods of feature extraction and feature selection for biological datasets that incorporate prior knowledge from on-line pathway databases are described. Biological datasets can be exceptionally noisy and are high-dimensional and therefore prior knowledge can help improving the accuracy and overcome noise. A key point was that prior knowledge was combined with the Isomap algorithm in such a way so that it can only be used when constructing the manifold and not for filtering data out as is the case with most methods that make use of prior knowledge. The method was much more effective without increasing the complexity of the algorithm. In addition the CML dataset that was deemed impossible to analyse up to now due to its very high dimensionality was successfully analysed. It was broken down in pathways and each pathway was analysed separately. A gene was discovered, that has been associated with other forms of leukaemia before, that is related to the response to CML treatment. The gene was included in the two higher scoring pathways found to be related to CML. Four pathways were found to be related to LGG and lists of genes that can classify response to treatment with very high accuracy for CML and LLG were composed.

8.2 Future Work

8.2.1 *A Priori* Manifold Learning Using Locally-Linear Embedding

Locally-linear embedding (LLE) [RS00] appeared around the same time as Isomap and has several advantages over Isomap which include faster optimization and sparse matrix algorithms. A comparison between the two is shown in table 8.1.

Isomap	Locally Linear Embedding
Applies Multidimensional Scaling (MDS) on the geodesic distances	Models local neighbourhoods as linear relations and then embeds them on a lower dimension
Global approach that preserves the mapping and geometry of all points which gives a more faithful representation	Local approach that might not be as accurate but it is faster (polynomial speed-up) due to sparse matrix computations. Could find results whose local geometry is close to Euclidean but not their global geometry
Might fail for non-convex manifolds	Can work with any manifold

Table 8.1: Comparison between Isomap and Locally Linear Embedding

8.2.2 Learning Causality Using Bayesian Networks

An extension to the feature selection over pathway segmentation algorithm was proposed in a master's thesis [Suk15] by Nimalesh Sukumar. In his thesis Sukumar investigated ways in which the causality between the genes in the dataset could be inferred.

Constrained Based Bayesian Learner

The gene pathway sets were analysed using a constrained based Bayesian learner. The first step of the algorithm was to infer the Bayesian Network with undirected edges and then

use different hypotheses to construct the directed edges. The null hypothesis in this case is that the genes are independent. The X^2 statistic shown in equation 8.1 was used with a 95% confidence interval to test the hypothesis. Continuous and discrete learners were used. Continuous Learners assume linear relationships between genes, while for discrete learners the data had to be quantised using different number of quantiles. The continuous learner performed the worst (no edges were identified), while the discrete learner with 2 quantiles performed better. The performance was measured using the number of Bayesian networks produced with the highest number of edges (41 of the pathways had 6 or more edges). However only quantising the dataset in 2 quantiles leads to loss of information and therefore the results were not accurate.

$$X^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \quad (8.1)$$

Kruskal's Algorithm With Mutual Information

A different method using Kruskal's algorithm applied to mutual information was used. For this experiment only a subset of the pathways was used which only included the 4 highest scoring pathways from the Lower Grade Glioma dataset. The algorithm is shown in algorithm 8.1. The equation used as the dependency metric is the mutual entropy or the Kullback-Leibler divergence shown in equation 8.2. Boosting was also applied to test the robustness of the algorithm. The dataset was sampled with replacement 100 times to produce sets with the same number of data points as the original dataset as shown in algorithm 8.2.

Algorithm 8.1. Kruskal's algorithm

```

for each pair of genes in dataset do
    calculate mutual entropy
    order variable pairs in terms of mutual entropy
    A is empty set
for each gene G in dataset do
    Make-Set (G)
    for each (a,b) in ordered mutual entropy list do
        if GET-SET(a) does not equal GET-SET(b) then
            A = A union (a,b)
        end if
    end for
    end for
end for

```

Algorithm 8.2. Boosting algorithm

```

for  $j = 1$  to 100 do
    GenSet is empty set
    while  $i \leq N$  do
        let  $k$  be random int in 1 to  $N$ 
        add sample  $k$  to GenSet
        increment  $i$ 
    end while
    Apply Bayesian Network Algorithm to GenSet
end for

```

$$CondDep(A, B) = \sum_{A \times B} P(a_i \& b_j) \log_2 \frac{P(a_i \& b_j)}{P(a_i)P(b_j)} \quad (8.2)$$

The results showed that when the labels of the samples were not taken into account, many edges kept appearing a large number of times during the boosting process. The mutual information with the 95% confidence interval show that 8 edges satisfy the constraint. Specifically gene ANK1 had a significant dependency with SLC04A1, PIK3CD, GPX4, STAT3 and IRS1. When the labels were taken into account for the responsive to treatment pa-

tients, there were no significant results while for the unresponsive patients the edge ANK1–PIK3CD appeared to be significant

Marginal Independence

An experiment to identify colliders (shown in figure 8.1) was also conducted using marginal independence. The marginal independence algorithm (shown in algorithm 8.3) was run on the Bayesian networks identified using the Kruskal’s algorithm. A score function using both the conditional independence and the mutual entropy was used to score the colliders as shown in equation 8.3.

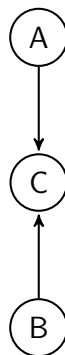


Figure 8.1: A Collider

Algorithm 8.3. Marginal Independence

```

for VarA in GeneSet do
  for VarCent in GeneSet do
    for VarB in GeneSet do
      if VarA-VarCent AND VarCent-VarB in Bayesian Network then
        calculate CondDep on (VarA,VarB,VarCent)
        calculate Mutual Information on (VarA,VarB)
        add (VarA,VarB,VarCent) to Collider List
      end if
    end for
  end for
end for

```

$$CondDep = \frac{CondInt(A, B, C) - MutEntr(A, B)}{CondInt(A, B, C)} \quad (8.3)$$

When the algorithm was run on the Bayesian networks without the labels taken into account one collider was identified (NF1, STX1A, NRCAM) with a high score. When the labels for the unresponsive and responsive Patients were taken into account the algorithm identified 12 colliders for unresponsive patients and 11 for the responsive ones. GAB2 was consistently the child variable in 5 out of the 11 colliders identified in the responsive patients.

Cyclic Bayesian Networks

In order to identify potential cycles in the dataset and have directed edges in the networks the Cyclic Bayesian algorithm was used (shown in algorithm 8.4). The dataset for this experiment only included the 4 highest scoring pathways. It identified 4,729 colliders which made the identification of any important relations impossible. A threshold on the number of edges allowed in the network was added ($2 \times$ number of vertices, which would be approximately double the edges in a connected tree), and it produced 12 edges that were the same as the ones identified by the Kruskal algorithm. The marginal independence algorithm was run in order to identify colliders but nothing significant was produced. Inferring

the causality on the dataset unfortunately did not yield significant results.

Algorithm 8.4. Cyclic Bayesian Networks

```

for each pair of genes in dataset do
    calculate mutual entropy
end for

order variable pairs in terms of mutual entropy
VisitedVertices is emptyset
BayesNet is emptyset

for each (v1, v2) in ordered mutual entropy list do
    if v1 not in VisitedVertices then
        Add v1 to VisitedVertices
        if v2 not in VisitedVertices then
            Add v2 to VisitedVertices
            add (v1, v2) to BayesNet
            if  $\text{len}(\text{VisitedNodes}) \geq \text{threshold}$  then
                Break
            end if
        end if
    end if
end for

```

Overall Bayesian methods for identifying to identify conditional relationships and interactions among genes did not yield any meaningful result. Improvements to the above methods could be made in order to incorporate prior knowledge from directed gene pathway databases. A combination of the prior knowledge and the information found in the microarray dataset might be shown to be a better of way of identifying relationships among genes.

Bibliography

- [ABB⁺00] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000.
- [AJL⁺02] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. Molecular biology of the cell, 2002.
- [Alb08] B. Alberts. *Molecular Biology of the Cell: Reference edition*. Number v. 1 in Molecular Biology of the Cell: Reference Edition. Garland Science, 2008.
- [AM02] Christophe Ambroise and Geoffrey J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences*, 99(10):6562–6566, May 2002.
- [ASH13] Dvir Aran, Sivan Sabato, and Asaf Hellman. DNA methylation of distal regulatory sites characterizes dysregulation of cancer genes. *Genome Biology*, 14(3):R21+, March 2013.
- [ASK⁺99] Z. Ahmed, B. J. Smith, K. Kotani, P. Wilden, and T. S. Pillay. Aps, an adapter protein with a PH and SH2 domain, is a substrate for the insulin receptor kinase. *Biochem J*, 341 (Pt 3):665–8, 1999.

- [Bay05] Stephen B. Baylin. DNA methylation and gene silencing in cancer. *Nature clinical practice. Oncology*, 2 Suppl 1:S4–11, December 2005.
- [BB12] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, February 2012.
- [Bel57] Richard Ernest Bellman. *Dynamic programming*. Number ISBN 978-0-691-07951-6. Princeton University Press, 1957.
- [Bes14] Emmanuel C. Besa. Chronic myelogenous leukemia. <http://emedicine.medscape.com/article/199425-overview>, March 2014.
- [Bez81] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.
- [BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.
- [BG05] I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.
- [BG14] Daniel J. Brat and The TCGA Lower Grade Gliomas Analysis Working Group. Integrative genomic characterization of lower grade gliomas. *Neuro-Oncology*, 16(suppl 3):iii3, 2014.
- [BGZJ11] Elnaz Barshan, Ali Ghodsi, Zohreh Azimifar, and Mansoor Zolghadri Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recogn.*, 44(7):1357–1371, July 2011.
- [BGL⁺00] Michael P. S. Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terrence S. Furey, Manuel Ares, and David Hausler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences*, 97(1):262–267, 2000.

- [BKG⁺09] Nirmalya Bandyopadhyay, Tamer Kahveci, Steve Goodison, Yijun Sun, and Sanjay Ranka. Pathway-based feature selection algorithm for cancer microarray data. *Adv. Bioinformatics*, 2009, 2009.
- [BKR⁺10] Christoph Bartenhagen, Hans-Ulrich Klein, Christian Ruckert, Xiaoyi Jiang, and Martin Dugas. Comparative study of unsupervised dimension reduction techniques for the visualization of microarray gene expression data. *BMC Bioinformatics*, 11(1):567, 2010.
- [BL97] Avrim L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.
- [BN01] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, 2001.
- [Boc12] Christoph Bock. Analysing and interpreting DNA methylation data. *Nat Rev Genet*, 13(10):705–19, 2012.
- [Box87] Joan Fisher Box. Guinness, Gosset, Fisher, and small samples. *Statistical Science*, 2(1):45–52, 02 1987.
- [BPT06] Eric Bair, Debashis Paul, and Robert Tibshirani. Prediction by supervised principal components. *Journal of the American Statistical Association*, 101:119–137, 2006.
- [BR92] Avrim L. Blum and Ronald L. Rivest. Training a 3-node neural network is NP-Complete, 1992.
- [BRS⁺01] A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceed-*

ings of the National Academy of Sciences of the United States of America, 98(24):13790–13795, November 2001.

- [BS02] M. Balasubramanian and E. L. Schwartz. The Isomap Algorithm and Topological Stability. *Science*, 295(5552):7, January 2002.
- [BT04] Eric Bair and Robert Tibshirani. Semi-supervised methods to predict patient survival from gene expression data, 2004.
- [BTGM04] Jean-Philippe Brunet, Pablo Tamayo, Todd R. Golub, and Jill P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169, 2004.
- [Cai08] Caitlin Smith . Methylation detection. <http://www.biocompare.com/Editorial-Articles/41604-Methylation-Detection/>, 2008.
- [Cay05] L. Cayton. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 2005.
- [CBD⁺13] Angelo M. Carella, Susan Branford, Michael Deininger, Francois X. Mahon, Giuseppe Saglio, Anna Eiring, Jamshid Khorashad, Thomas O’Hare, and John M. Goldman. What challenges remain in chronic myeloid leukemia research? *Haematologica*, 98(8):1168–1172, 2013.
- [CCC⁺03] L. J. Cao, K. S. Chua, W. K. Chong, H. P. Lee, and Q. M. Gu. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputing*, 55(1-2):321–336, September 2003.
- [CCM⁺04] J. Cheng, M. Cline, J. Martin, D. Finkelstein, T. Awad, D. Kulp, and M.A. Siani-Rose. A knowledge-based clustering algorithm driven by gene ontology. *J Biopharm Stat*, 14(3):687–700, 2004.
- [CHA⁺12] J. Michael Cherry, Eurie L. Hong, Craig Amundsen, Rama Balakrishnan, Gail Binkley, Esther T. Chan, Karen R. Christie, Maria C. Costanzo,

Selina S. Dwight, Stacia R. Engel, Dianna G. Fisk, Jodi E. Hirschman, Benjamin C. Hitz, Kalpana Karra, Cynthia J. Krieger, Stuart R. Miyasato, Robert S. Nash, Julie Park, Marek S. Skrzypek, Matt Simison, Shuai Weng, and Edith D. Wong. Saccharomyces genome database: the genomics resource of budding yeast. *Nucleic Acids Research*, 40(Database-Issue):700–705, 2012.

[CL04] Shi Chao and Chen Lihui. Feature dimension reduction for microarray data analysis using locally linear embedding. In *APBC*, pages 211–217, 2004.

[Cle79] William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.

[CLL⁺07] Han-Yu Chuang, Eunjung Lee, Yu-Tsueng Liu, Doheon Lee, and Trey Ideker. Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, 3(1), October 2007.

[COJT⁺97] Martin Carroll, Sayuri Ohno-Jones, Shu Tamura, Elisabeth Buchdunger, Jürg Zimmermann, Nicholas B. Lydon, D. Gary Gilliland, and Brian J. Druker. Cgp 57148, a tyrosine kinase inhibitor, inhibits the growth of cells expressing BCR-ABL, TEL-ABL, and TEL-PDGFR fusion proteins. *Blood*, 90(12):4947–4952, 1997.

[Con15] The Gene Ontology Consortium. Gene ontology consortium: going forward. *Nucleic Acids Research*, 43(D1):D1049–D1056, 2015.

[Cri58] F. H. C. Crick. On Protein Synthesis. *The Symposia of the Society for Experimental Biology*, 12:138–163, 1958.

[CRI70] FRANCIS CRICK. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 08 1970.

- [CSG09] Cesar Cobaleda and Isidro Sanchez-Garcia. B-cell acute lymphoblastic leukaemia: towards understanding its cellular origin. *BioEssays*, 31(6):600–609, 2009.
- [CSTK01] Nello Cristianini, John Shawe-Taylor, and Jaz S. Kandola. Spectral kernel methods for clustering. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *NIPS*, pages 649–655. MIT Press, 2001.
- [CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.
- [CVBM02] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Mach. Learn.*, 46(1-3):131–159, March 2002.
- [CW09a] Xi Chen and Lily Wang. Integrating biological knowledge with gene expression profiles for survival prediction of cancer. *Journal of Computational Biology*, 16(2):265–278, February 2009.
- [CW09b] Xi Chen and Lily Wang. Integrating biological knowledge with gene expression profiles for survival prediction of cancer. *Journal of Computational Biology*, 16(2):265–278, 2009.
- [CX04] Y. Chen and D. Xu. Global protein function annotation through mining genome-scale data in yeast *Saccharomyces cerevisiae*. *Nucleic Acids Res*, 32(21):6414–6424, 2004.
- [Das01] Sanmay Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 74–81, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [DB79] David L. Davies and Donald W. Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1(2):224–227, 1979.

- [DFB⁺14] P. Daniel, G. Filiz, D. V. Brown, F. Hollande, M. Gonzales, G. D. Abaco, N. Papalexis, W. A. Phillips, J. Malaterre, R. G. Ramsay, and T. Mantamadiotis. Selective creb-dependent cyclin expression mediated by the PI3K and MAPK pathways supports glioma cell proliferation. *Oncogenesis*, 3:e108, 2014.
- [DG03] David L. Donoho and Carrie Grimes. Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data, 2003.
- [DHHR02] B. P. Durbin, J. S. Hardin, D. M. Hawkins, and D. M. Rocke. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics*, 18(suppl 1):S105–S110, 2002.
- [Die00] Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15, London, UK, UK, 2000. Springer-Verlag.
- [Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.
- [dMFF⁺12] Catherine de Martel, Jacques Ferlay, Silvia Franceschi, Jérôme Vignat, Freddie Bray, David Forman, and Martyn Plummer. Global burden of cancers attributable to infections in 2008: a review and synthetic analysis. *The Lancet Oncology*, 13(6):607–615, June 2012.
- [DP03] Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of the IEEE Computer Society Conference on Bioinformatics*, CSB '03, pages 523–, Washington, DC, USA, 2003. IEEE Computer Society.
- [DR04] Blythe P. Durbin and David M. Rocke. Variance-stabilizing transformations for two-color microarrays. *Bioinformatics*, 20(5):660–667, 2004.
- [DRM05] Kevin Dawson, Raymond L. Rodriguez, and Wasyl Malyj. Sample phenotype clusters in high-density oligonucleotide microarray data sets are

revealed using isomap, a nonlinear algorithm. *BMC Bioinformatics*, 6:195, 2005.

- [DUdA06] Ramon Diaz-Uriarte and Sara Alvarez de Andres. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3, 2006.
- [Dun73] J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.
- [EBES05] Paul F. Evangelista, Piero Bonissone, Markj. Embrechts, and Boleslaw K. Szymanski. Unsupervised fuzzy ensembles and their use in intrusion detection. In *Proceedings of the European Symposium on Artificial Neural Networks*, 2005.
- [EDMA10] Jesse M. Engreitz, Bernie J. Daigle, Jonathan J. Marshall, and Russ B. Altman. Independent component analysis: mining microarray data for fundamental human gene expression modules. *Journal of biomedical informatics*, 43(6):932–944, December 2010.
- [ERZ⁺11] Martin Ehler, Vinodh Rajapakse, Barry Zeeberg, Brian Brooks, Jacob Brown, Wojciech Czaja, and Robert Bonner. Nonlinear gene cluster analysis with labeling for microarray gene expression data in organ development. *BMC Proceedings*, 5(Suppl 2):S3, 2011.
- [ESBB98] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, December 1998.
- [ET93] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.
- [Faw06] Tom Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, 2006.

- [Fie07] Andy P. Field. *Analysis of Variance (ANOVA)*, pages 33–36. SAGE Publications, Inc., 1 edition, 2007.
- [Fis36] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.
- [Flo62] Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345–, June 1962.
- [Fre05] David Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, August 2005.
- [FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.
- [GC65] H. Glass and L. Cooper. Sequential search: A method for solving constrained optimization problems. *J. ACM*, 12(1):71–82, January 1965.
- [Gei93] Seymour Geisser. *Predictive inference: an introduction*. Chapman & Hall, New York, 1993.
- [GFHK09] Martin Gütlein, Eibe Frank, Mark Hall, and Andreas Karwath. Large scale attribute selection using wrappers. In *Proc. of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2009)*. 2009.
- [GHHZ09] Peng Guan, Desheng Huang, Miao He, and Baosen Zhou. Lung cancer gene expression database analysis incorporating prior knowledge with support vector machine-based classification method. *Journal of Experimental & Clinical Cancer Research*, 28(1):103+, 2009.
- [GK03] Peter J. Giles and David Kipling. Normality of oligonucleotide microarray data and implications for parametric statistical analyses. *Bioinformatics*, 19(17):2254–2262, 2003.

- [GS03] Ali Ghodsi and Dale Schuurmans. Automatic basis selection techniques for {RBF} networks. *Neural Networks*, 16(5-6):809 – 816, 2003. Advances in Neural Networks Research: {IJCNN} '03.
- [GST⁺99] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and C. D. Bloomfield. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.
- [GTS⁺01] M. E. Garber, O. G. Troyanskaya, K. Schluens, S. Petersen, Z. Thaesler, M. Pacyna-Gengelbach, M. van de Rijn, G. D. Rosen, C. M. Perou, R. I. Whyte, R. B. Altman, P. O. Brown, D. Botstein, and I. Petersen. Diversity of gene expression in adenocarcinoma of the lung. *Proceedings of the National Academy of Sciences of the United States of America*, 98(24):13784–13789, November 2001.
- [GWBV02] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.
- [Hal98] Mark A. Hall. Correlation-based feature selection for machine learning. Technical report, 1998.
- [Hal00] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 359–366, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [Han89] T.R. Hancock. *On the Difficulty of Finding Small Consistent Decision Trees*. 1989.
- [Han05] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

- [HCO06] Sepp Hochreiter, Djork-Arne Clevert, and Klaus Obermayer. A new summarization method for Affymetrix probe level data. *Bioinformatics*, 22(8):943–949, 2006.
- [HDH08] Edmundo Bonilla Huerta, Beatrice Duval, and Jin-Kao Hao. Gene selection for microarray data by a lda-based genetic algorithm. In Madhu Chetty, Alioune Ngom, and Shandar Ahmad, editors, *PRIB*, volume 5265 of *Lecture Notes in Computer Science*, pages 250–261. Springer, 2008.
- [Hei05] Anne-Mette K. Hein. BGX: a fully Bayesian integrated approach to the analysis of Affymetrix GeneChip data. *Biostatistics*, 6(3):349–373, July 2005.
- [HN09] Paul J. Hurd and Christopher J. Nelson. Advantages of next-generation sequencing versus the microarray in epigenetic research. *Briefings in Functional Genomics and Proteomics*, 8(3):174–183, 2009.
- [HP06] Desheng Huang and Wei Pan. Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Bioinformatics*, 22(10):1259–1268, 2006.
- [HRZZ09] Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. Multi-class AdaBoost. *Statistics and Its Interface*, 2(3):349–360, 2009.
- [HS98] Mark A. Hall and Lloyd A. Smith. Practical feature subset selection for machine learning, 1998.
- [HSL09a] Da Wei Huang, Brad T Sherman, and Richard A Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1):1–13, 2009.
- [HSL09b] Da Wei a. . W. Huang, Brad T. Sherman, and Richard A. Lempicki. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature protocols*, 4(1):44–57, December 2009.

- [HTE⁺00] Trevor Hastie, Robert Tibshirani, Michael Eisen, Ash Alizadeh, Ronald Levy, Louis Staudt, Wing Chan, David Botstein, and Patrick Brown. 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1(2):research0003.1–research0003.21, 2000.
- [HTG14] Zena M. Hira, George Trigeorgis, and Duncan F. Gillies. An algorithm for finding biologically significant features in microarray data based on a priori manifold learning. *PLoS ONE*, 9(3):e90562, 03 2014.
- [HTMZ⁺97] L. R. Hendricks-Taylor, D. G. Motto, J. Zhang, R. P. Siraganian, and G. A. Koretzky. SLP-76 is a substrate of the high affinity IgE receptor-stimulated protein tyrosine kinases in rat basophilic leukemia cells. *J Biol Chem*, 272(2):1363–7, 1997.
- [HvHS⁺02] Wolfgang Huber, Anja von Heydebreck, Holger SÄijltmann, Annemarie Poustka, and Martin Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18(suppl 1):S96–S104, 2002.
- [HW79] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *JSTOR: Applied Statistics*, 28(1):100–108, 1979.
- [HW05] Sheri L Holmen and Bart O Williams. Essential role for RAS signaling in glioblastoma maintenance. *Cancer research*, 65(18):8250–8255, 2005.
- [IBC⁺03] Rafael A. Irizarry, Benjamin M. Bolstad, Francois Collin, Leslie M. Cope, Bridget Hobbs, and Terence P. Speed. Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Research*, 31(4):e15, 2003.
- [JA06] Peyman Jafari and Francisco Azuaje. An assessment of recently published gene expression data analyses: reporting experimental design and statistical factors. *BMC Medical Informatics and Decision Making*, 6(1), 2006.

- [Jac01] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.
- [JC97] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, cmp-lg/9709008, 1997.
- [JDC⁺04] Hongying Jiang, Youping Deng, Huann-Sheng Chen, Lin Tao, Qiuying Sha, Jun Chen, Chung-Jui Tsai, and Shuanglin Zhang. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC Bioinformatics*, 5:81, 2004.
- [JHE04] Hruschka Estevam R. Jr., Eduardo R. Hruschka, and Nelson F. F. Ebecken. Feature selection by bayesian networks. In Ahmed Y. Tawfik and Scott D. Goodwin, editors, *Advances in Artificial Intelligence*, volume 3060 of *Lecture Notes in Computer Science*, pages 370–379. Springer Berlin Heidelberg, 2004.
- [JL99] P. A. Jones and P. W. Laird. Cancer epigenetics comes of age. *Nature genetics*, 21(2):163–167, Feb 1999.
- [JS08] Sudhakar Jonnalagadda and Rajagopalan Srinivasan. Principal components analysis based methodology to identify differentially expressed genes in time-course microarray data. *BMC Bioinformatics*, 9, 2008.
- [JUA05] Thanyaluk Jirapech-Umpai and Stuart Aitken. Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes. *BMC bioinformatics*, 6:148, January 2005.
- [Kan97] Minoru Kanehisa. A database for post-genome analysis. *Trends in Genetics*, 13(9):375 – 376, 1997.
- [Kea98] Michael Kearns. Thoughts on hypothesis boosting, 1998.

- [KH10] Audrey Kauffmann and Wolfgang Huber. Microarray data quality control improves the detection of differentially expressed genes. *Genomics*, 95(3):138–142, March 2010.
- [KM09] S.Y. Kung and M.W. Mak. *Machine Learning in Bioinformatics*, volume Chapter 1: Feature Selection for Genomic and Proteomic Data Mining. New Jersey : John Wiley & Sons, 2009.
- [KNT⁺01] Samuel Kaski, Janne Nikkila, Petri Toronen, Eero Castren, and Garry Wong. Analysis and visualization of gene expression data using self-organizing maps, 2001.
- [Koh88] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. In James A. Anderson and Edward Rosenfeld, editors, *Neurocomputing: Foundations of Research*, pages 509–521, Cambridge, MA, USA, 1988. MIT Press.
- [KPG⁺11] Atanas Kamburov, Konstantin Pentchev, Hanna Galicka, Christoph Wierling, Hans Lehrach, and Ralf Herwig. ConsensusPathDB: toward a more complete picture of cell biology. *Nucleic Acids Research*, 39(suppl 1):D712–D717, 2011.
- [KPR⁺14] Jennifer E. Kim, Mira Patel, Jacob Ruzevick, Christopher M. Jackson, and Michael Lim. Stat3 activation in glioblastoma: Biochemical and therapeutic implications. *Cancers*, 6(1):376, 2014.
- [KSLH13] Atanas Kamburov, Ulrich Stelzl, Hans Lehrach, and Ralf Herwig. The consensusPathDB interaction database: 2013 update. *Nucleic Acids Research*, 41(D1):D793–D800, 2013.
- [KSO02] M. Kotani, A. Sugiyama, and S. Ozawa. Analysis of DNA microarray data using self-organizing map and kernel based clustering. In *Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference on*, volume 2, pages 755–759 vol.2, Nov 2002.

- [KWLH09] Atanas Kamburov, Christoph Wierling, Hans Lehrach, and Ralf Herwig. ConsensusPathDB - a database for integrating human functional interaction networks. *Nucleic Acids Research*, 37(suppl 1):D623–D628, 2009.
- [KYQG03] Yuval Kluger, Haiyuan Yu, Jiang Qian, and Mark Gerstein. Relationship between gene co-expression and probe localization on microarray slides. *BMC Genomics*, 4(1):49+, December 2003.
- [KZ10] Rafal Kustra and Adam Zagdanski. Data-fusion in clustering microarray data: Balancing discovery and interpretability. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 7(1):50–63, 2010.
- [Lar05] R.S. Larson. *Bioinformatics and Drug Discovery*. Methods in Molecular Biology. Humana Press, 2005.
- [Lau08] B. Laura. Epigenomics: The new tool in studying complex diseases. *Nature Education*, 1(178), 2008.
- [LB] Susan Lindsay and Adrian P. Bird. Use of restriction enzymes to detect potential gene sequences in mammalian DNA, url = <http://dx.doi.org/10.1038/327336a0>, volume = 327, year = 1987. *Nature*, (6120):336–338, May.
- [LB03] S. I. Lee and S. Batzoglou. Application of independent component analysis to microarrays. *Genome Biol*, 4(11), 2003.
- [LCB05] Zhenqiu Liu, Dechang Chen, and Halima Bensmail. Gene expression data classification with kernel principal component analysis. *Journal of Biomedicine and Biotechnology*, 2005(2):155–159, 2005.
- [LEAK11] Leo Lahti, Laura L. Elo, Tero Aittokallio, and Samuel Kaski. Probabilistic analysis of probe reliability in differential gene expression studies with short oligonucleotide arrays. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 8(1):217–225, January 2011.

- [Lee04] M.L.T. Lee. *Analysis of Microarray Gene Expression Data*. Trends in Logic Series. Springer, 2004.
- [Lin98] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine learning (ICML-98)*, Madison, Wisconsin, 1998.
- [LK07] Andrei Yakovlev Lev Klebanov. How high is the level of technical noise in microarray data? *Biology Direct*, 2:9, 2007.
- [LKM05] Xiaoxing Liu, Arun Krishnan, and Adrian Mondry. An Entropy-based gene selection method for cancer classification using microarray data. *BMC Bioinformatics*, 6(1):76+, 2005.
- [LL08] Caiyan Li and Hongzhe Li. Network-constrained Regularization and Variable Selection for Analysis of Genomic Data. *Bioinformatics*, March 2008.
- [LM12] Victor V. Levenson and Anatoliy A. Melnikov. DNA methylation as clinically useful biomarkers-light at the end of the tunnel. *Pharmaceuticals*, 5(1):94–113, 2012.
- [IS02] B.S. Ikonf and A.J. Smola. *Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. Adaptive computation and machine learning series. The MIT Press, 2002.
- [LSC⁺09] Qingzhong Liu, Andrew H. Sung, Zhongxue Chen, Jianzhong Liu, Xudong Huang, and Youping Deng. Feature selection and classification of MAQC-II breast cancer and multiple myeloma microarray gene expression data. *PLoS ONE*, 4(12):e8250, 12 2009.
- [LSC⁺11] Qingzhong Liu, Andrew Sung, Zhongxue Chen, Jianzhong Liu, Lei Chen, Mengyu Qiao, Zhaohui Wang, Xudong Huang, and Youping Deng. Gene selection and classification for cancer microarray data based on machine learning and similarity measures. *BMC Genomics*, 12(Suppl 5):S1, 2011.

- [LSJ⁺98] Q. Liu, F. Shalaby, J. Jones, D. Bouchard, and D. J. Dumont. The SH2-containing inositol polyphosphate 5-phosphatase, ship, is expressed during hematopoiesis and spermatogenesis. *Blood*, 91(8):2753–9, 1998.
- [LW01] Cheng Li and Wing Hung Wong. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Sciences*, 98(1):31–36, 2001.
- [LWW02] Jobst Landgrebe, Wolfgang Wurst, and Gerhard Welzl. Permutation-validated principal components analysis of microarray data. *Genome Biol*, 3(4), 2002.
- [LY09] Xiaofang Liu and Chun Yang. Greedy kernel PCA for training data reduction and nonlinear feature extraction in classification, 2009.
- [MAQ06] MAQC consortium. The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nat Biotechnol*, 24(9):1151–1161, 2006.
- [MBM⁺04] Géraldine Mercier, Nathalie Berthault, Jérémie. Mary, Julie Peyre, Anestis Antoniadis, Jean-Paul Comet, Antoine Cornuejol, Christine Froidevaux, and Marie Dutreix. Biological detection of low radiation doses by combining results of two microarray analysis methods. *Nucleic Acids Res.*, 32(1):12–12, January 2004.
- [MC11] Alisdair Mcneill and Patrick F. Chinnery. Chapter 9 - neurodegeneration with brain iron accumulation. In William J. Weiner and Eduardo Tolosa, editors, *Hyperkinetic Movement Disorders*, volume 100 of *Handbook of Clinical Neurology*, pages 161 – 172. Elsevier, 2011.
- [MCYC06] Patrick C. H. Ma, Keith C. C. Chan, Xin Yao, and David K. Y. Chiu. An evolutionary clustering algorithm for gene expression microarray data analysis. *IEEE Trans. Evolutionary Computation*, 10(3):296–314, 2006.

- [MDA05] G. McLachlan, K.A. Do, and C. Ambroise. *Analyzing Microarray Gene Expression Data*. Wiley Series in Probability and Statistics. Wiley, 2005.
- [MDH⁺05] S. Maier, C. Dahlstroem, C. Haefliger, A. Plum, and C. Piepenbrock. Identifying DNA methylation biomarkers of cancer drug response. *Am J Pharmacogenomics*, 5(4):223–32+, 2005.
- [MJN⁺94] K. Mengubas, S. A. Jabbar, K. E. Nye, A. V. Wilkes, S. and Hoffbrand, and R. G. Wickremasinghe. Inactivation of calcium ion-regulating inositol polyphosphate second messengers is impaired in subpopulations of human leukemia cells. *Leukemia*, 8(10):1718–1725, 1994.
- [MML03] M. Niranjana M. Milo, A. Fazeli and N. Lawrence. A probabilistic model for the extraction of expression levels from oligonucleotide arrays. *Biochemical Society Transactions*, 31:1510–1512, 2003.
- [MMRK96] D. G. Motto, M. A. Musci, S. E. Ross, and G. A. Koretzky. Tyrosine phosphorylation of Grb2-associated proteins correlates with phospholipase c gamma 1 activation in t cells. *Mol Cell Biol*, 16(6):2823–9, 1996.
- [MMS⁺07] Shuangge Ma, Shuangge Ma, Xiao Song, Xiao Song, Jian Huang, and Jian Huang. Supervised group lasso with applications to microarray data analysis., 2007.
- [MSH⁺02] J. Misra, W. Schmitt, D. Hwang, L. L. Hsiao, S. Gullans, G. Stephanopoulos, and G. Stephanopoulos. Interactive exploration of microarray gene expression patterns in a reduced dimensional space. *Genome research*, 12(7):1112–1120, July 2002.
- [Mur02] David Murphy. Gene expression studies using microarrays: Principles, problems and prospects. *Advances in Physiology Education*, 26(4):256–270, December 2002.

- [MWB11] Sebastián Maldonado, Richard Weber, and Jayanta Basak. Simultaneous feature selection and classification using kernel-penalized support vector machines. *Inf. Sci.*, 181(1):115–128, 2011.
- [MWLN09] Carl Murie, Owen Z. Woody, Anna Y. Lee, and Robert Nadon. Comparison of small n statistical tests of differential expression applied to microarrays. *BMC Bioinformatics*, 10, 2009.
- [Nat13] National Cancer Institute. What you need to know about leukemia, 2013.
- [NM09] Vladimir Nikulin and Geoffrey J. McLachlan. Penalized principal component analysis of microarray data. In Francesco Masulli, Leif E. Peterson, and Roberto Tagliaferri, editors, *CIBB*, volume 6160 of *Lecture Notes in Computer Science*, pages 82–96. Springer, 2009.
- [OKZ⁺09] Hideho Okada, Gary Kohanbash, Xinmei Zhu, Edward R. Kastenhuber, Aki Hoji, Ryo Ueda, and Mitsugu Fujita. Immunotherapeutic approaches for glioma. *Critical Reviews & trade; in Immunology*, 29(1):1–42, 2009.
- [OS10] A. Osareh and B. Shadgar. Machine learning techniques to diagnose breast cancer. In *Health Informatics and Bioinformatics (HIBIT), 2010 5th International Symposium on*, pages 114–120, 2010.
- [OT03] C. H. Ooi and Patrick Tan. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. *Bioinformatics*, (1):37–44, 2003.
- [OV12] C. Orsenigo and C. Vercellis. An effective double-bounded tree-connected isomap algorithm for microarray data classification. *Pattern Recognition Letters*, 33(1):9 – 16, 2012.
- [PAV14] Mahalakshmi Palani, R Arunkumar, and Arrambakam Janardhanam Vanisree. Methylation and expression patterns of tropomyosin-related kinase genes in different grades of glioma. *Neuromolecular medicine*, 16(3):529–539, 2014.

- [PBZ⁺06] Amela Prelic, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Buhlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele, and Eckart Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 2006.
- [Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.
- [PFS11] A. Posekany, K. Felsenstein, and P. Sykacek. Biological assessment of robust noise models in microarray data analysis. *Bioinformatics*, 2011.
- [PKMJ07] J. Pratik, K. Konstantinos, B. Michael, and S. Joseph. A perspective on microarrays: current applications, pitfalls, and potential uses. *Microbial Cell Factories*, 2007.
- [PM12] M. Perez and T. Marwala. Microarray data feature selection using hybrid genetic algorithm simulated annealing. *Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of*, pages 1 – 5, 2012.
- [PMQC11] Kate Patterson, Laura Molloy, Wenjia Qu, and Susan Clark. DNA methylation: bisulphite modification and analysis. *Journal of visualized experiments : JoVE*, (56), 2011.
- [POH⁺10] Konstantin Pentchev, Keiichiro Ono, Ralf Herwig, Trey Ideker, and Atanas Kamburov. Evidence mining and novelty assessment of protein-protein interactions with the consensusPathDB plugin for cytoscape. *Bioinformatics*, 26(21):2796–2797, 2010.
- [PSYK12] Brent S. Pedersen, David A. Schwartz, Ivana V. Yang, and Katherina J. Kechris. Comb-p: software for combining, analyzing, grouping and correcting spatially correlated p-values. *Bioinformatics*, 28(22):2986–2988, 2012.

- [PTOK00] William D. Pennie, Jonathan D. Tugwood, Gerry J. A. Oliver, and Ian Kimber. The principles and practice of toxicogenomics: Applications and opportunities. *Toxicological Sciences*, 54(2):277–283, 2000.
- [PVG⁺11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [PYK⁺03] Taesung Park, Sung-Gon Yi, Sung-Hyun Kang, Seung Yeoun Lee, Yong-Sung Lee, and Richard Simon. Evaluation of normalization methods for microarray data. *BMC Bioinformatics*, 4:33, 2003.
- [RB12] N. Revathy and R. Balasubramanian. GA-SVM wrapper approach for gene ranking and classification using expression of very few genes. *Journal of Theoretical and Applied Information Technology*, 40(2):113 – 119, 2012.
- [Rei10] Mark Reimers. Making informed choices about microarray data analysis. *PLoS Comput Biol*, 6(5):e1000786, 05 2010.
- [Res95] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1, IJCAI’95*, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
- [RFAM13] Fakoor R, Ladhak F, Nazi A, and Huber M. Using deep learning to enhance cancer diagnosis and classification. In *Proceedings of the ICML Workshop on the Role of Machine Learning in Transforming Healthcare (WHEALTH)*, ICML, 2013.
- [RGS08] D. Rossell, R. Guerra, and C. Scott. Semi-parametric differential expression analysis via partial mixture estimation. *Statistical Applications in Genetics and Molecular Biology*, 7(1), 2008. article 15.

- [RJFD10] Andrea Rau, Florence JaffrÃ©zic, Jean-Louis Foulley, and Rebecca W Dorge. An empirical bayesian method for estimating biological networks from temporal microarray data. *Stat Appl Genet Mol Biol*, 9:Article 9, 2010.
- [RJG⁺12] Jianhua Ruan, Md. Jamiul Jahid, Fei Gu, Chengwei Lei, Yi-Wen Huang, Ya-Ting Hsu, David G. Mutch, Chun-Liang Chen, Nameer B. Kirma, and Tim H. Huang. Network-based classification of recurrent endometrial cancers using high-throughput DNA methylation data. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, BCB '12, pages 418–425, New York, NY, USA, 2012. ACM.
- [RLM12] Jordi Remon, Pilar Lianes, and Susana Martinez. Brain metastases from renal cell carcinoma. should we change the current standard? *Cancer Treatment Reviews*, 38(4):249 – 257, 2012.
- [RMM] Vernon Bond Jr. Georges E. Haddad Richard M. Millis, M. Sadegh Asadi.
- [Rou87] Peter Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, November 1987.
- [RRAR06] Roberto Ruiz, Jos  C. Riquelme, and Jes  S. Aguilar-Ruiz. Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recogn.*, 39(12):2383–2392, December 2006.
- [RS00] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.
- [RSA00] Soumya Raychaudhuri, Joshua M. Stuart, and Russ B. Altman. Principal components analysis to summarize microarray experiments: Application to sporulation time series. In *in Pacific Symposium on Biocomputing*, pages 452–463, 2000.

- [RVO14] Ferran Reverter, Esteban Vegas, and Josep Oller. Kernel-PCA data integration with enhanced interpretability. *BMC Systems Biology*, 8(Suppl 2):S6, 2014.
- [RW98] R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*. Springer Verlag, Heidelberg, Berlin, New York, 1998.
- [RWL00] Dan R. Robinson, Yi-Mi Wu, and Su-Fang Lin. The protein tyrosine kinase family of the human genome. *Oncogene*, 19(49):5548–5557, November 2000.
- [RZD⁺07] Franck Rapaport, Andrei Zinovyev, Marie Dutreix, Emmanuel Barillot, and Jean-Philippe Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8, 2007.
- [Sac96] L Sachs. The control of hematopoiesis and leukemia: from basic biology to the clinic. *Proc Natl Acad Sci U S A*, 93(10):4742–9, 1996.
- [SB08] Miho Suzuki and Adrian Bird. DNA methylation landscapes: provocative insights from epigenomics. *Nat Rev Genet*, 9(6):465–476, June 2008.
- [SDRL06] Andreas Schlicker, Francisco S. Domingues, Jorg RahnenfÄijhrer, and Thomas Lengauer. A new measure for functional similarity of gene products based on gene ontology. *BMC Bioinformatics*, 7:302, 2006.
- [SDVDVDM⁺03] E. H. A. Spaenij-Dekking, J. Van Delft, E. Van Der Meijden, H. S. Hiemstra, J. H. F. Falkenburg, F. Koning, J. W. Drijfhout, and J. C. Kluin-Nelemans. Synaptojanin 2 is recognized by HLA class ii-restricted hairy cell leukemia-specific t cells. *Leukemia*, 17(12):2467–73, 2003.
- [Sei14] Karen Seiter. Acute myelogenous leukemia, June 2014.
- [SH07] Maria Stalteri and Andrew Harrison. Interpretation of multiple probe sets mapping to the same gene in Affymetrix genechips. *BMC Bioinformatics*, 8(1):13, 2007.

- [SHPF94] Susan, Janet Harrison, Cheryl L. Paul, and Marianne Frommer. High sensitivity mapping of methylated cytosines. *Nucl. Acids Res.*, 22(15):2990–2997, August 1994.
- [SILn07] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, September 2007.
- [SK10] Gregor Stiglic and Peter Kokol. Stability of ranked gene lists in large microarray analysis studies. *BioMed Research International*, 2010, 2010.
- [SKFJ05] Eran Segal, Daphne Koller, Nir Friedman, and Tommi Jaakkola. Learning module networks. In *Journal of Machine Learning Research*, pages 525–534, 2005.
- [SKK⁺06] Axel Schumacher, Philipp Kapranov, Zachary Kaminsky, James Flanagan, Abbas Assadzadeh, Patrick Yau, Carl Virtanen, Neil Winegarten, Jill Cheng, Thomas Gingeras, and Arturas Petronis. Microarray-based DNA methylation profiling: technology and applications. *Nucleic acids research*, 34(2):528–542, 2006.
- [SLW97] Diane M. Strong, Yang W. Lee, and Richard Y. Wang. Data quality in context. *Commun. ACM*, 40(5):103–110, May 1997.
- [SPH07] B. Schölkopf, J. Platt, and T. Hofmann. *MLLE: Modified Locally Linear Embedding Using Multiple Weights*, pages 1593–1600. MIT Press, 2007.
- [SPT⁺01] T. Sørlie, C. M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, T. Thorsen, H. Quist, J. C. Matese, P. O. Brown, D. Botstein, P. Eystein Lønning, and A. L. Børresen-Dale. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Natl. Acad. Sci. U.S.A.*, 98(19):10869–74, 2001.

- [SS11] Catherine C. Smith and Neil P. Shah. Tyrosine kinase inhibitor therapy for chronic myeloid leukemia: Approach to patients with treatment-naive or refractory chronic-phase disease. *ASH Education Program Book*, 2011(1):121–127, 2011.
- [SSB96] D. Shalon, S. J. Smith, and P. O. Brown. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Res*, 6(7):639–45, July 1996.
- [SSM99] Bernhard Schölkopf, Alexander J. Smola, and Klaus R. Müller. Kernel principal component analysis. *Advances in kernel methods: support vector learning*, pages 327–352, 1999.
- [SSZ⁺98] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. of the Cell*, 9:3273–3297, 1998.
- [SSZ⁺13] Lihong Shi, Xiuning Sun, Jin Zhang, Chunling Zhao, Hongli Li, Zhi-jun Liu, Chunyan Fang, Xuejian Wang, Chunzhen Zhao, Xiurong Zhang, Fenghua Zhou, Shijun Lu, Rongcheng Luo, and Baogang Zhang. Gab2 expression in glioma and its implications for tumor invasion. *Acta Oncologica*, 52(8):1739–1750, 2013.
- [Suk15] Nimalesh Sukumar. Bayesian network analysis of cancer microarray data, September 2015.
- [SYD10] Junjie Su, Byung-Jun Yoon, and Edward R. Dougherty. Identification of diagnostic subnetwork markers for cancer in human protein-protein interaction network. *BMC Bioinformatics*, 11(S-6):S8, 2010.
- [TdSL00] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, December 2000.

- [TFG⁺07] Thomas Tang, Nicolas Francois, Annie Glatigny, Nicolas Agier, Marie-Helene Mucchielli, Lawrence Aggerbeck, and Herve Delacroix. Expression ratio evaluation in two-colour microarray experiments is significantly improved by correcting image misalignment. *Bioinformatics*, 23(20):2686–2691, 2007.
- [Tho15] W. M. Thorburn. Occam’s razor. *Mind*, 24(2):287–288, 1915.
- [TL11] Laura Tolosi and Thomas Lengauer. Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics*, 27(14):1986–1994, 2011.
- [TLL08] Jengnan Tzeng, Henry Horng-Shing Lu, and Wen-Hsiung Li. Multidimensional scaling for large genomic data sets. *BMC Bioinformatics*, 9, 2008.
- [TOR⁺01] George C. Tseng, Min-Kyu Oh, Lars Rohlin, James C. Liao, and Wing Hung Wong. Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects. *Nucleic Acids Research*, 29(12):2549–2557, 2001.
- [TP07] Feng Tai and Wei Pan. Incorporating prior knowledge of predictors into penalized classifiers with multiple penalty terms. *Bioinformatics*, 23(14):1775–1782, 2007.
- [TSY05] E. Ke Tang, Ponnuthurai N. Suganthan, and Xin Yao. Feature selection for microarray data using least squares SVM and particle swarm optimization. In *CIBCB*, pages 9–16. IEEE, 2005.
- [TSY06] E. Ke Tang, Ponnuthurai N. Suganthan, and Xin Yao. Gene selection algorithms for microarray data based on least squares support vector machine. *BMC Bioinformatics*, 7:95, 2006.
- [TTC01] Virginia Goss Tusher, Robert Tibshirani, and Gilbert Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, 98(9):5116–5121, 2001.

- [TWZ⁺12] Xiaochao Tan, Shan Wang, Liyuan Zhu, Chao Wu, Bin Yin, Jizong Zhao, Jiangang Yuan, Boqin Qiang, and Xiaozhong Peng. cAMP response element-binding protein promotes gliomagenesis by modulating the expression of oncogenic microRNA-23a. *Proceedings of the National Academy of Sciences*, 109(39):15805–15810, 2012.
- [TZH07] Yuchun Tang, Yan-Qing Zhang, and Zhen Huang. Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 4(3):365–381, 2007.
- [Van07] Vantage - Vanderbilt technologies for Advanced Genomics. Normalization, 2007.
- [VBL94] J.Y. Delattre F. Vega M. Poisson B. Dutrillaux V. Bardot, A.M. Dutrillaux and C. Luccionil. Purine and pyrimidine metabolism in human gliomas: relation to chromosomal aberrations. *British Journal of Cancer - Nature*, 70:212–218, 1994.
- [VP13] Nams Vo and Vinhthuy Phan. Exploiting dependencies of patterns in gene expression analysis using pairwise comparisons. In Zhipeng Cai, Oliver Eulenstein, Daniel Janies, and Daniel Schwartz, editors, *Bioinformatics Research and Applications*, volume 7875 of *Lecture Notes in Computer Science*, pages 173–184. Springer Berlin Heidelberg, 2013.
- [WABD04] Haiying Wang, Francisco Azuaje, Olivier Bodenreider, and Joaquín Dopazo. Gene expression correlation and gene ontology-based similarity: an assessment of quantitative relationships. In *Computational Intelligence in Bioinformatics and Computational Biology, 2004. CIBCB'04. Proceedings of the 2004 IEEE Symposium on*, pages 25–31. IEEE, 2004.
- [WBG⁺14] James Wagner, Stephan Busche, Bing Ge, Tony Kwan, Tomi Pastinen, and Mathieu Blanchette. The relationship between DNA methylation, ge-

netic and expression inter-individual variation in untransformed human fibroblasts. *Genome Biology*, 15(2):R37+, February 2014.

- [WCW⁺13] Aleksandra Wandzilak, Mateusz Czyzycki, Pawel Wrobel, Magdalena Szczerbowska-Boruchowska, Edyta Radwanska, Dariusz Adamek, and Marek Lankosz. The oxidation states and chemical environments of iron and zinc as potential indicators of brain tumour malignancy grade - preliminary results. *Metallomics*, 5:1547–1553, 2013.
- [WDW⁺05] Michael Weber, Jonathan J. Davies, David Wittig, Edward J. Oakeley, Michael Haase, Wan L. Lam, and Dirk Schubeler. Chromosome-wide and promoter-specific analyses identify sites of differential DNA methylation in normal and transformed human cells. *Nature Genetics*, 37(8):853–862, July 2005.
- [Wei14] Mark E Weinblatt. Pediatric acute myelocytic leukemia, 2014.
- [WG05] Antai Wang and Edmund A. Gehan. Gene selection for microarray data analysis using principal component analysis. *Stat Med*, 24:2069–2087, 2005.
- [WH12] Cheryl Lyn Walker and Shuk-mei Ho. Developmental reprogramming of cancer susceptibility. *Nat Rev Cancer*, 12(7):479–86, 2012.
- [WI05] Zhijin Wu and Rafael A. Irizarry. Stochastic models inspired by hybridization theory for short oligonucleotide arrays. *Journal of Computational Biology*, 12(6):882–893, 2005.
- [Wil45] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, December 1945.
- [Wil13] Cheryl L. Willman. SH2B3: a new leukemia predisposition gene. *Blood*, 122(14):2293–2295, 2013.

- [WM04] Yuhang Wang and Fillia Makedon. Application of relief-f feature filtering algorithm to selecting informative genes for cancer classification using microarray data. In *CSB*, pages 497–498. IEEE Computer Society, 2004.
- [WS12] Gareth H. Williams and Kai Stoeber. The cell cycle and cancer. *J. Pathol.*, 226(2):352–364, January 2012.
- [WTH⁺05] Yu Wang, Igor V. Tetko, Mark A. Hall, Eibe Frank, Axel Facius, Klaus F.X. Mayer, and Hans W. Mewes. Gene selection from microarray data for cancer classification – a machine learning approach. *Computational Biology and Chemistry*, 29(1):37 – 46, 2005.
- [WVOM04] Griffin Weber, Staal Vinterbo, and Lucila Ohno-Machado. Multivariate selection of genetic markers in diagnostic classification. *Artif. Intell. Med.*, 31(2):155–167, June 2004.
- [XJK01] Eric P. Xing, Michael I. Jordan, and Richard M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608. Morgan Kaufmann, 2001.
- [XXL13] Xiao-Lei Xia, Huanlai Xing, and Xueqin Liu. Analyzing Kernel Matrices for the Identification of Differentially Expressed Genes. *PLoS ONE*, 8(12):e81683+, December 2013.
- [YB05] Ka Yeung and Roger Bumgarner. Correction: Multiclass classification of microarray data with repeated measurements: application to cancer. *Genome Biology*, 6(13):405, 2005.
- [YBS01] Yee Hwa Yang, Michael J. Buckley, and Terence P. Speed. Analysis of cDNA microarray images. *Briefings in Bioinformatics*, 2(4):341–349, 2001.
- [YS02] Y. H. Yang and T. Speed. Design issues for cDNA microarray experiments. *Nat. Rev. Genet.*, 3(8):579–588, 2002.

- [YZZZ10] Pengyi Yang, Bing Bing Zhou, Zhang Zili, and Albert Zomaya. A multi-filter enhanced genetic ensemble system for gene selection and sample classification of microarray data. *BMC Bioinformatics*, 11(Supl1:S5), January 2010.
- [ZBU01] E. Zwick, J. Bange, and A. Ullrich. *Receptor Tyrosine Kinase Signalling as a Target for Cancer Intervention Strategies*. Journal of Endocrinology Limited, 2001.
- [ZLS⁺06] X. Zhang, X. Lu, Q. Shi, X. Q. Xu, H. C. Leung, L. N. Harris, J. D. Iglehart, A. Miron, J. S. Liu, and W. H. Wong. Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics*, 7, 2006.
- [ZW04] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: a quantitative study of their impacts. *Artif. Intell. Rev.*, 22(3):177–210, November 2004.

Appendix A

Cell Cycle

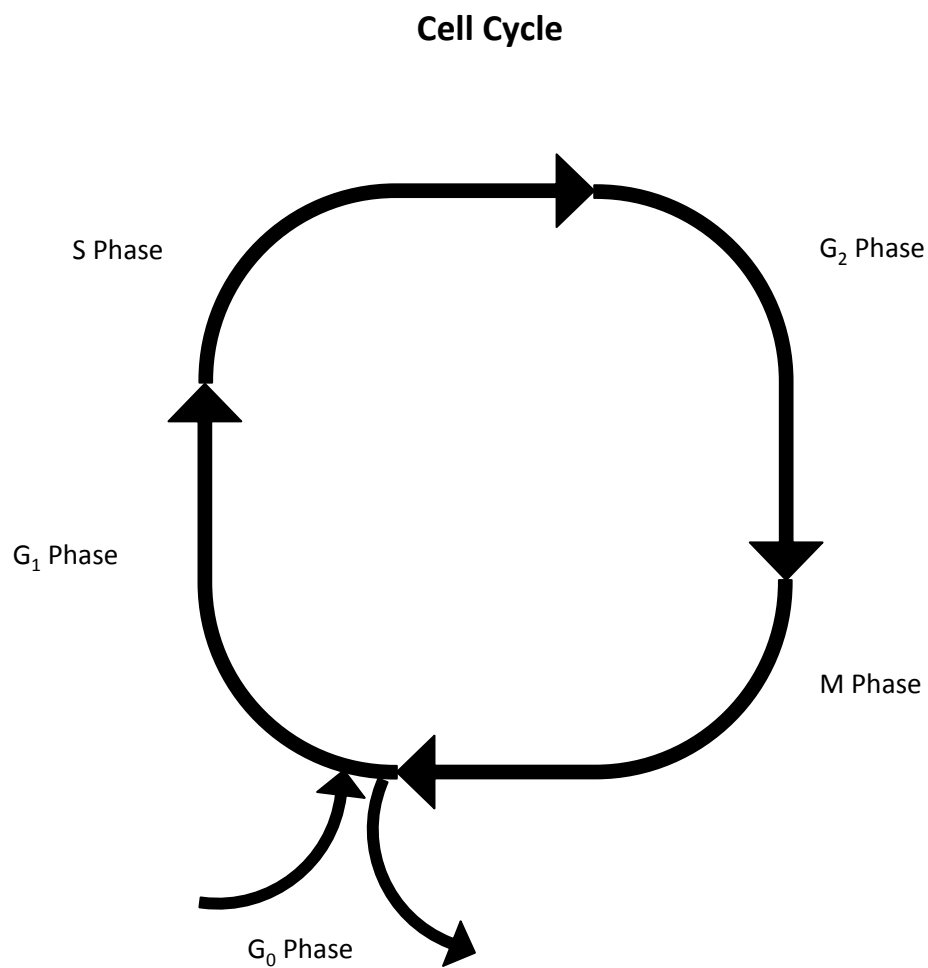


Figure A.1: Cell Cycle

Phase		Description
Interphase	G_0	Resting Phase: The cell performs normal functions
Interphase	G_1	Gap 1 - first growth phase: The cell prepares to start division
Interphase	S	Synthesis phase: The cell copies its DNA
Interphase	G_2	Gap 2 - second growth phase: The cell makes more proteins before it divides
Prophase	M	Mitosis: The cell divides into 2 new cells
Metaphase		
Anaphase		
Telophase		

Table A.1: The stages of the cell cycle ¹

¹<http://www.cancer.ca/en/cancer-information/cancer-101/what-is-cancer/the-cell-cycle/?region=on>

Appendix B

A priori Manifold Learning

B.1 Variances

The variances for the k -NN classifier for the gene-by-gene experiments are shown in table B.1 and for the sample-by-sample experiments in table B.2. For the LDA the variance is shown in table B.3 for the gene-by-gene experiments and in table B.4 for the sample-by-sample experiments. The variance was calculated using 10 fold cross-validation.

Type Of Cancer	A Priori Manifold Learning	Isomap	PCA
Breast cancer	32.09034e-5	37.52164e-5	35.38524e-5
Colon cancer	29.24537e-5	29.91476e-5	28.95183e-5
Kidney cancer	6.72999e-5	11.64989e-5	12.68591e-5
Ovary cancer	21.39207e-5	11.13463e-5	12.88114e-5
Lung cancer	14.09877e-5	5.26385e-5	3.13050e-5
Uterus cancer	13.01978e-5	3.44257e-5	5.51030e-5
Omentum cancer	2.54772e-5	0.80620e-5	0.80620e-5
Prostate cancer	2.34272e-5	6.79816e-5	4.34986e-5
Endometrium cancer	1.58922e-5	1.92059e-5	1.10440e-5

Table B.1: **10 Fold Cross Validation Variance On a Gene-by-Gene Transformation using k -Nearest Neighbours:** The results show that the variance of the cross validation is very small and thus we can safely compare the methods tested.

Type Of Cancer	A Priori Manifold Learning	Isomap	PCA
Breast cancer	32.09034e-5	27.91171e-5	18.32800e-5
Colon cancer	29.24537e-5	26.86585e-5	16.95718e-5
Kidney cancer	6.72999e-5	10.34294e-5	9.40982e-5
Ovary cancer	21.39207e-5	24.88867e-5	14.85025e-5
Lung cancer	14.09877e-5	14.62143e-5	12.39355e-5
Uterus cancer	13.01978e-5	16.97889e-5	18.60610e-5
Omentum cancer	2.54772e-5	2.79939e-5	2.12314e-5
Prostate cancer	2.34272e-5	2.07739e-5	2.17724e-5
Endometrium cancer	1.58922e-5	5.77868e-5	6.19262e-5

Table B.2: **10 Fold Cross Validation Variance On a Sample-by-Sample Transformation using k -Nearest Neighbours:** The results show that the variance of the cross validation is very small and thus we can safely compare the methods tested.

Type Of Cancer	A Priori Manifold Learning	Isomap	PCA
Breast cancer	4.43639e-5	3.18558e-5	1.64494e-5
Colon cancer	3.97728e-5	1.79713e-5	5.60684e-5
Kidney cancer	6.28824e-5	2.24769e-5	2.73758e-5
Ovary cancer	2.94021e-5	3.21893e-5	3.50449e-5
Lung cancer	1.58082e-5	2.14339e-5	1.26192e-5
Uterus cancer	1.04442e-5	7.45783e-5	7.01667e-5
Omentum cancer	1.21062e-5	3.76439e-5	2.42125e-5
Prostate cancer	4.12092e-5	1.40641e-5	4.41161e-5
Endometrium cancer	1.14222e-5	1.62528e-5	8.67444e-5

Table B.3: **10 Fold Cross Validation Variance On a Gene-by-Gene Transformation using Linear Discriminant Analysis:** The results show that the variance of the cross validation is very small and thus we can safely compare the methods tested.

Type Of Cancer	A Priori Manifold Learning	Isomap	PCA
Breast cancer	4.43639e-05	2.70937e-5	1.62271e-5
Colon cancer	3.97728e-5	3.55322e-5	5.32299e-5
Kidney cancer	6.28824e-5	4.56036e-5	3.06760e-5
Ovary cancer	2.94021e-5	2.80513e-5	4.17136e-5
Lung cancer	1.58082e-5	1.97068e-5	1.47822e-5
Uterus cancer	1.04442e-5	4.53130e-05	7.25349e-5
Omentum cancer	1.21062e-5	9.24128e-5	2.14339e-5
Prostate cancer	4.12092e-5	1.25679e-5	2.42809e-5
Endometrium cancer	1.14222e-5	8.63512e-5	8.75652e-5

Table B.4: **10 Fold Cross Validation Variance On a Sample-by-Sample Transformation using Linear Discriminant Analysis:** The results show that the variance of the cross validation is very small and thus we can safely compare the methods tested.

B.2 Accuracy error

For the k -NN gene-by-gene experiments the graphs are shown in figure B.1 and for the sample-by-sample in figure B.2. For the Linear Discriminant Analysis gene-by-gene experiments graphs are shown in figure B.3 and for the sample-by-sample in figure B.4. The accuracy was calculated using 10 fold cross-validation.

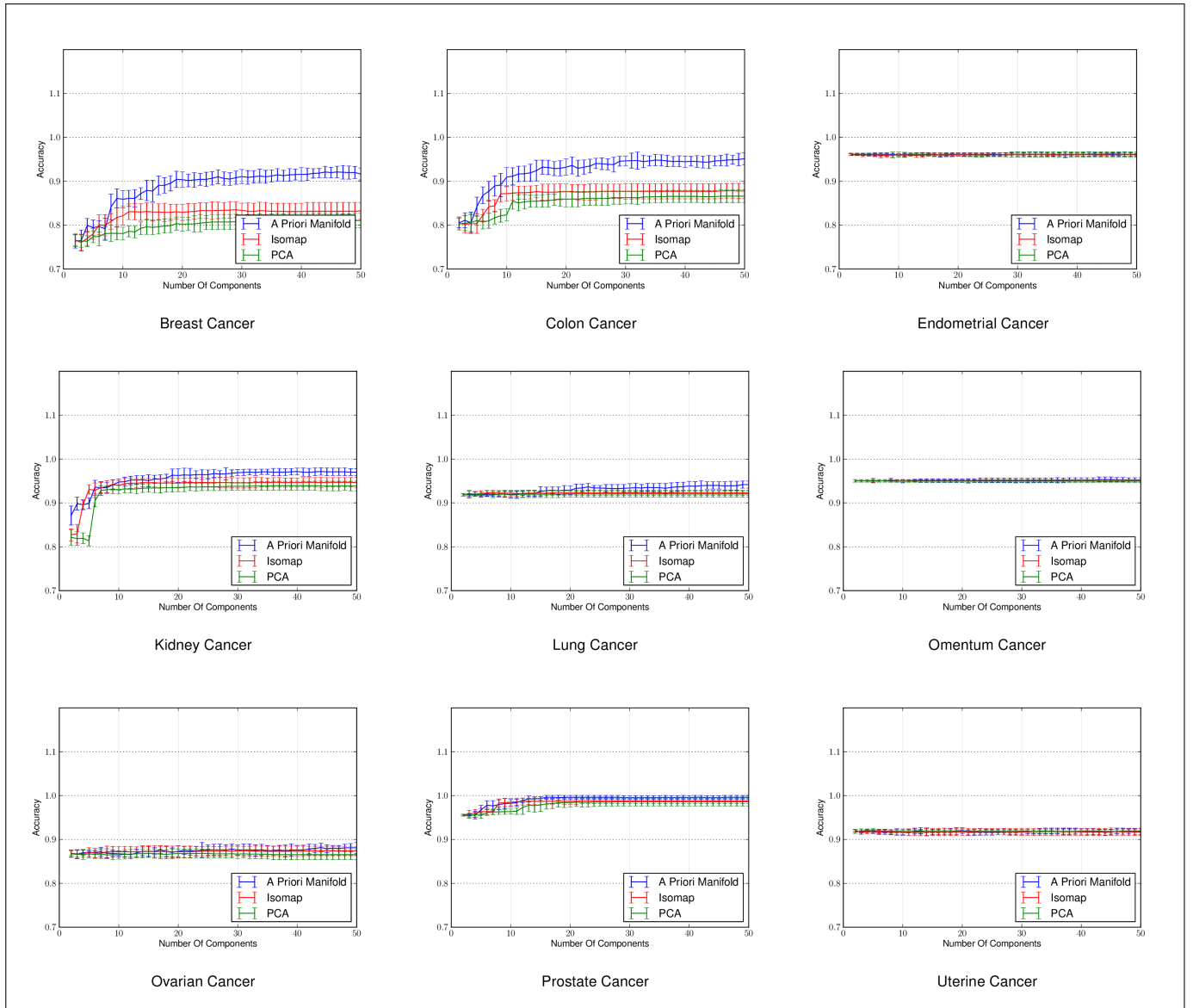


Figure B.1: **Accuracy with variance for all nine datasets using the gene-by-gene affinity matrices k -Nearest Neighbours:** Accuracy with variance calculated for *a priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red) computed using the gene-by-gene affinity matrix and the k -NN classifier

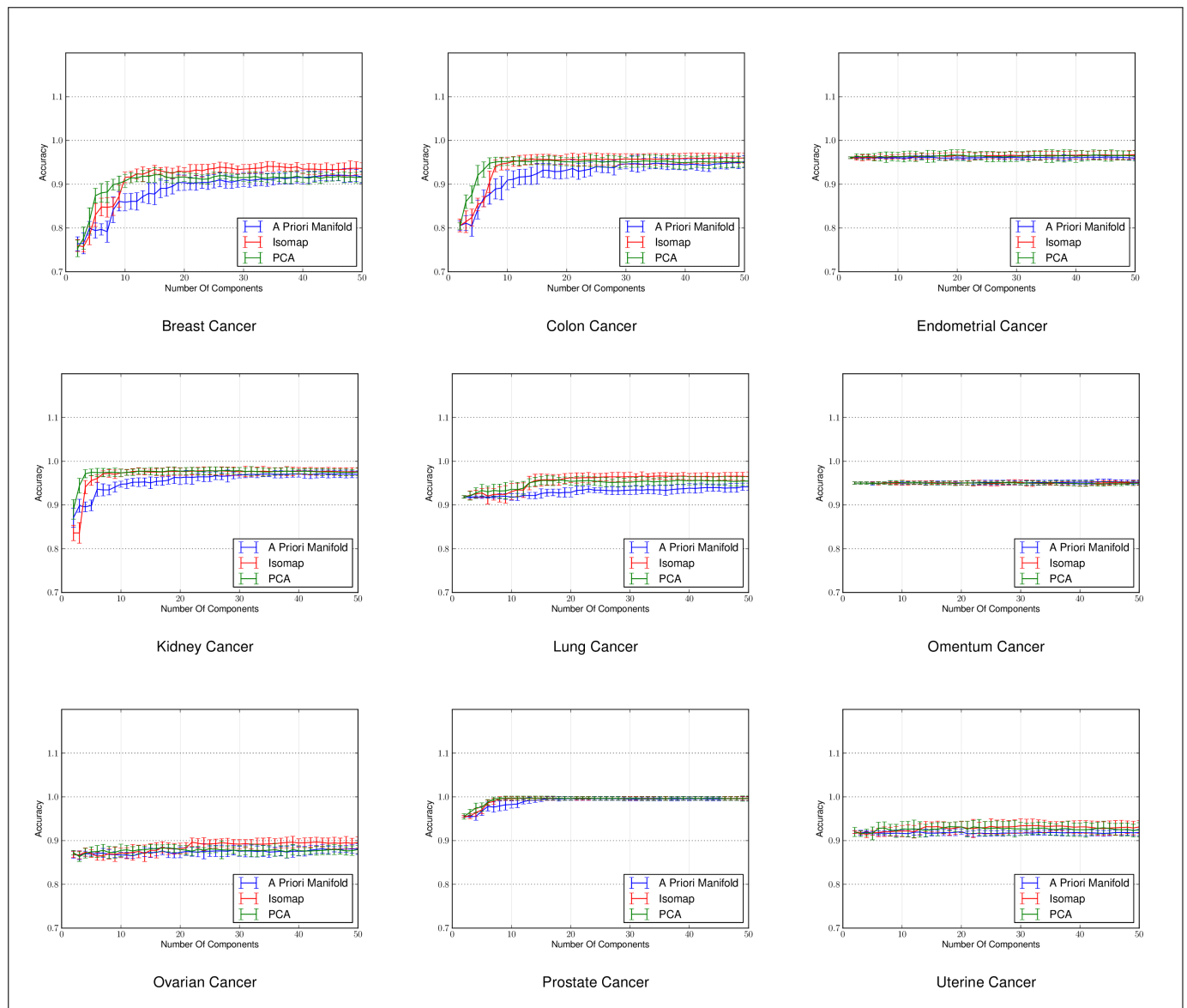


Figure B.2: **Accuracy with variance for all nine datasets using the sample-by-sample affinity matrices using k -Nearest Neighbours:** Accuracy with variance calculated for *a priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red) computed using the sample-by-sample affinity matrix and the k -NN classifier

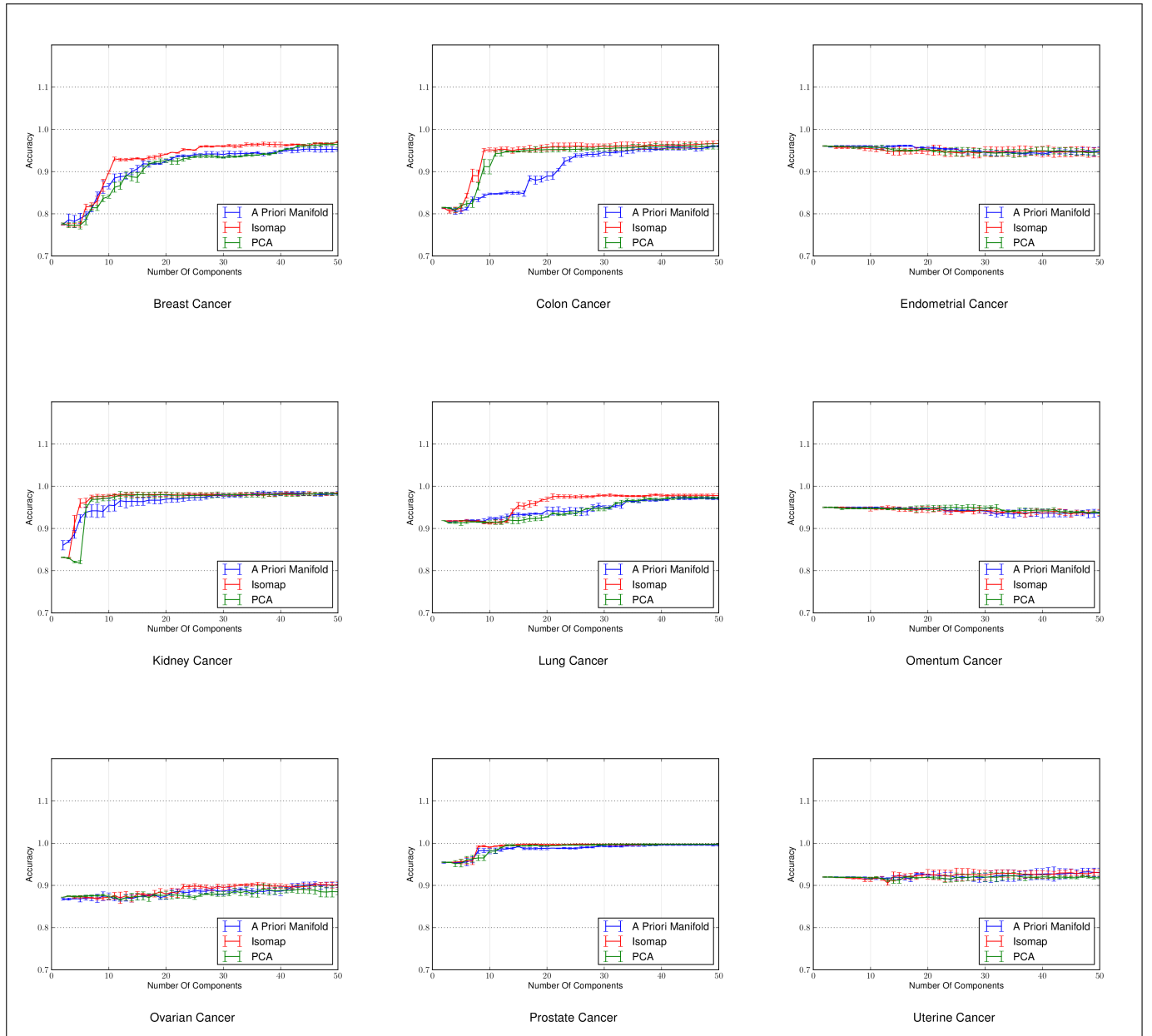


Figure B.3: **Accuracy with variance for all nine datasets using the gene-by-gene affinity matrices using Linear Discriminant Analysis:** Accuracy with variance calculated for *a priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red) computed using the gene-by-gene affinity matrix and the LDA classifier

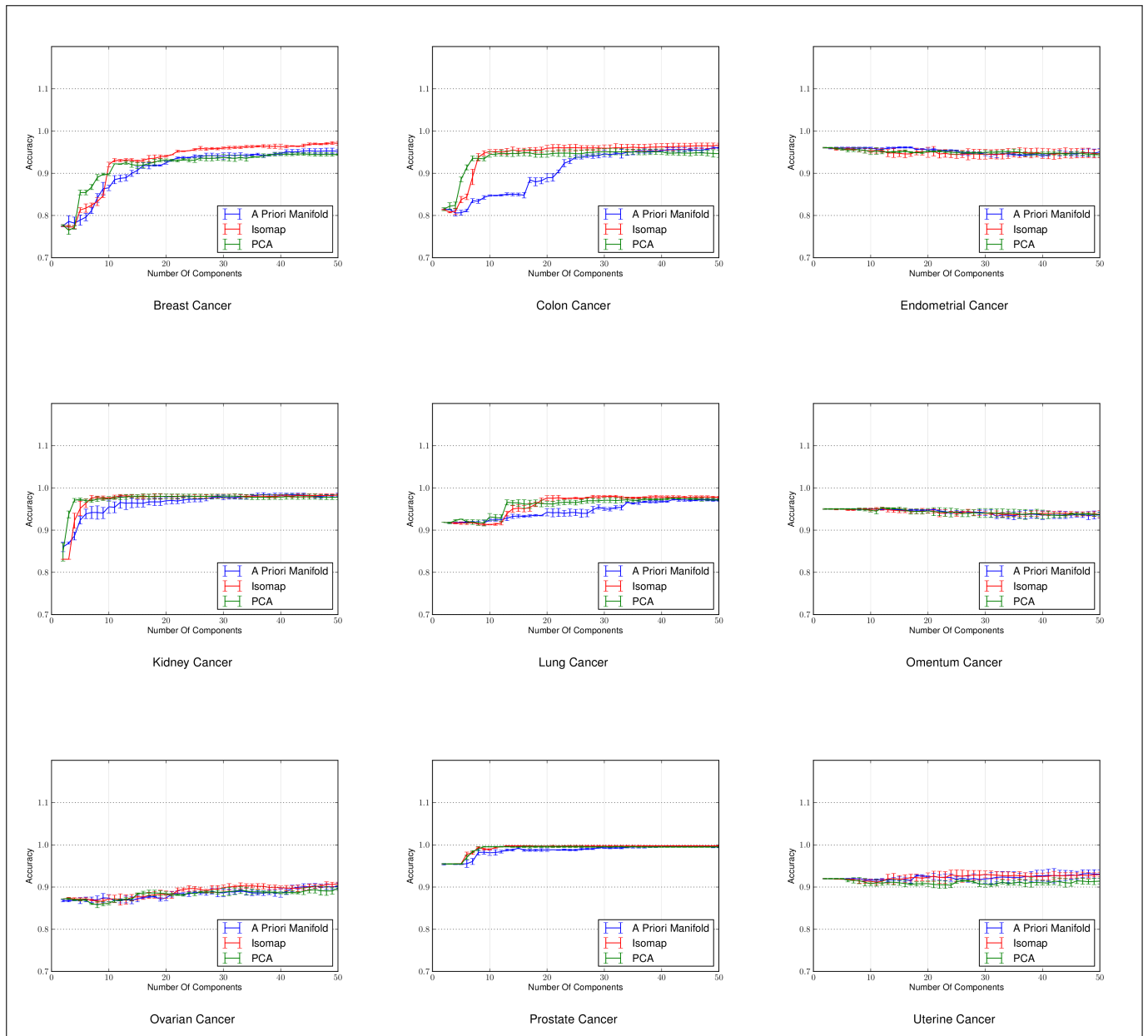


Figure B.4: Accuracy with variance for all nine datasets using the sample-by-sample affinity matrices using Linear Discriminant Analysis: Accuracy with variance calculated for *a priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red) computed using the sample-by-sample affinity matrix and the LDA classifier

B.3 Pathway Robustness

We show how the Dunn Index is affected in the Endometrium (figure B.5), Prostate (figure B.6) and Lung (figure B.7) datasets. We also show how the ROC curves are affected for Breast in figure B.8, Colon in figure B.9, Kidney in figure B.10, Omentum in figure B.11 and Ovary

in figure B.12.

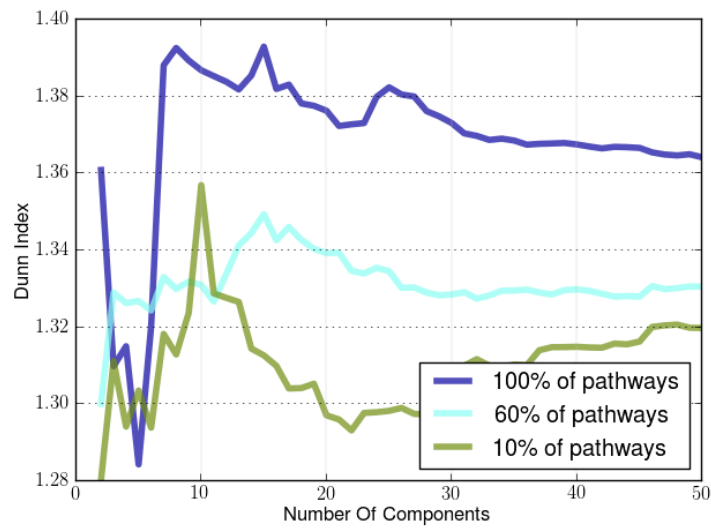


Figure B.5: **Pathway Robustness (Endometrium)**: A plot of the Dunn Index with different percentages of pathways

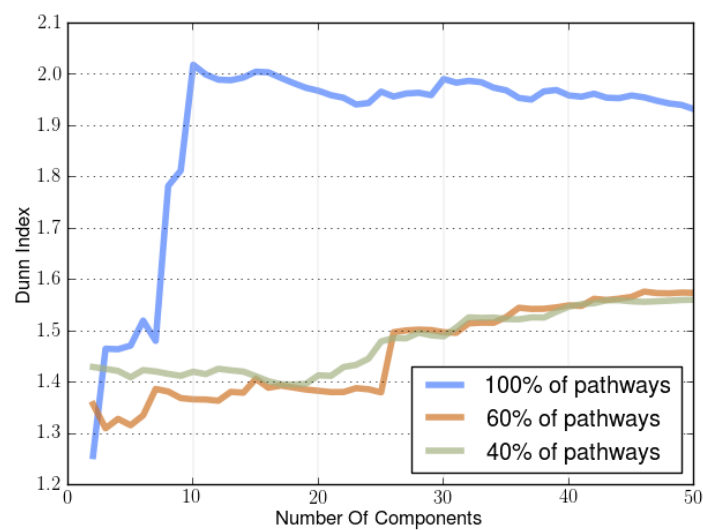


Figure B.6: **Pathway Robustness (Prostate)**: A plot of the Dunn Index with different percentages of pathways

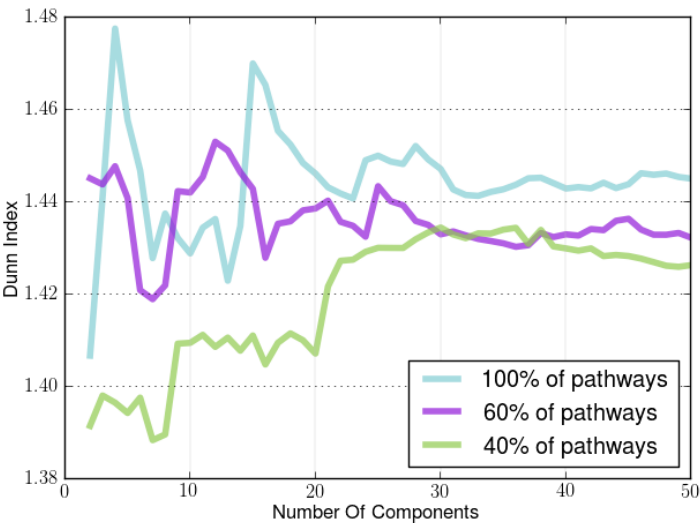


Figure B.7: **Pathway Robustness (Lung)**: A plot of the Dunn Index with different percentages of pathways

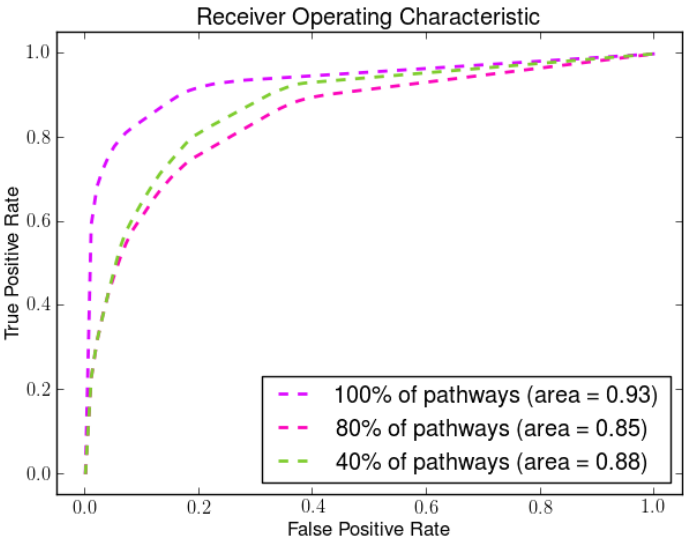


Figure B.8: **Pathway Robustness (Breast)**: A plot of ROC curves with different percentages of pathways

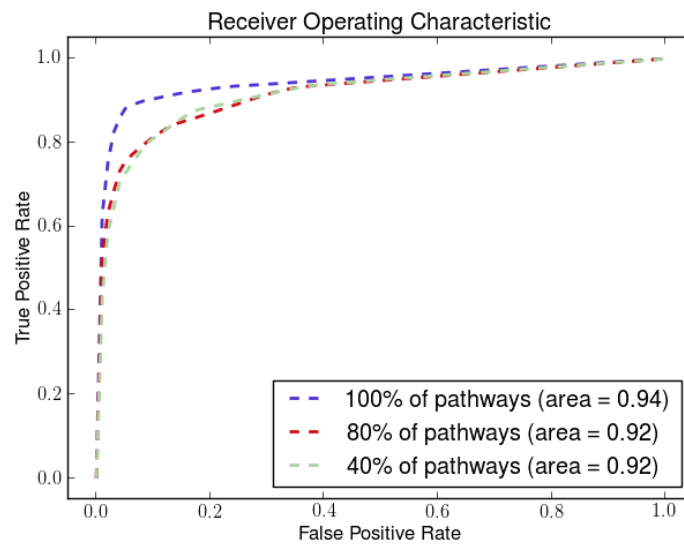


Figure B.9: **Pathway Robustness (Colon)**: A plot of ROC curves with different percentages of pathways

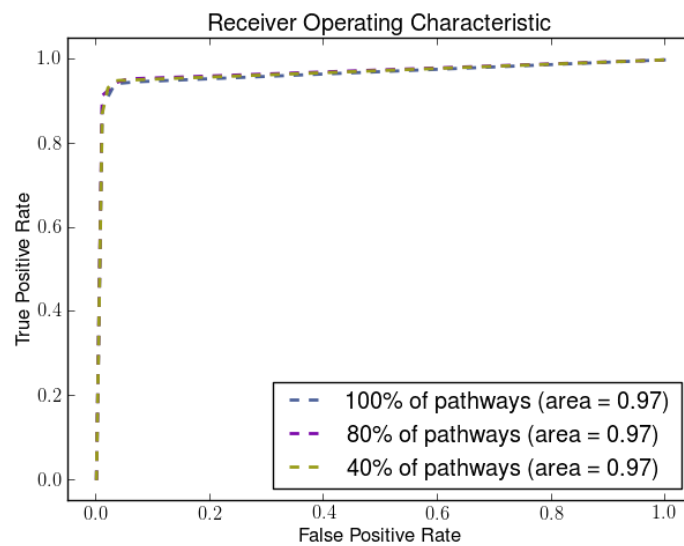


Figure B.10: **Pathway Robustness (Kidney)**: A plot of ROC curves with different percentages of pathways

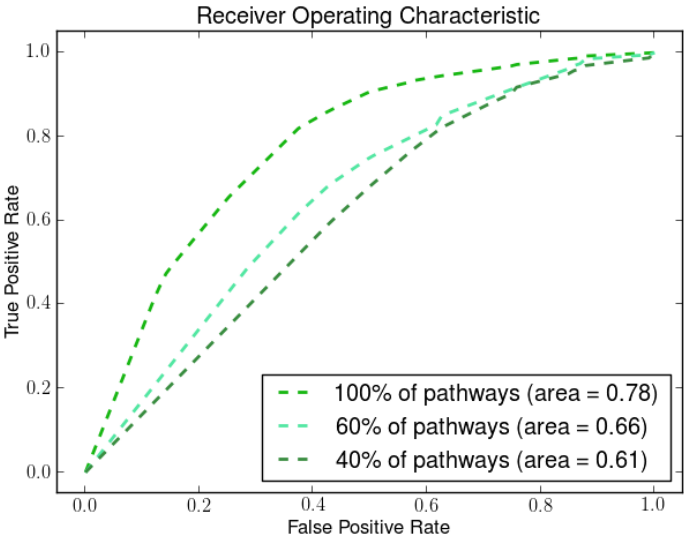


Figure B.11: **Pathway Robustness (Omentum)**: A plot of ROC curves with different percentages of pathways

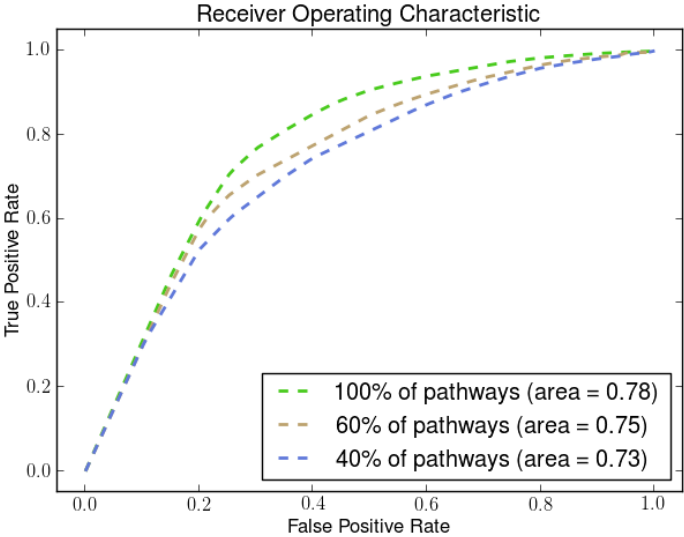


Figure B.12: **Pathway Robustness (Ovary)**: A plot of ROC curves with different percentages of pathways

Appendix C

A priori Manifold Learning using GO terms

C.1 ROC Curves

The ROC curves for the k -NN classifier for the gene-by-gene experiments are shown in figure C.1 and for the sample-by-sample experiments in figure C.2. For the LDA they are shown in figure C.3 for the gene-by-gene experiments and in figure C.4 for the sample-by-sample experiments. For the SVMs the gene-by-gene experiments can be found in figure C.5 for the sample-by-sample experiments in figure C.6. In all cases the *a priori* manifold learning method uses features from the gene-by-gene affinity matrix to build the manifold.

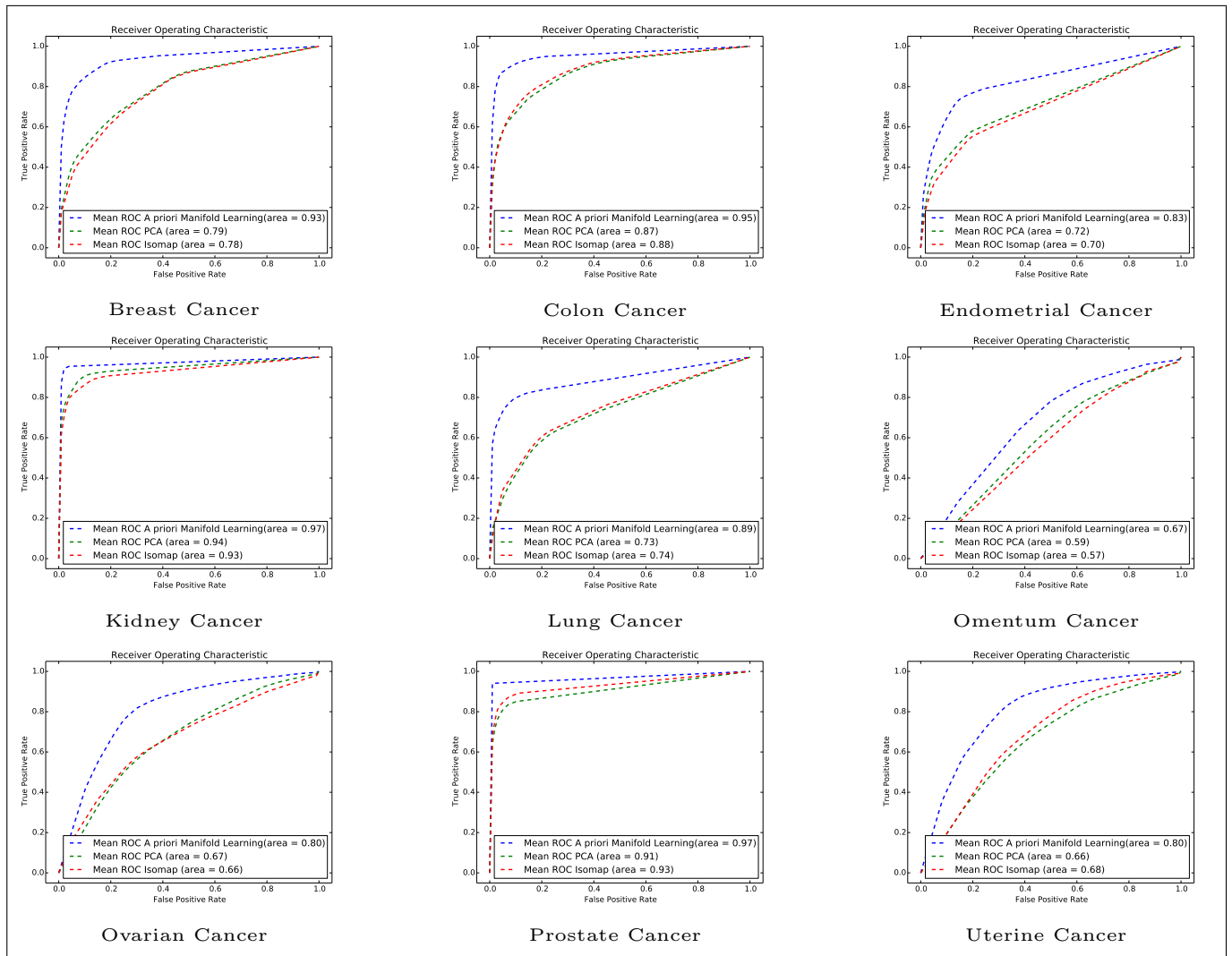


Figure C.1: **ROC curves for gene-by-gene affinity matrices using k -Nearest Neighbours:** ROC curves found for *a priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red) computed using the gene-by-gene affinity matrix and the k -NN classifier.

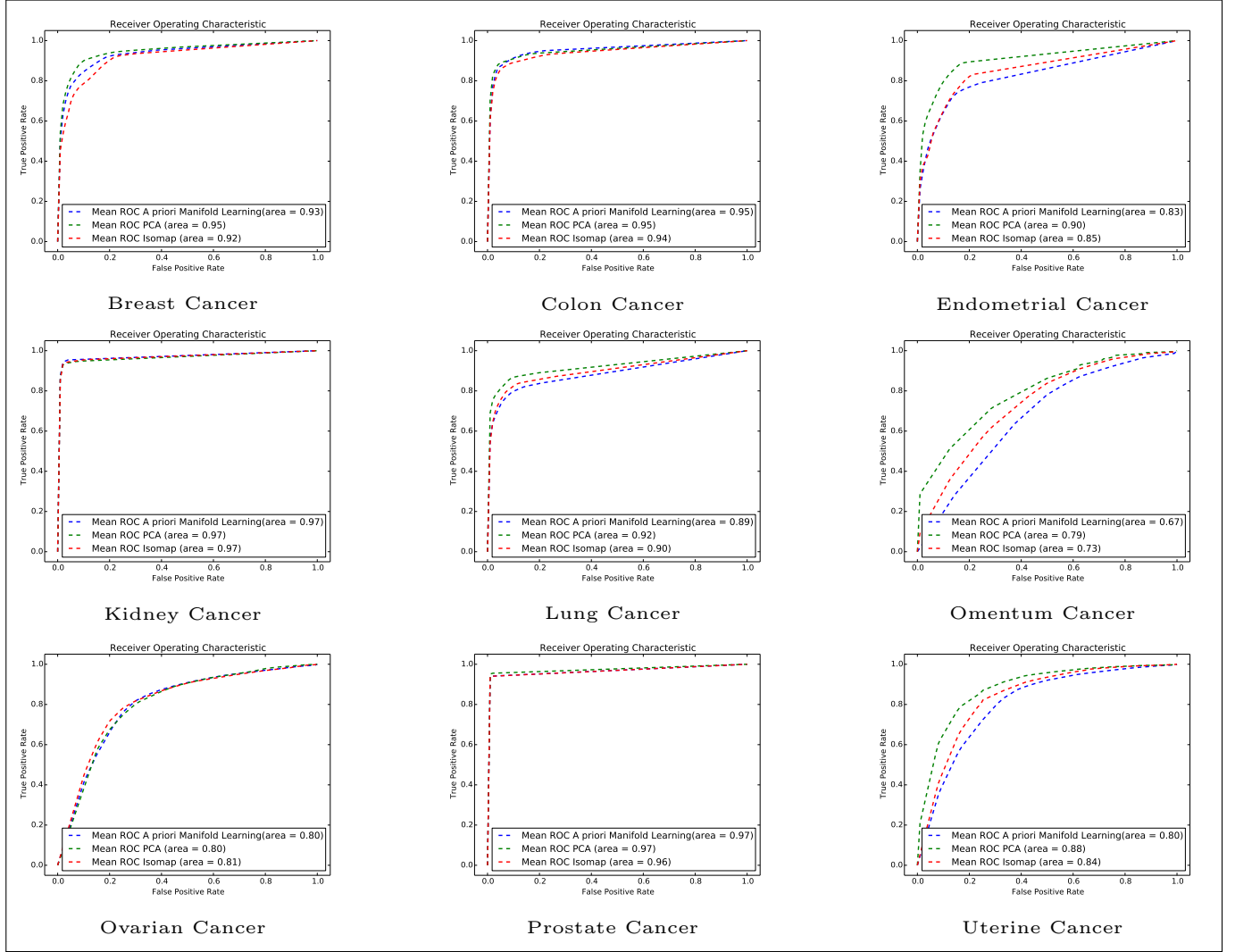


Figure C.2: ROC curves for sample-by-sample affinity matrices using k -Nearest Neighbours: ROC curves found for *a priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red) computed using the sample-by-sample affinity matrix and the k -NN classifier.

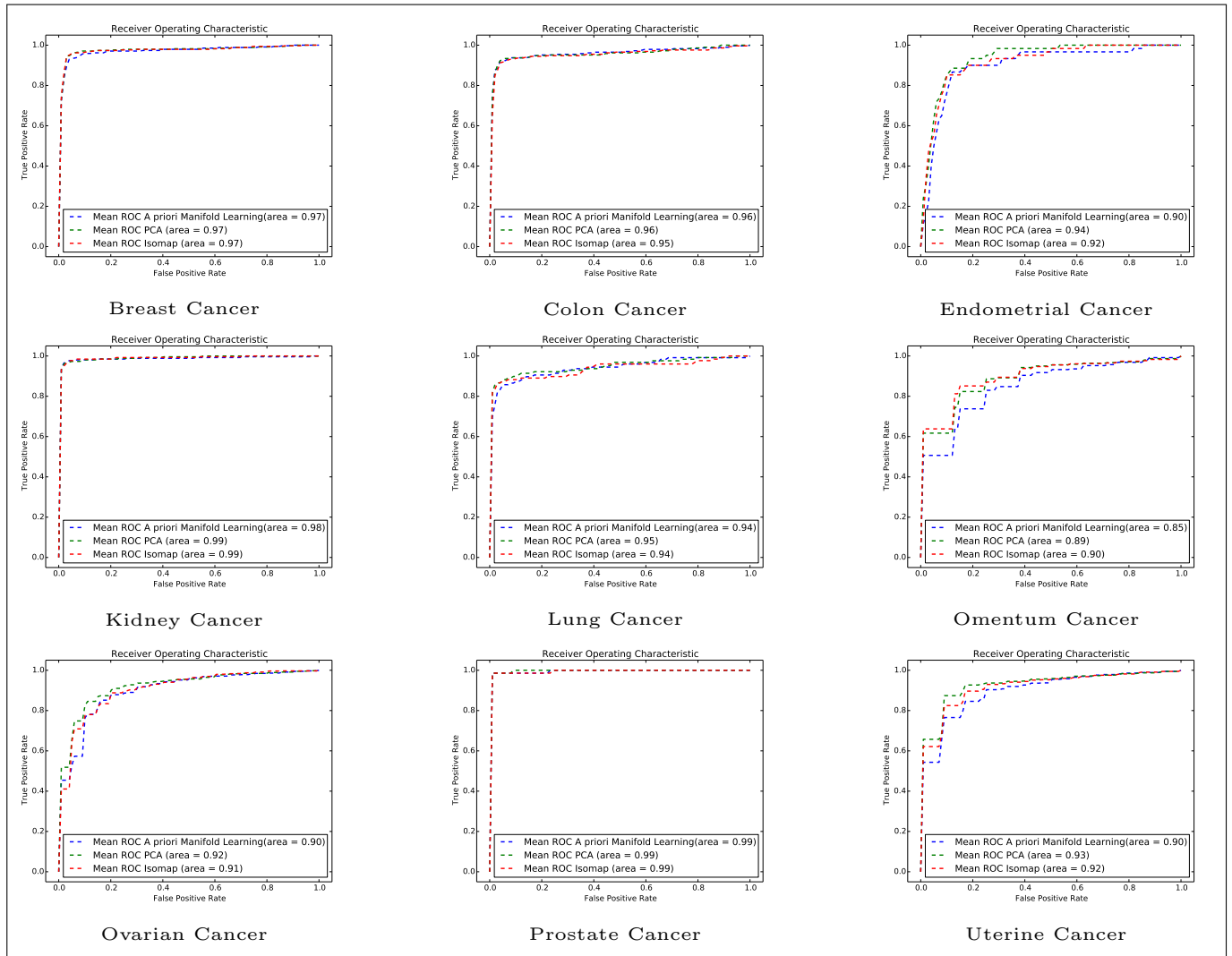


Figure C.3: ROC curves for gene-by-gene affinity matrices using Linear Discriminant Analysis: ROC curves found for *a priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red) computed using the gene-by-gene affinity matrix and Linear Discriminant Analysis.

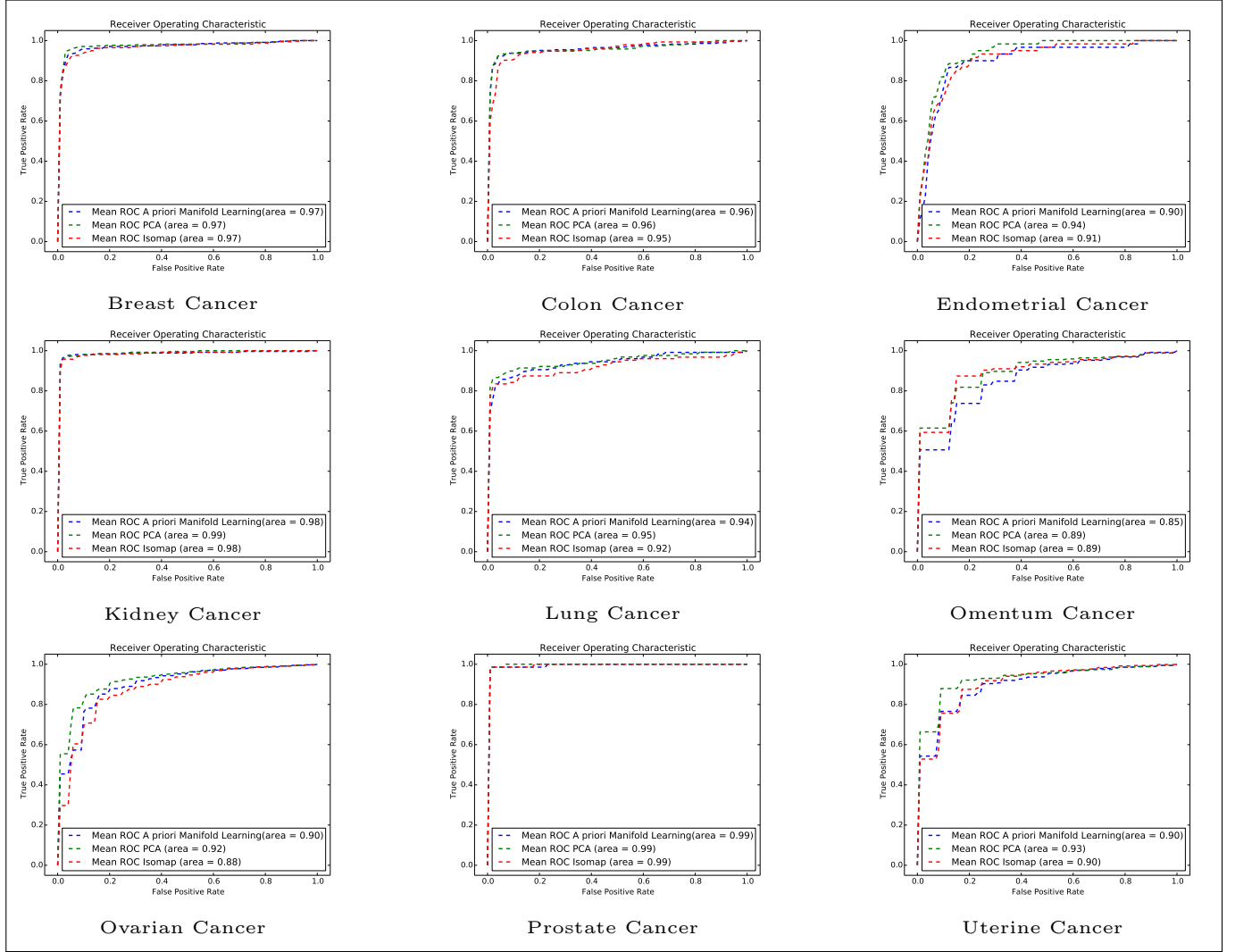


Figure C.4: ROC curves for sample-by-sample affinity matrices using Linear Discriminant Analysis: ROC curves found for *a priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red) computed using the sample-by-sample affinity matrix and Linear Discriminant Analysis.

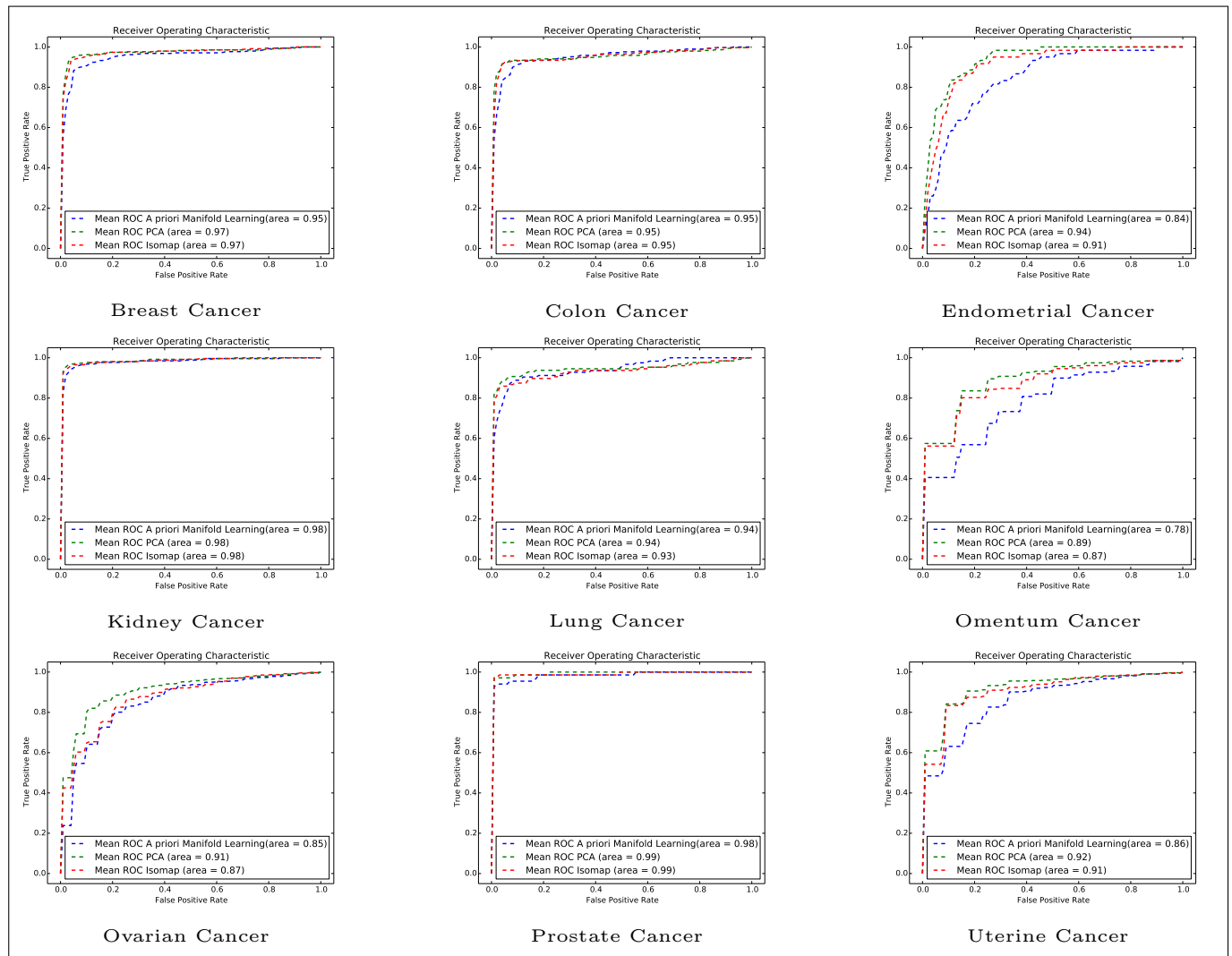


Figure C.5: ROC curves for gene-by-gene affinity matrices using Support Vector Machines: ROC curves found for *a priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red) computed using the gene-by-gene affinity matrix and Support Vector Machines.

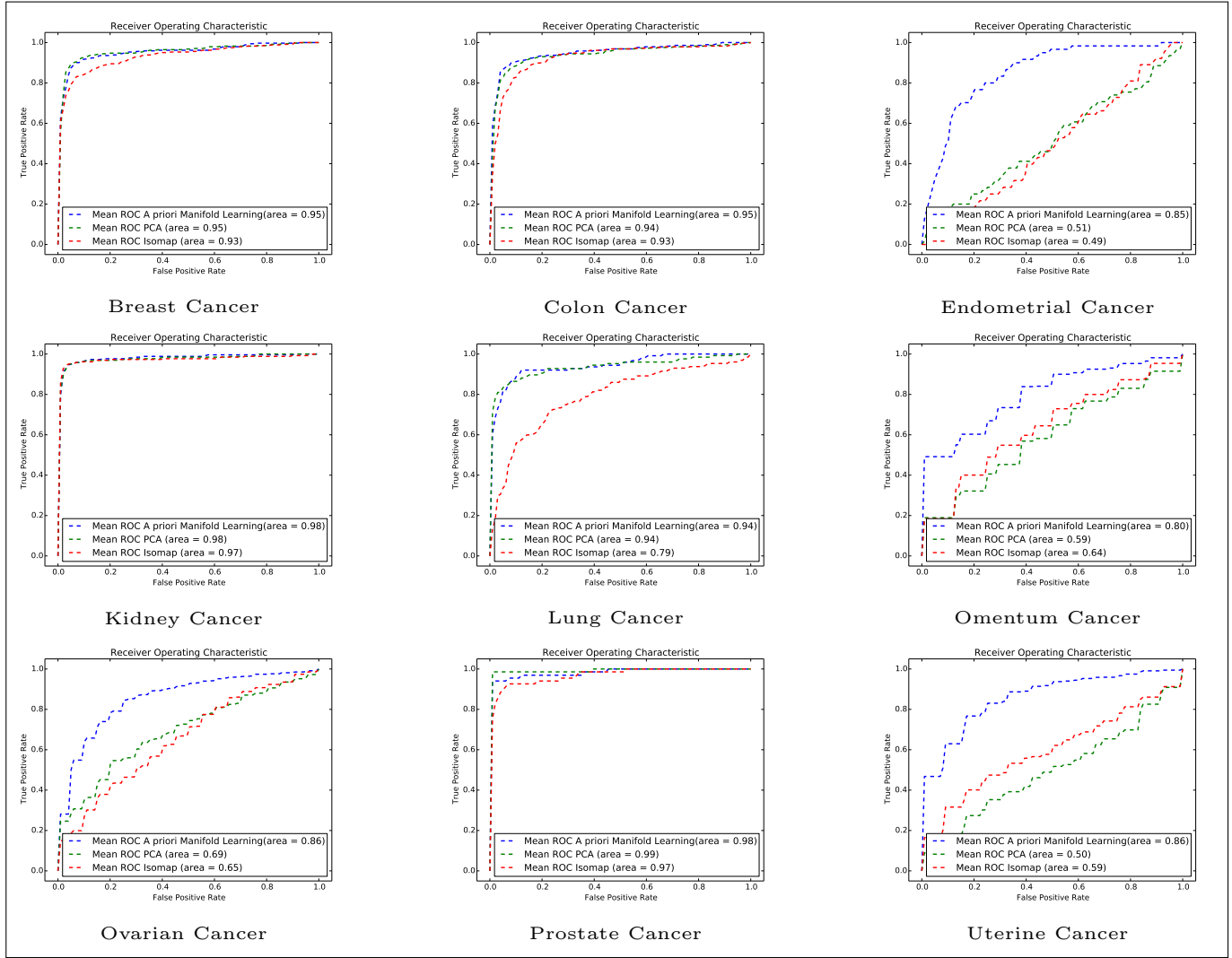


Figure C.6: ROC curves for sample-by-sample affinity matrices using Support Vector Machines: ROC curves found for *a priori* manifold learning (blue) compared with PCA (Green) and Isomap (Red) computed using the sample-by-sample affinity matrix and Support Vector Machines.

Appendix D

Acronyms

ALL: Acute Lymphoblastic Leukaemia

ANOVA: Analysis Of Variance

BDLDA : Block Diagonal Linear Discriminant Analysis

BIRS: Best Incremental Ranked Subset

BPFS: Biological Pathway - based Feature Selection Algorithm

CART: Classification And Regression Tree

cDNA : Complementary DNA

CFS: Correlation - Based Feature Selection

CML : Chronic Myeloid Leukaemia

DNA: Deoxyribonucleic Acid

EWUSC: Error - Weighted Uncorrelated Shrunk Centroid

GA: Genetic Algorithms

GA-SVM: Genetic Algorithm - Support Vector Machine

GLGS: Gradient - based - Leave - One - Out Gene Selection

GO: Gene Ontology

IC: Information Content

***k*-NN:** *k* - Nearest Neighbours

KEGG: Kyoto Encyclopedia of Genes and Genomes

LDA: Linear Discriminant Analysis

LLE: Locally - Linear Embedding

LOOCSFS: Leave - One - Out Calculation Sequential Forward Selection

LOOCV: Leave - One - Out Cross Validation

LOOE: Leave - One - Out Cross - Validation Error

LOWESS: Locally Weighted Scatter Plot Smoothing

MAQC: MicroArray Quality Control

MBD: Methyl - Binding Domain

MDS: Multi - Dimensional Scaling

mRNA: Messenger Ribonucleic Acid

PCA: Principal Components Analysis

PPI: Protein - To - Protein Interaction

R-SVM: Recursive - Support Vector Machine

RNA: Ribonucleic Acid

ROC: Receiver Operating Characteristic

SAMBA: Statistical - Algorithmic Method for Bicluster Analysis

SC: Shrunk Centroid

SFC: Sequential Forward Selection

SFS: Sequential Forward Selection

SNP: Single - Nucleotide Polymorphism

SOMs: Self Organizing Maps

SPCA: Supervised Principal Components Analysis

SVM: Support Vector Machine

SVM - RFE: Support Vector Machines - Recursive Feature Elimination

TCGA LGG: The Cancer Genome Atlas Lower Grade Glioma

USC: Uncorrelated Shrunk Centroid