Integrative Methods for Gene Data Analysis and Knowledge Discovery on the Case Study of KEDRI's Brain Gene Ontology

Yuepeng Wang

A thesis submitted to
Auckland University of Technology
in partial fulfilment of the requirements for the degree of
Master of Computer and Information Sciences

2008

School of Computing and Mathematical Sciences

Primary Supervisor: Nikola Kasabov

Abstract

In 2003, Pomeroy et al. published a research study that described a gene expression based prediction of central nervous system embryonal tumour (CNS) outcome. Over a half of decade, many models and approaches have been developed based on experimental data consisting of 99 samples with 7,129 genes. The way, how meaningful knowledge from these models can be extracted, and how this knowledge for further research is still a hot topic. This thesis addresses this and has developed an information method that includes modeling of interactive patterns, important genes discovery and visualisation of the obtained knowledge. The major goal of this thesis is to discover important genes responsible for CNS tumour and import these genes into a well structured knowledge framework system, called Brain-Gene-Ontology.

In this thesis, we take the first step towards finding the most accurate model for analysing the CNS tumour by offering a comparative study of global, local and personalised modeling. Five traditional modeling approaches and a new personalised method – WWKNN (weighted distance, weighted variables K-nearest neighbours) – are investigated. To increase the classification accuracy, an one-vs.-all based signal-to-noise ratio is also developed for pre-processing experimental data.

For the knowledge discovery, CNS-based ontology system is developed. Through ontology analysis, 21 discriminant genes are found to be relevant for different CNS tumour classes, medulloblastoma tumour subclass and medulloblastoma treatment outcome. All the findings in this thesis contribute for expanding the information space of the BGO framework.

Acknowledgements

Firstly I would like to thank my primary supervisor Prof. Nikola Kasabov for providing me the opportunity to start this project, and also for his guidance and patience throughout the course of this thesis.

I am especially thankful to my secondary supervisor Dr. Lubica Benuskova for her guidance and helpful assistance throughout the researching and the writing of the thesis.

I would like to thank the Knowledge Engineering & Discovery Research Institute for providing a pleasant working environment and great technical support. My special thanks goes to Stefan Schliebs and Yingjie Hu. Their great proof-reading and editing advice make my thesis far more understandable.

Finally I am indebted to my parents and family for encouragement and support throughout.

Contents

Abstract i

Acknowledgements ii

Table of Abbreviations vii

List of Figures xi

List of Tables xiv

1 Introduction 1

1.1 Research Background 1

1.1.1 CNS Tumours 2

1.1.2 Microarray Study on CNS Tumours 3

1.2 Motivation 3

1.3 Organisation of the Thesis 4

2 Computational Modeling and Gene Expression Studies: A Literature Review 6

2.1 Modeling 7

2.1.1 Reasoning 7

2.1.2 Global Modeling 8
2.1.3 Local Modeling 14
2.1.4 Personalised Modeling 16
2.2 Contemporary Gene Expression Data Analysis on Cancer Studies . . 18
2.2.1 Normalisation 19
2.2.2 Gene Selections 19
2.2.3 Cross Validation 22
2.3 Case Studies: Gene Expression Analysis on CNS Tumours 23

3 Ontology Systems: A Literature Review 25
3.1 Overview of Ontology Development Environment and Systems 25
3.2 Overview of the Semantic Web 29
3.2.1 OWL 31
3.3 Applications of Ontology-based System 33
3.3.1 Gene Ontology 33
3.3.2 The KEDRI's Brain-Gene Ontology (BGO) 34

4 A Methodology for Ontology-based Gene Expression Analysis and Personalised Modeling 36
4.1 Modeling Experiment 37
4.2 Gene Selection 38
4.2.1 One-Vs.-All based SNR Gene Selection Method 39
4.2.2 Case Studies on Gene Selection 40
4.2.3 Summary of OVA-SNR Gene Selection 44
4.3 Classifiers 44
4.3.1 Weighted-Weighted K Nearest Neighbours 44

4.4 Multi-classes Classifications in WWKNN 46
4.4.1 Multilayered Threshold Approach 46
4.4.2 OVA Approach 47
4.5 Knowledge Representation with Ontology 49
4.5.1 CNS Ontology 49
4.5.2 The Factors of CNS Ontology 50
4.5.3 The Connection between the CNS Ontology and BGO 52

5 Experimental Results on the CNS Case Study Problem 55

5.1 Datasets 55
5.2 Experimental Purposes 56
5.3 Experiment Setup 57
5.3.1 Principal Component Analysis (PCA) 58
5.3.2 Microarray Diagnosis Setup 58
5.3.3 Parameters Setup for Relevant Algorithms 58
5.3.4 Selected Genes 59
5.4 Experimental Results 61
5.4.1 Dataset A 61
5.4.2 Dataset A1 63
5.4.3 Dataset A2 65
5.4.4 Dataset B 66
5.4.5 Dataset C 67
5.5 Comparison of Two Approaches of WWKNN on Multi-class Classification 72
5.6 Summary of Modeling Experiment 75

6 Ontology-based Modeling and Knowledge Discovery Illustrated on the Case Study Problem 76

6.1 Knowledge Discovery Method 77

6.1.1 Knowledge Discovery with WWKNN Modeling 77

6.1.2 Knowledge Discovery with the CNS Ontology 78

6.2 Knowledge Discovery in Multi-class Problem 79

6.2.1 Analysis on Medulloblastomas 80

6.2.2 Analysis on Malignant Gliomas 83

6.2.3 Analysis on Atypical Teratoid/Rhabdoid Tumours (AT/RTs) . 86

6.2.4 Analysis on Primitive Neuroectodermal Tumours 89

6.3 Knowledge on Principal Histological Subclass of Medulloblastomas . . 93

6.3.1 Discriminant Gene Discovery on Classic Medulloblastomas . . 93

6.3.2 Discriminant Gene Discovery on Desmoplastic Medulloblastomas 95

6.4 Analysis on Clinical Outcome of Medulloblastomas 97

6.4.1 Discriminant Gene Discovery for Clinical Outcome Prediction of Medulloblastomas 97

6.4.2 Knowledge Discovery on Interaction between Genes and Drugs 99

6.5 Upgrading BGO with the newly Discovered Knowledge 101

6.6 Conclusion 102

7 Conclusion 104

7.1 Summary 104

7.2 Future Work 106

A MATLAB code of One-Vs.-All scheme and WWKNN 108

B Selected genes 118

Table of Abbreviations

BGO	Brain Gene Ontology system
CNS	Central nervous system embryonal tumours
ECF	Evolving Classifier Function
ECOS	Evolving Connectionist System
kNN	K-nearest neighbours
LOOCV	Leave-one-out cross validation
MLP	Multi-layer perceptron
OVA	One-Vs.-All scheme
OWL	Web Ontology Language
PCA	Principal component analysis
SNR	Signal-to-Noise ratio
SVM	Support vector machine
WKNN	Weighted K-nearest neighbours
WWKNN	Weighted distance, weighted variables K-nearest neighbours Or weighted-weighted K-nearest neighbours

List of Figures

2.1 Feedforward architecture of Multi-layer perceptron 9

2.2 An overview of the SVM process. 12

2.3 Defination of optimal hyperplane. 13

2.4 ECOS architecture (Kasabov, 2001). 15

2.5 Example of kNN classification (Scholarpedia, 2008). 17

2.6 General workflow of gene expression data analysis 19

3.1 Ontology example on animal category. 27

3.2 An ontology class example 28

3.3 An ontology class example 29

3.4 The Semantic Web layer infrastructure (Jacco, Lynda et al., 2002). . 30

3.5 The semantic part from Semantic Web infrastructure. 30

3.6 Brain-Gene ontology is constructed based on the complex relationships between genes, their influence upon neurons and the brain (Kasabov et al., 2007 & 2008). 34

4.1 General work flow of this research 37

4.2 The correlation between the genes and samples. Genes 1-10 are correlative genes to class 1. Genes 11-20 are correlative genes to class 2. Genes 21-30 are high correlative genes to class 3. Genes 31-40 are high correlated genes to class 4. Genes 41-50 are high correlated genes to class 5. 42

4.3 Comparison of OVA SNR and Normal SNR on dataset B. In the subfigure (a), genes 1-25 represent the correlative genes of class 1, and gene 26-50 represent the correlative genes of class 2. The signals of dataset B to each class are clearly presented in subfigure (a). 43

4.4 OVA WWKNN 48

4.5 Relationship between Pomeroy's data, BGO, GO and CNS ontology. The literature of Pomeroy provides a basic knowledge to structure the patterns in CNS knowledge domain. The structured knowledge of CNS can be exchange with the BGO. The GO provides the external knowledge to the CNS ontology 49

4.6 Snapshot of CNS ontology showing hierarchical structure in CNS knowledge domain. 51

4.7 The objects relations between the Samples class and subclass of the Common information of samples class. 51

4.8 The connection between CNS ontology and BGO. 53

5.1 PCA using 35 selected genes that are associated with each tumour type in dataset A. 62

5.2 PCA using selected 35 genes to describe the dataset A1. 64

5.3 PCA using selected 35 genes to describe the dataset A2. 65

5.4 PCA using selected 36 genes to describe the dataset B. The most samples of class 1 and class 2 can be linearly separated. 67

5.5 PCA using selected 30 genes to describe the dataset C. 68

5.6 PCA using selected 40 genes to describe the dataset C. 69

5.7 PCA using selected 50 genes to describe the dataset C. 70

5.8 PCA using selected 60 genes to describe the dataset C. 71

5.9 Two multi-class classification of WWKNN on dataset A, A1 and A2 . 74

6.1 Snapshot of CNS ontology detail showing query research system looking for medulloblastoma samples that closely correlate to gene D20124 (gene accession No.) as an example. 79

6.2 Visualising medulloblastomas with 3 discriminant genes in 2D. Sample no. 1-10 represents the samples who has medulloblastoma. M93119 indicates the most variant values in samples of class 1. 81

6.3 Boxplots of three discriminant genes in Medulloblastomas 82

6.4 Visualising medulloblastomas with 3 discriminant genes. 83

6.5 Visualising malignant gliomas with 3 discriminant genes in 2D 84

6.6 Boxplots of three discriminant genes in Malignant gliomas. Gene X86693 and U45955, the malignant glioma boxes have longer interquartile range (the distance between the top and bottom of the box) that implies a widely separated value distribution of gene expression on analysed discriminant genes. 85

6.7 Visualising malignant gliomas with 3 discriminant genes in 3D 86

6.8 Visualising atypical teratoid/rhabdoid tumours with 3 discriminant genes in 2D 87

6.9 Boxplots of three discriminant genes in Atypical teratoid/rhabdoid tumoursthe. AT/RTs appear higher median line and separable distribution of gene expression values. 88

6.10 Visualising atypical teratoid/rhabdoid tumours with 3 discriminant genes in 3D 89

6.11 Visualising Primitive neuroectodermal tumours with 3 discriminant genes in 2D 90

6.12 Boxplots of three discriminant genes in Primitive neuroectodermal tumours. Means of class PNETs and means of class malignant glimas with gene HG4178-HT4448. The samples of PNETs and samples of AT/RTs has a significant overlapped gene expression values with gene X14830. 91

6.13 Visualising Primitive neuroectodermal tumours with 3 discriminant genes in 3D 92

6.14 Three discriminant genes of classic Medulloblastomas 95

6.15 Three discriminant genes of desmoplastic medulloblastomas 97

6.16 Gene expression values of TrkC across the samples of dataset C. . . . 99

6.17 Visualising 3 discriminant genes in 3D 101

List of Tables

3.1 An example of a metadata record 26

4.1 Performance accuracy of SVM and kNN that is built on dataset A with the 50 OVA-SNR ranked genes. 41

4.2 Performance accuracy of SVM and kNN that is built on dataset B with the 50 OVA-SNR ranked genes. 43

4.3 List of individual in CNS ontology. 52

5.1 Summary of microarray datasets used for experiment 55

5.2 Relevant software used for experiment 57

5.3 The number of top-ranked genes to be selected in dataset A 60

5.4 The number of top-ranked genes to be selected in dataset B 60

5.5 The number of top-ranked genes to be selected in dataset C 61

5.6 The best classification result of every applied modeling method on dataset A. 63

5.7 The best classification result of every applied modeling method on dataset A1 64

5.8 The best classification result of every applied modeling method on dataset A2 66

5.9 The best classification result of every applied modeling method on dataset B 66

5.10 The best classification result of every applied modeling method on dataset C with 30 top-ranked genes 68

5.11 The best classification result of every applied modeling method on dataset C with 40 top-ranked genes 69

5.12 The best classification result of every applied modeling method on dataset C with 50 top-ranked genes 71

5.13 The best classification result of every applied modeling method on dataset C with 60 top-ranked genes 72

5.14 Results comparison of two approaches 73

6.1 An example for personalised data sample analysis using WWKNN on Pomeroy's dataset B with 6 genes. 78

6.2 Three defined discriminant genes of Medulloblastomas. 80

6.3 Three defined discriminant genes of malignant gliomas. 83

6.4 Three defined discriminant genes of AT/RTs. 86

6.5 Three defined discriminant genes of PNET. 90

6.6 Results of two samples t-test from MATLAB 92

6.7 Discriminant genes for subclass of medulloblastomas 94

6.8 The means of gene expression values on gene HG1980-HT2023, U63842 and X67951 across samples of dataset B 94

6.9 The standard deviation of gene expression values on gene HG1980-HT2023, U63842 and X67951 across samples of dataset B 94

6.10 Discriminant genes for desmoplastic medulloblastomas 96

6.11 Means of gene expression values on gene HG3543-HT3739, X53331 and X65724 across samples of dataset B 96

6.12 Standard deviation of gene expression values on gene HG3543-HT3739, X53331 and X65724 across samples of dataset B 96

6.13 Mean of gene expression values on gene M93119, X06617 and U05012_ s across samples of dataset C. 98

6.14 Standard deviation of gene expression values on gene M93119, X06617 and U05012_ s across samples of dataset C. 98

6.15 Discriminant genes for outcomes of medulloblastomas 100

6.16 WWKNN results on drugs 100

6.17 The official symbols for these discriminant genes 102

Chapter 1

Introduction

1.1 Research Background

The human brain controls the central nervous system (CNS), by way of the cranial nerves and spinal cord, the peripheral nervous system (PNS) and regulates virtually all human activity (Richard, 2000). The study of how brain functions can be extremely difficult. For example, the human brain contains roughly 100 billion neurons, each links to as many as 10,000 other neurons. With advanced software program and modeling algorithms, we could simulate some of brain function and gene regulatory network (Serruya et al., 2002). Such systems could contribute in many research areas including artificial intelligence (AI) and brain-related disease diagnosis. However, the rapid growing data and disparate data sources in the related area are bringing out a new challenge to acquire, represent maintain and share knowledge from large and widely distributed data resource.

In 2007, KEDRI research team published the first version of Brain Gene Ontology (BGO) system (Kasabov et al., 2007, 2008). It focuses on the integration of human brain information from different disciplinary domains such as neuroscience, bioinformatics, genetics, computer and information science. BGO is completed as an ontology-based knowledge framework system, which is usable by both computers and users. Based on the ontology knowledge framework, users could trace the rich information space of brain functions and related diseases, brain related genes and their activities in certain parts of the brain and their relation to brain diseases (Kasabov

et al., 2007, 2008). BGO is further discussed in Chapter 3. As part of brain gene ontology development, this thesis focuses on knowledge discovery between the genes and cancer diseases that occur in the central nervous system (CNS).

1.1.1 CNS Tumours

The CNS controls the five senses (smell, touch, taste, hearing and sight), movement, as well as other basic functions of our body, including heartbeat, circulation, and beathing (Anthea et al., 1993). The spinal cord consists of nerves that carry information to transform between the body and the brain. CNS tumours are the most feared cancers. Although cancers involving the CNS can cause pain, substantial disability and even death, they attack the body whereas CNS tumours cause seizures, dementia, paralysis and aphasia that attack the self (Lisa et al., 2002).

Based on statistics of American Society of Clinical Oncology (ASCO), until 2008 approximately 3,200 central nervous system tumours are diagnosed each year in children under the age of 20. About 800 of these are considered benign (non-cancerous) tumours (Oncology, 2008). In America, central nervous system tumours are the second most common childhood cancer, after leukemia.

For the diagnosis of CNS tumour (Oncology, 2008), doctor may suggests to use one or more following tests:

- **Computed tomography (CT or CAT) scan** creates a three-dimensional picture of the inside of the child's body with an x-ray machine.
- **Magnetic resonance imaging (MRI)** uses magnetic fields, not x-rays, to produce detailed images of the brain and spinal column.
- **Biopsy** is the removal of a small amount of tissue for examination under a microscope.

The traditional treatments focus on surgery, radiation therapy, and chemotherapy. Both of diagnosis and treatment may occur several side effects such as anaemia, fatigue, alopecia, mucosities and nervous system disturbances (Oncology, 2008).

All of cancer diseases arise with gene mutation that will result as DNA damages. If we could model the these genes and define the cancer related ones, we are able to find a way to cure these genes. It will not only reduce the side effects to patient from the treatment, but also increase the opportunity of patient to survive.

1.1.2 Microarray Study on CNS Tumours

The microarray technology is a newer method of monitoring expression levels for thousands of genes simultaneously. We could apply several computational algorithms to model gene expression data. Currently, the microarray data has been used in many research areas, such as genome annotation, gene expression analysis, regulation anaylsis, protein expression anaylsis, analysis of mutations in cancer, prediction of protein structure and modeling biological systems. It has been reported to be able to produce highly accurate result in clinical decision making in complex disease diagnosis such as (Petricion, Ardekani et al., 2002; Zhu, Wang et al., 2003). Microarray technology is thus considered as a revolution for knowledge discovery in human disease(Schena, 2002).

Pomeroy et al. (2002) firstly published the CNS related microarray study based on the data, that contains 99 sample with 7,129 genes. Their results have shown that the clinical outcome of children with medulloblastomas is highly predictable based on gene expression profiles of their tumours. This thesis also uses the Pomeroy's data to conduct the anaylsis.

1.2 Motivation

Over last half decade, Pomeroy's data has been so far extensively studied. And many models have been developed using this data. All these models can be classified into three major categories including global, local and personalised modeling. However, many proposed models do not have the ability to provide reliable information of the patients who require individual therapy schemes. In addition, most of researches focused on the performance of modeling algorithms. This leaves us a open question as to which genes are related to CNS tumours. These issues and question motivate us to deliver two major contributions that are presented as following:

1. It is reveal in empirical research that personalised modeling has been reported efficient for clinical and medical applications of learning system. Because its focus is not only on the model, but also on the individual sample. This thesis is offering a comparative study of major modeling and approaches of CNS cancer diagnosis. This study applies a newer personalised modeling method - WWKNN and several major algorithms from global, local and personalised modeling approaches.

2. In microarray based cancer diagnosis, one fundamental tasks is the performance of computational algorithm. The other task is to find the reusable knowledge from the experimental results. This thesis pays more attention on the reusable knowledge discovery. All the discovered knowledge will be imported into the large information space of BGO for the future researching and teaching.

1.3 Organisation of the Thesis

The organization of the thesis is as follows:

Chapter 2 reviews several widely-used computational algorithms, including multi-layer perceptron (MLP), support vector machine (SVM), evolving classifier function (ECF), k-nearest neighbours (kNN), weighted k-nearest neighbours (WKNN). We then discuss a typical procedure of microarray data analysis which includes normalisation, gene selection, cross validation and microarray classification. Three case studies on gene expression analysis of central nervous system tumours are reviewed at the end of this chapter.

Chapter 3 provides a literature review of ontologies. Five main components of ontology presentation and Web-based ontology language (OWL) are discussed. Two ontology systems are discussed as well.

Chapter 4 presents a detailed procedure of the proposed experiment. A newer personalised modeling algorithm, weighted-weighted K nearest neighbour is introduced. However, I address the limitation of WWKNN on the multi-class problem. The two approaches of multi-class classification are developed. One is the layered threshold. The other one is One-vs.-All (OVA) sheme WWKNN. The OVA scheme is also

applied on the gene selection method in this study. Finally prototype of the CNS ontology framework is described.

Chapter 5 summarise the experimental results obtained from six proposed classification methods (MLP, SVM, ECF, kNN, WKNN and WWKNN) on Pomeroy's data. The best results from each modeling approaches are compared as well. This chapter also discusses different classification performance from two different approaches of WWKNN.

Chapter 6 presents the discriminant genes discovery of using the CNS ontology system and WWKNN modeling. Several statistical techniques have been applied for analysis. All the discriminant genes are imported into both CNS ontology and Brain Gene Ontology system.

Finally, Chapter 7 contains the conclusion of this thesis as well as the areas identified for future research.

Chapter 2

Computational Modeling and Gene Expression Studies: A Literature Review

The knowledge discovery approaches to the area of data analysis and decision support system can be divided into two main modules: ontology, knowledge-based module and computational modeling module (Kasabov, 2006). Ontology module displays a higher ability to structure and represent the relationship between objects. Modeling module on the other hand uses the special skills to produce reliable and useful results. Both modules evolve through continuous learning from new data. Produced outputs of the modeling can be added to the ontology thus enriching its knowledge base and facili-tating new discovery. This chapter gives an overview of modeling in the field of data analysis. Firstly we discuss two reasoning approaches (e.g. induction and transduction) as the main theory of modeling methods with the description of global, local and personalised modeling approach. For each modeling approach, we explain several representative algorithms. Since this research involves the study of gene expression modeling, we will briefly reviews the microarracy gene expression data analysis. We then conclude on previous studies on brain tumour gene discovery.

2.1 Modeling

The word "modeling" comes from the Latin word modellus. It is well known that by 2.000 BC man had a decent knowledge of mathematics and used mathematical models to solve specific problems in their every-day life (Schichl, 2000). Currently, the model is described as a pattern or representation that theoretically describes some objects, with a set variables and number of relationships between these variables and objects (William, 2001). Modeling is a process to construct a logical and a formal framework that represents those relationships. The technique of modeling is widely used throughout many natural and social sciences including psychology and philosophy(Winsberg, 1999).

In the area of gene expression analysis, modeling aims to understand and extract interaction patterns from the thousands of gene data (Sobral, 1999). Gene interaction involves the dynamics of thousands of genes, proteins, and is influenced by many environmental and developmental factors (D'Haeseleer et al. 2000).

2.1.1 Reasoning

Reasoning is mental action or process of seeking for reasons for beliefs, summaries or feelings (Kirwin, 1995). In context of models, reasoning is determined by a set of logical principles, although rarely is the reasoning used completely mathematical. The main division between forms of reasoning for the modeling is concluded between transductive reasoning and inductive reasoning. In contemporary medical and bioinformatics modelling study, both inductive and transductive reasoning are widely used for different tasks.

Induction is reasoning from observed training cases to a global problem space, which also represents on the test case (Quinlan, 1986). It is used to describe properties or relations to types based on an observation instance (i.e., on a number of observations or experiences); or to formulate laws based on limited observations of recurring phenomenal patterns. An inductive model proceeds from a study about collected samples to a conclusion of about the global population. It is often represented as global modeling approach. However, the accumulation of clinical and statistical

data, many evidences pointed out that the patients are differently responding to same medical treatment (Barton, 2008). This means predictions of treatment outcome may be not achievable by the inductive modeling. Then researchers started to pay more attention on the idea of transductive reasoning in area of bioinformatics.

Transductive inference was firstly introduced by Vladimir Vapnik in the 1990's, motivated by his view that transduction is preferable to induction since, according to him, induction requires solving a more general problem before solving a more specific problem:"When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one (Vapnik, 1998)." This theory is very much connected to current clinical and medical arguments that the patient treatment needs to focus on individual conditions. Based on theory of transductive reasoning, this approach has been implemented into two categories of modeling: local and personalised modellings.

2.1.2 Global Modeling

The theory of global modelling is inherited from the inductive reasoning which is a single function that is created from the whole problem space in one task. The global models have widely used linear and logistic regression methods for gene expression modeling and for gene regulatory networks modellings. In this section we describe two representive algorithms for global modeling. They are known as the multi-layer perceptron (MLP) and support vector machine (SVM).

Multi-layer Perceptron

A name "artificial neural network" represents a mathematical model or computational model to discover complex relationships between inputs and outputs in the given dataset. The perceptron is a type of artificial neural network firstly invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt (Rosenblatt, 1958). In 1987, David Rumelhart, Geoffrey Hinton and Ronald Williams implemented a multilayer perceptron with nonlinear but differentiable transfer functions. The multi-layer perceptron has a feed-forward architecture as shown in Figure 2.1.

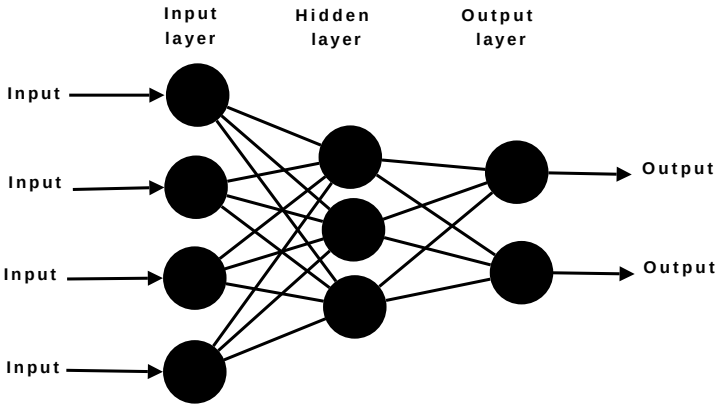


Figure 2.1: *Feedforward architecture of Multi-layer perceptron*

Such networks have an input layer (on the left in Figure 2.1) , one or more hidden layers and an output layer (on the right). All the layers of the network are fully connected to each other which means every processing neuron in one particular layer is connected to every neuron in the layer above and below. The connections carry weights which influence the behaviour of the network and can be adjusted during training (Kanellopoulos, 1997). This training operation consists of two stages: the "forward pass" and the "back-propagation". In the "forward pass" an input pattern vector is presented to the network and each neuron in the network computes an output according in the same of this inputs. For successive layers the input to each node is then the sum of the scalar products of the incoming vector components with their respective weights.

Input layer presents the variable values of input vector X. The input variable values are represented as $x_1, x_2, ..., x_I$.

Hidden layer receives the value from input layer. The value from each input neuron is multiplied by a weight (v_{ji}), and the resulting weighted values are added together producing a combined value net_j as shown in Function 2.1. The weighted sum (net_j) is fed into a transfer function, which outputs a value y_j as shown in Function 2.2. The outputs from the hidden layer are distributed to the output layer. .

$$net_j = \sum_{i=1}^{I} v_{ji} x_i \tag{2.1}$$

where v_{ji} is the weight connecting input node i to hidden node j and x_i is the output from input node i.

$$y_j = f(net_j) \tag{2.2}$$

Output layer receives the value from hidden layer. The value from each hidden neuron (y_j) is multiplied by a weight (w_{kj}), and the resulting weighted values are added together producing a combined value net_k as shown in Function 2.3. The weighted sum (net_k) is fed into a transfer function, which outputs a value o_k as Function 2.4. The o_k values are the outputs of the network.

$$net_k = \sum_{j=1}^{J} w_{kj} y_j \tag{2.3}$$

where w_{kj} is the weight connecting hidden node j to output node k and y_j is the output from hidden node i.

$$o_k = f(net_k) \tag{2.4}$$

The multi-layer perceptron is trained by supervised learning method with a back-propagation algorithm. Each input pattern of the network is required to adjust the weights attached to the connections so that the difference between the network's output and the desired output for that input pattern is decreased. Based on this difference error terms or δ terms for each node in the output layer is computed. The terms of δ are presented as Equation 2.5 and 2.6(Rumelhart et al., 1988).

$$\delta_{ok} = (d_k - o_k) f'_{net_k} \tag{2.5}$$

where d_k is the desired output for a node k.

$$\delta_{yj} = (\sum_{k=1}^{K} \delta_{ok} w_{kj}) f'_{net_k} \tag{2.6}$$

The weights between the output layer and the hidden layer are then adjusted:

$$w_{kj} = w_{kj} + \alpha \delta_{ok} y_j \tag{2.7}$$

where $\delta_{ok}$is the rate of change of error with respect to the input to node k, and is given by Equation 2.5. α is a learning rate parameter.

The weights adjustment is then computed as:

$$v_{ji} = v_{ji} + \alpha \delta_{yj} x_i \tag{2.8}$$

Multilayer perceptrons are very useful to get approximate solutions for extremely complex problems such as speech recognition, image recognition, and machine translation software. In general, the most popular use of MLP has been in the growing field of artificial intelligence, where the multilayer perceptron is often used to simulate biological neural networks in the human brain.

Support Vector Machine

Vapnik (1998) firstly introduced the support vector machine (SVM) that performs classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories (Vapink, 1998). A SVM model consists of a set of vectors described by a kernel function that compute a hyperplane to maximize the margin between the training samples and class boundary (Huerta, Duval & Hao, 2006). In the parlance of SVM literature, the input variables are called attribute, and a transformed attribute that is used to define the hyperplane is called a feature. A set of features describes one vector. So the goal of SVM modeling is to find the optimal hyperplane that separates clusters of vectors in such a way that vectors of the same class are on one side of the plane and vectors of other class are on the other size of the plane. The vectors near the hyperplane are the support vectors. The Figure 2.2 below presents an overview of the SVM process.

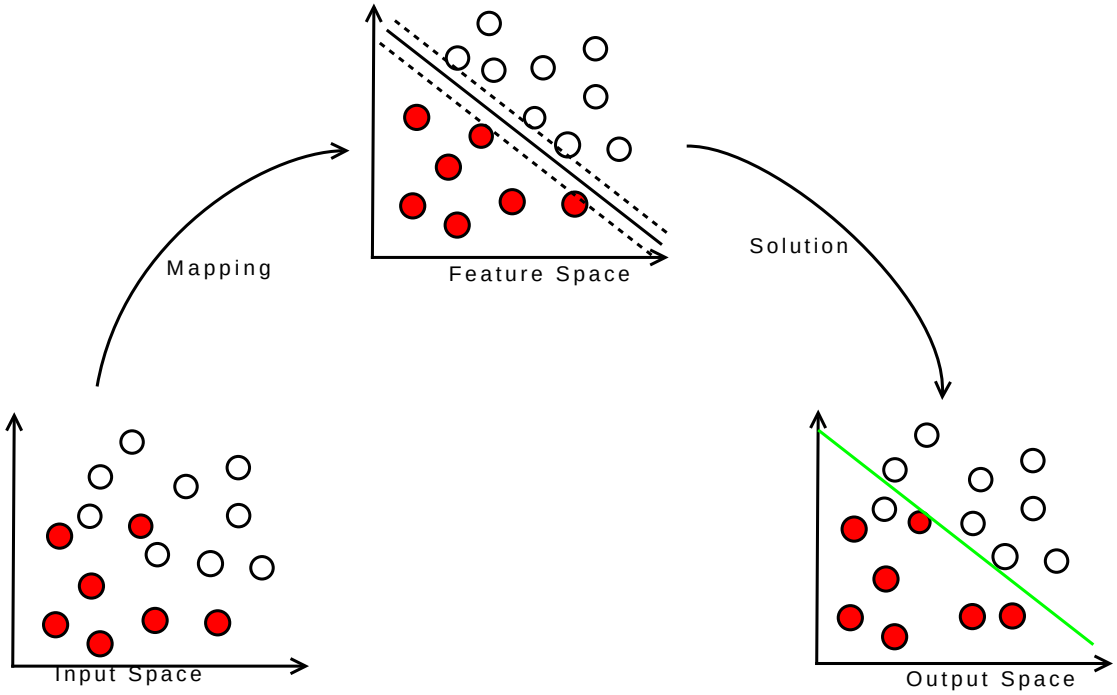


Figure 2.2: *An overview of the SVM process.*

Assume we have a 2-dimensional classification problem that vectors of the dataset has a categorical target variable with two classes. The Figure 2.3 describes this example. One class of the target vectors is represented by rectangles while the other class is represented by ovals. In this Figure, the vectors with one class are located in the left hand side and the vectors with the other class are in the right hand side; the vector are completely separated. The SVM analysis attempts to find a 1-dimensional hyperplane (i.e. a line) that separates the cases based on their desired class labels. The possible hyperplane could be infinite number by using Equation 2.9. In the Figure 2.3, we identified two candidate lines in both left hand side figure and right hand side figure. The question now is which line is better, and how do we define the optimal line.

$$w \times x + b = 0 \tag{2.9}$$

The Figure 2.3 describes the answer. The dashed lines drawn parallel to the separating line mark the distance between the dividing line and the closest vectors to the line. The left lower parallel is presented as Equation 2.10, and the right higher line

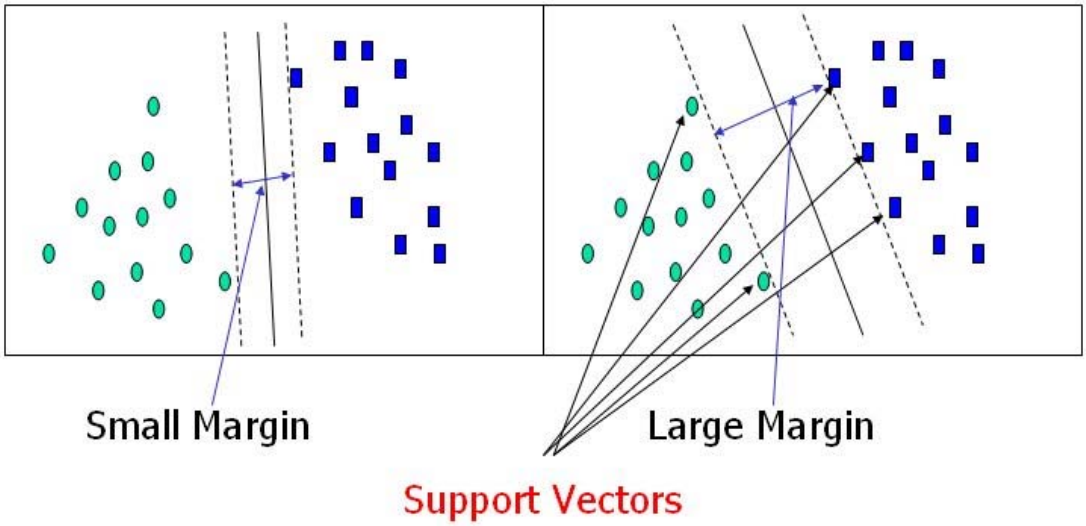


Figure 2.3: *Defination of optimal hyperplane.*

is represented as equation 2.11. The distance between the dashed lines is called the margin. The vectors that constrain the width of the margin are the support vectors. The optimal hyperplane is oriented so that the margin between the support vectors is maximized, which is shown in the right hand of Figure 2.3.

$$w \times x + b = -1 \tag{2.10}$$

$$w \times x + b = 1 \tag{2.11}$$

The margin is calculated as Equation 2.12.

$$Margin = \frac{2}{\|w\|^2} \tag{2.12}$$

If we are given a training dataset of form: $\{(x_1,\ y_1),\ (x_2,\ y_2)...(x_n,\ y_n)\}$. where the y represents the desired class labels as known either 1 or 2, indicating the class to which the point belongs. The completed classification function is described as following Function 2.13

$$f(x) = \begin{cases} 1 & \text{if } w \times x + b = -1 \\ 2 & \text{if } w \times x + b = 1 \end{cases} \tag{2.13}$$

SVM have been widely used in many research paper for classification and regression (Shipp et al., 2002). And it has been described as very accurate classification model. However, the knowledge extraction from the SVM is very limited (kasabov, 2006).

2.1.3 Local Modeling

Local modelling algorithms are created on data but representing on only a sub-space of the problem space. They are often built on clustering techniques that a cluster means subset of similar data. Such techniques include K-means (Hartigan and Wong, 1976), Self-Organising Maps (Ultsch, 2007) and fuzzy clustering (Kasabov, 2007). In this section, we only present one algorithm that is called "evolving classifier function"(ECF) (Kasabov, 2002) which is an implementation of the evolving connectionist system architecture.

Evolving Classifier Function

The many problem of neural network models are seen as "black box" which are not useful discover new patterns from data. Kasabov (1998) introduced a new type of neural network, called evolving connectionist system (ECOS). It allows for structural adaptation, fast incremental, on-line learning, and rule extraction and rule adaptation (Kasabov, 2001, 2002). Figure 2.4 illustrates an evolving connectionist systems that consists of five layers of neurons and four layers of connections. The first layer (bottom layer) is input layer that receives the information. The second layer calculates the fuzzy membership degrees (e.g. Small, Medium, or Large) to which the input values belong to predefined fuzzy membership function. The third layer represents the connections between the input and output variables. The fourth layer calculates the fuzzy membership degrees to which the output belongs according to predefined fuzzy membership functions. The last layer is the output layer which performs defuzzification and calculates output values. The evolving classifier function (ECF) is an algorithm that is built on the ECOS architecture (Kasabov, 2001;

Kasabov and Song,2003).

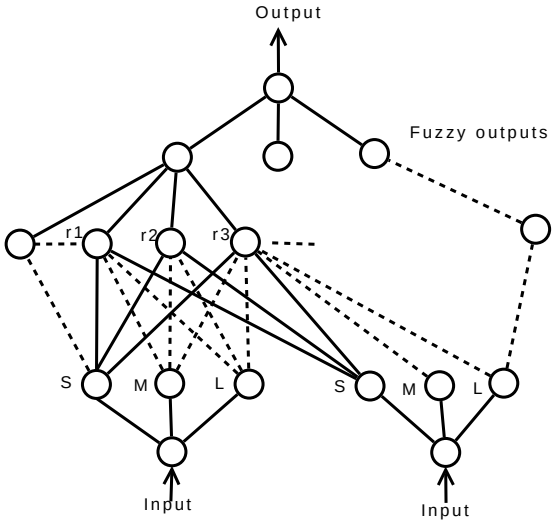


Figure 2.4: *ECOS architecture (Kasabov, 2001).*

There are no fuzzy output nodes as each evolving node r1, r2,... represents a cluster centre of input vectors that belong to the same output class using a defined maximum cluster radius R_{max} with the use of Euclidean distance (Kasabov 2007). ECF is a typical supervised learning which involves training and testing. The learning process is described as following steps.

1. Input the intended training vector from the dataset to the ECF model and calculate the distances between this vector and inputted rule nodes by using Euclidean distance.

2. Create a new rule node, if calculated distance is greater than R_{max}.

3. If there is a rule node with a distance to the input vector less then or equal to its radius and its class is different from this inputted vector, its influence field should be reduced. Otherwise, nothing change and go to step 1 inputing new vector.

4. If one rule node with a distance to the input vector less then or equal to the R_{max}, and its class is same as the input vector, increase the influence field by taking the distance as a new radius. Otherwise, repeat the step 2, and go to step 1.

ECF has been applied on many research area such as cancer diagnosis (Kasabov, 2006), robotic study (Huang et al., 2008) and image recognition (Li and Chua, 2003).

2.1.4 Personalised Modeling

A model that is created only for a single point (sample) of the problem space is named personalised model (Kasabov, 2007, 2008). Usually, the class label of new sample is defined by applying Euclidean distance and a voting scheme to the closed sample in a same dataset (Mitchell et al., 1997). In mathematics, Euclidean distance is the most common use of distance between two points that one would measure with a ruler. It examines the root of square differences between coordinates of a pair of objects (Black, 2004). In an Euclidean space, the Euclidean distance between points $X = (x_1, x_2, ..., x_i)$ and $Y = (y_1, y_2, ..., y_i)$ is defined as:

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.14}$$

K-Nearest Neighbour

K-nearest neighbour is a typical method used in personalised modelling. It is a very simple algorithm, which is based on minimum distance from the query vector to the training samples to determine the neighbourhood of samples. The distance is measured using a distance measurement approach, such as Euclidean distance as shown in 2.14. After we gather K nearest neighbors, we take sample majority of these K-nearest neighbors to be the prediction of the query vector.

Suppose we have been given a query vector $(X)=(x_1, x_2)$. Our purpose is to classify this vector based on the existed training dataset $(Y^1, Y^2, ..., Y^n)$. Notice that the samples from training dataset have same dimensional as the query vector X. Based on kNN learning process, we firstly determine the K as the number of nearest neighbour. Suppose we determine $K = 4$. Then we calculate the distance between the query vector and all the training samples individually. The distance is calculated by using Euclidean distance. The next step is to find the K. In Figure 2.5, our example is visually described, where the arrows point out the nearest neighbour to

query vector in a two dimensional space. In this neighbourhood, three training samples belong to class A and one sample is class B. Therefore the predicted class label of query vector is class A based on the sample majority of the class of four nearest neighbours.

In many literatures, the K-nearest neighbors is suggested to use odd number. But we still can use even number of K-nearest neighbours as our example. In this case, if the number of different classes are equal in the neighbourhood, we can choose arbitrary for one of the class labels.

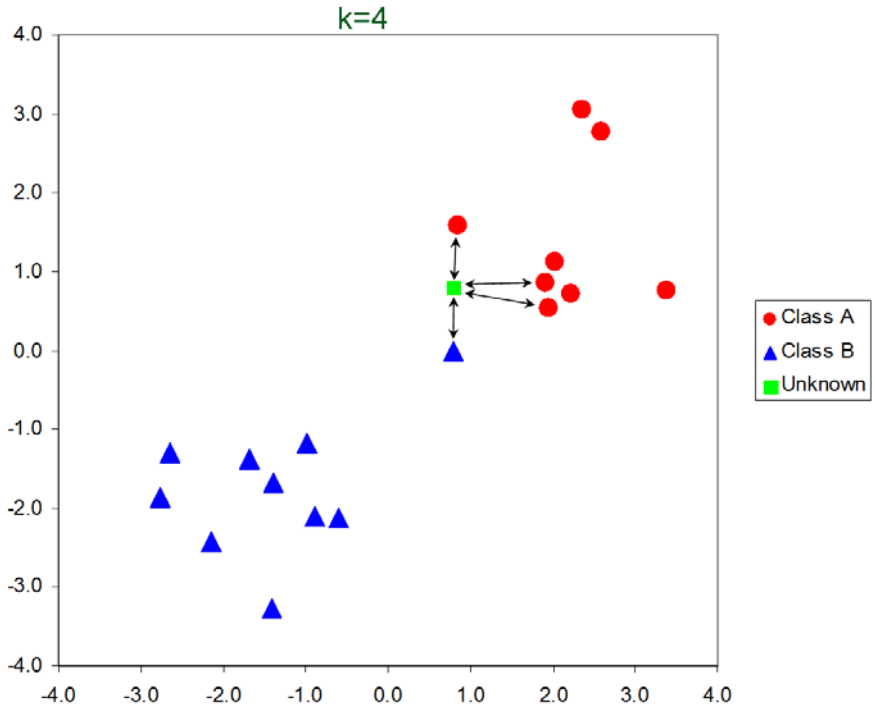


Figure 2.5: *Example of kNN classification (Scholarpedia, 2008).*

Weighted K-Nearest Neighbours

Weighted K nearest neighbours (WKNN) is a classification algorithm, that may also applys Euclidean distance to define the class label of query vector (Kozak and Kozak, 2004). The difference from kNN is that WKNN predict the class of query vector based on a weighted majority vote of the nearest neighbours. The weight is

calculated based on the distance to the query vector as shown in Equation 2.15.

$$w_j = [max\,(d) - (d_j - min\,(d))]\,/max\,(d) \tag{2.15}$$

where the distance d is calculated by using Euclidean distance as Equation 2.14.

The learning process of weight-kNN is quite similar to kNN. For a given query vector, we firstly decide for a number K of nearest neighbours. Then the distance of training vectors to query vector is calculated by using Euclidean distance. Additionally the weight for each distance is calculated as Equation 2.15. The next step is to record the K nearest neighbors based on the weighted distances. Notice that the predicted class label of a query vector is based on a "personalised probability" (y_i). It is calculated as Equation 2.16.

$$y_i = \sum_{j=1}^{k} w_j y_j / \sum_{j=1}^{k} w_j \tag{2.16}$$

where y_j is the class label for the recorded nearest neighbours. K is representing the number of the nearest neighbour.

In order to finally classify the query vector in WKNN, we have to select a probability threshold (p_{thr}). Such that y_i is classified in class 2, if the output of this vector is greater than p_{thr} in a two classes problem. The threshold is normally setup at beginning of the learning process with the range between 0 and 1.

2.2 Contemporary Gene Expression Data Analysis on Cancer Studies

Above section presents three different modeling approaches. In a gene expression study, completed DNA microarray experiment generally required two main parts of process. The first part is the collection of microscopic DNA spots, which involved sample, purification, reverse transcriptase, coupling, hybridisation and washes, scanning (Chen, 2007). The aim of this part is to transform a state of the cell or a tissue to a numerical raw data. The next part is to analyse the obtained raw data by using several mathematical algorithms as described in previous sections of this chapter. Before we apply algorithms, we need to prepare the dataset and set up an

experiment. Four steps cannot be omitted in data analysis of a DNA microarray experiment, which are shown in Figure 2.6. Following subsections describes these four steps in detail.

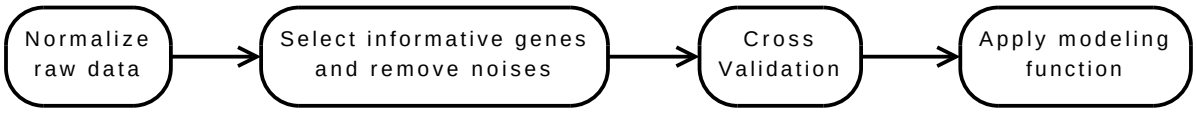


Figure 2.6: *General workflow of gene expression data analysis*

2.2.1 Normalisation

Normalisation is the first transformation applied to expression data. There are several reasons to normalise the microarracy data, including unequal quantities of starting RNA, difference between real biological variations of gene expression and variations of the measurement process, and systematic biases in the measured expression levels (Quackenbush, 2002). The most popular normalisation method is linear normalisation that transfers the value of vectors to between 0 and 1. This function is denoted as follows:

$$normX_i = X_i - \min(X) \,/\, \max(X) - \min(X), X = (X_1, X_2, ..., X_i) \tag{2.17}$$

where X is denoted as vectors in obtained raw dataset.

2.2.2 Gene Selections

With contemporary technologies, such as microarrays or microfluidics, we are allowed to measure the level of expression of up to 30,000 genes in RNA sequence that is generated by transcription from DNA, the information is already present in the cell's DNA (Gollub et al., 2003). A huge number of genes usually not only included many noise genes causing a high generalisation error, but also increase the cost of cancer diagnosis in terms of time and computation. The aim of gene selection is to explore a small group of informative genes that can successfully classify any samples from a randomly collected dataset into represented classes (Tang, Suganthan, et al., 2006). Gene selection is operated as feature selection in terms of data mining, which it uses

to data pre-processing. However, gene selection also distinguish feature selection in field of machine learning, is to select features from a normally thousands genes set and a small number of samples. Two popular methods of gene selection are identified in the literature: T-test and signal-to-noise.

T-test is firstly published by Gosset in 1908 as a classical statistical theory (Gorsset, 1908). It is proposed to measure the different means between classes in a same problem space. Theoretically, the T-test can perform well even if the sample size is very small (Triola, 1998). Due to this reason, T-test has widely applied for gene selection in microarray studies (Arfin, Long et al., 2000; Ding et al., 2003; Thomas, Olson et al., 2001).

In the gene selection, the main idea of T-test algorithm is to judge the extent of each gene in between of every sample. The extent is computed as:

$$T_i = \left|\mu_i^{class1} - \mu_i^{class2}\right| / \sqrt{\left(\frac{1}{n_i^{class1} + n_i^{class2}}\right) \times \sigma_i} \tag{2.18}$$

where T_i denotes the output value of T-test for i^{th} gene or variable in the problem space. The μ_i^{class1} and μ_i^{class2} is the mean of i^{th} gene or variable corresponding to both class 1 and 2. n_{class1} and n_{class2} give the total amount of sample numbers in each class. σ_i represents the standard deviation for the i_{th} gene, which is calculated as:

$$\sigma_i = \sqrt{\left((n_{class1} - 1) \times \sigma_{class1}^2 + (n_{class2} - 1) \times \sigma_{class2}^2\right) / df} \tag{2.19}$$

where df is the degrees of freedom that is proposed to present the number of independent pieces of information available to estimate another piece of information. In this equation, we calculated our df by:

$$df = (n_{class1} - n_{class2} - 2) \tag{2.20}$$

In Equation of σ_i calculation, σ_{class1}^2 and σ_{class2}^2 are variance of two classes respectively. They are described as following equations:

$$\sigma^2_{class1} = \sum_{i=1}^{n_{class1}} \left(x_i - \mu_{class1}\right)^2 / \left(n_{class1} - 1\right) \tag{2.21}$$

$$\sigma^2_{class2} = \sum_{i=1}^{n_{class2}} \left(x_i - \mu_{class2}\right)^2 / \left(n_{class1} - 1\right) \tag{2.22}$$

One of major benefit of T-test is the simplicity and robustness that leads to a faster operating process for gene selection, but we must bear in mind that T-test is accurate under assumptions of data normality and variance equality in classes (Doug, 2002). The assumptions of T-test also become to the biggest weakness, since it could occur the high false in gene expression analysis. When multiple tests are applied in the microarray data analysis, the differentially expressed genes cannot be discovered from a dataset that have equal variances and independent genes. Empirical studies have indicated that the selected genes by simple T-test are not reliable in patterns discovery. This issue has led scientists to develop more specific gene selection approach for microarray data study such as SNR.

Signal-to-noise-ratio (SNR) is another popular algorithm that is implemented for gene selection. SNR is often adopted for evaluating the expression level of each gene to conduct the search for an informative gene set. SNR normally starts with the evaluation of a single informative gene iteratively defines the importance of every gene in entire dataset (Veer, Dai et al., 2002). The gene which has higher value of SNR will be chosen as a representative gene of the classes. The algorithm of SNR is represented as follows:

$$S_i = \left|\mu_i^{class1} - \mu_i^{class2}\right| / \left(\delta_i^{class1} - \delta_i^{class2}\right) \tag{2.23}$$

where μ and δ denote the mean (or median) and standard deviation for each class, and i=1,2,...,n that n is the number of genes in the dataset.

Since the theory of SNR is to rank the importance of each gene, SNR is a perfect algorithm to study the gene correlation coefficient to other genes such as the sutdy of (Goh, Song et al., 2004). In their study, SNR is used to selected the set of high-ranked genes that correspond to target classes. All genes were computed by Equation 2.23. Their results showed that SNR can remove many noise genes and improve the

accuracy of classification.

SNR method is also conducted as variables weighted calculation approach to evaluate the response of drugs in real clinical studies (Iwao-Koizumi et al., 2005). Iwao-Koizumi et al. introduced a weighted-voting (WV) algorithm that is denoted as follows:

$$v_i = w_i \times \left| x_i - \frac{\mu_i^{class1} - \mu_i^{class2}}{2} \right| \tag{2.24}$$

where x_i is the repression level of the i^{th} gene of query sample. μ_i^{class1} and μ_i^{class2} represent the means of gene expression in two classes samples. w_i is the weight of the i^{th} gene that is calculated by SNR as Equation 2.23.

2.2.3 Cross Validation

In the analysis of microarray data, the experiments are often leaded to some bias and unidentifiable errors. Therefore a sampling method must be employed in every microarray study. It is used to decrease the biases in process of classification by splitting the training and testing datasets. This section reviews a common sampling technique, which is called cross-validation.

Cross-validation is a sampling technique that widely used in the field of microarray data analysis. The main idea is to split parts of the original experimental dataset into the training set and testing set; analysed the training set by using a learning algorithm on, then apply the predicted model on the test set (Pang, Havukkala, et al., 2006). The process will be repeated until the every sample has been assessed. The main advantage of cross-validation is that every sample from the same dataset can be used for both training and testing, and the testing set is totally independent to the training set.

In general, there are many ways to perform Cross-validation. K-fold cross-validation and leave-one-out cross-validation are more popular. K-fold represents that randomly separate sample into K mutually exclusive subsets of approximately equal size. In every validation process, one subset is used for testing, and the others are proposed to train the model. The process will be rounded K times until every fold has been tested. This approach has been suggested in analysis for larger sample size, such as over 100 samples (Breiman and Spector, 1992).

Leave-one-out cross validation (LOOCV) is a special kind of K-fold cross-validation in which the number of K equals to the number of samples. In LOOCV, all the samples will be separated to test the training. This approach has been widely applied on small number of samples (Kohavi, 1995).

2.3 Case Studies: Gene Expression Analysis on CNS Tumours

In 2003, Pomeroy et al. published a research study that described a gene expression based prediction of central nervous system embryonal tumour (CNS) outcome (Pomeroy et al., 2003). Their study was based on a gene expression dataset that is collected from 99 different patients. The researchers firstly identified three research hypotheses:

1. Different embryonal CNS tumours can be distinguished from each other based on the gene expression values.
2. Desmoplatic and classic medulloblastoma are separable by gene expression.
3. Clinical outcome of medulloblastomas is predictable on the basis of the gene expression profiles.

The experiment of their research was organised as a supervised learning process which we have described in the section 2.2 (see Figure 2.6). In their study, they introduced a newer gene selection method that select the 'marker' genes with the highest correlation with target class by using SNR (see Section 2.5). A personalised modeling method, kNN was applied as a classification modeling. The high accuracy strongly support their hypotheses. Their study also used modeling algorithm to determine some debates of cancer related genes which have been mentioned in other medical literature.

The results support their research hypotheses. But we noticed that their result is quite unbalanced between two classes, when kNN was applied to distinguish the clinical outcomes of medulloblastomas. This raised the question of which modeling

algorithm should be applied to maximise the likelihood of getting a balanced predictions. Xiong Zhang and Chen (2007) have addressed this problem in their study. They have compared the SVM and kNN classifiers on the Pomeroy's dataset C (60 samples). Their experiment was carried out with the selected gene number respectively set to 10, 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, and 2000. During their experiment, they employed the SNR for their gene selection. The highest accuracy was produced by kNN with 100 selected genes.

Niijima and Kuhara (2005) focused on the dataset A (multi CNS tumours classification with 42 samples), a multi-class problem with 42 samples. They compared three different classification algorithms on the Pomeroy's dataset. Their applied algorithms were SVM, kNN and kernel subspace. In the experiment, 100 genes were selected for classification. The result suggested that the lowest errors rate was predicted by SVM, a global modeling algorithm. The kNN performed the lowest accuracy, 78.57%. This result was also lower than Pomeroy's result.

Both of the studies compares the accuracies of SVM and kNN for Pomeroy's datasets classification. For the different problem, the performance of two algorithms are quite different. However, the Pomeroy's dataset have been studied by many other researchers (Ayers et al., 2004; Howard et al., 2004 and Rhodes et al., 2005). The most research goals of those studies are only to compare the classification accuracy, a few of researches studies on pattern discovery. Due to this reason, Our focus of this thesis is not only to offer a comparative study of global, local and personalised modeling on CNS gene expression classification, but also to discover the interactive patterns based on the highest accurate model.

Chapter 3

Ontology Systems: A Literature Review

In the last chapter, we pointed out two modules (i.e modeling modules and ontology modules) in knowledge discovery research. The aim of the modeling modules is to find the theoretical construct for queried problems, and use numbers or mathematical equations to describe the problems and solutions. Then the aim of ontology is to translate those numbers or equations to both human and machine readable presentation within an idealised structure. This chapter presents a review of literature in related field of ontology. The Section 3.1 summaries the theory and important components of ontology. The semantic web, the ontology based Web presentation is discussed in Section 3.2. It is followed by two applications of ontology in Section 3.3.

3.1 Overview of Ontology Development Environment and Systems

The word "Ontology" has a long history in philosophy, in which it refers to the subject of existence since year 1613 (Øhrstrøm et al., 2005). It is also often confused with epistemology, which is about knowledge and knowing (Gruber, 1993). In 1987, ontology was firstly introduced to the context of computer and information sciences where an ontology is a form of knowledge representation about the world or some

part of it (Gruber, 2008). An ontology formalises the semantics of the objects and relations in a universe of discourse and provides a set of terms which can be used to talk about these objects and relations. Contemporary ontologies are used in several areas such as artificial intelligence, the semantic web, software engineering, bioinformatics, and information architecture. It is used to share conceptualisations of a domain, and they possibly include the representations of these conceptualisations. Common components of ontologies include metadata, classes, individuals, attributes and relationships (Chandrasekaran et al., 1999).

Metadata

Metadata or metainformation represents the structured data which describes the characteristics of a resource in the context of an information science (Taylor, 2003). Sometime metadata is also described as the "information of data". An alternative use of the term holds that "metadata" is information provided for direct processing by computer, in opposition to "data" which needs to be interpreted by human knowledge (Warwick, 1997).

A metadata record consists of a number of pre-defined elements representing specific attributes of a resource, and each element can have one or more values. Table 3.1 shows an example of a simple metadata record:

Element name	**Value**
Title	CNS row data
Creator	Scott L. Pomeroy et al.
Publisher	The Broad Institute
Identifier	http://www.broad.mit.edu/mpr/CNS/
Format	Text

Table 3.1: *An example of a metadata record*

Metadata and ontologies are very closely related. In the ontology based knowledge representation, metadata records may be used for a variety of purposes: to identify a resource to meet a particular information need; to evaluate the quality or fitness for use of such a resource; to track the characteristics of a resource for subsequent maintenance or usage over time; and so on.

Individuals

An ontology is a hierarchical structure having different layers. Individuals or instances are the basic, "ground layer" components of ontology. An individual may include a concrete object such as a person, an animal, a table, and a planet. An ontology does not need to include any individuals, but one of the general purposes of an ontology is to describe an actual relationship between individuals and means of classifying individuals, even if those individuals are not explicitly part of the ontology.

Classes

In the context of ontology, classes are the abstract groups, sets, or collections of objects. A class may contains individuals, other classes, or a combination of both. Each class has a unique class name or a identification that must be different to the others in a same ontology. The class that contains other classes is called superclass. A class subsumed by another is called a subclass. Figure 3.1 presents a simple ontology for describing knowledge of animals. In this knowledge domain, the named class "Animal" is the superclass that represents all animals. The classes of "Cat" and "Dog" are the subclasses which only represent one kind of animal. The class "White Cat" and "Black Cat" also two subclasses to subsume the class of "Cat".

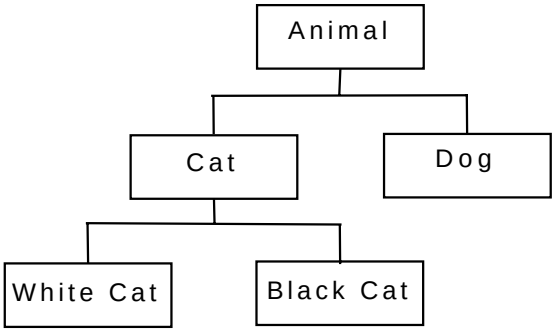


Figure 3.1: *Ontology example on animal category.*

A class is also a cohesive package that consists of a particular kind of metadata to the objects or individuals. It describes the rules by which objects behave. These rules are presented as attributes in ontology.

Attributes

An attribute is the information of a class within this ontology. Each attribute has at least a name and a value, and is used to store information that is specific to the class it is attached to. Suppose we have a class which is called "Car" as shown in Figure 3.2. The attributes of this class could be makes, model and color as well as specify the features of a car. Unlike to class name, an attribute of class does not need to be a unique value.

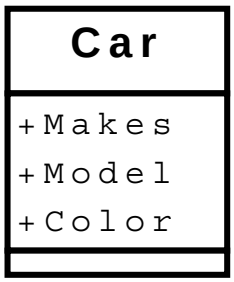


Figure 3.2: *An ontology class example*

Two types of attribute, objects attribute and datatype attribute would be defined in ontology based knowledge representation. The objects attribute is an important use of attributes that is to describe the ontology relations between objects. The datatype attribute is used to describe the non-related value of individuals. It does not need to connect with any other objects in the ontology.

Relations

A relation is still an attribute whose value is another object in the ontology. This is the only way to connect a class to others. Many different ways can be used to describe the relations such as "has-something", "part-of" or "is-something " as well as clearly depicts the relations. However the most important type of relation is the subsumption relation (is-superclass-of, the converse of is-a, is-subtype-of or is-subclass-of). This defines which objects are members of classes of objects. For example we have already seen that the "Car" class has three attributes: makes, model and color as shown in Figure 3.2. If we connect those objects attribute to

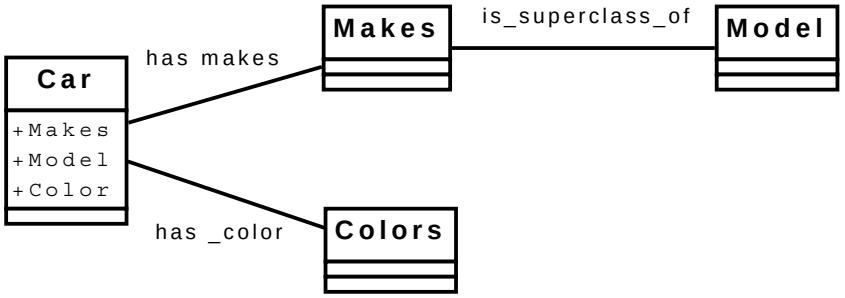


Figure 3.3: *An ontology class example*

related classes by using relations, the ontology would be shown as Figure 3.3. In this figure, "Car" class has two "has" relations which related to class "Makes" and "Color". The "is-superclass-of" relation is used to depict that class "Makes" is the parent class of class "Model". In the ontology, a subclass could connect other objects and classes throughout its superclass, which explain how the "Model" attribute is conducted to "Car" class in our example.

At its core, much of the power of ontologies comes from the ability to describe these relations. Together, the set of relations describes the semantics of the domain.

3.2 Overview of the Semantic Web

The Semantic Web is an evolving extension of the World Wide Web in which information is given explicit meaning, making it easier for machines to automatically process and use the web content. At its core, the Semantic Web is about two things. It is about a common framework that allows data to be shared and reused across application, enterprise, a variety of enabling technologies. It is also about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing (W3C, 2008).

Two important technologies for developing the Semantic Web are already in place: eXtensible Markup Language (XML) and the Resource Description Framework (RDF) (Berners-Lee, Hendler et al., 2001). XML is to define customised tagging schemes, which allows users to add arbitrary structure to their documents, but imposes no se-

mantic constraints on the meaning of these documents. RDF provides a simple data model for objects and relations between them, and associated serialisation in a XML syntax. Figure 3.4 provides an overview of the Semantic Web layer infrastructure on the Web. The first layer of this infrastructure (from the bottom of the figure),

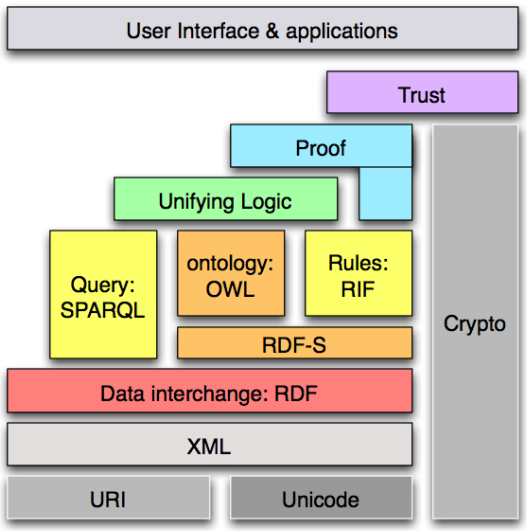


Figure 3.4: *The Semantic Web layer infrastructure (Jacco, Lynda et al., 2002).*

Universal Resource Identifier (URI) and Unicode are used to define the "address" of the web page or the database. The next layer is XML that give a tag to every document. If we only consider the semantic parts of the Semantic Web, four layers would be extracted from this infrastructure as shown in Figure 3.5.

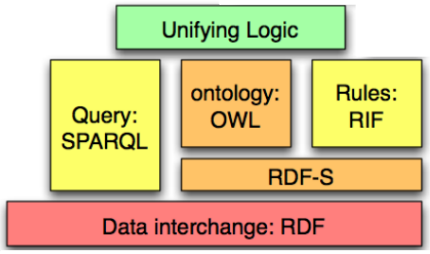


Figure 3.5: *The semantic part from Semantic Web infrastructure.*

- "**Data interchange: RDF**" is believed to be the most popular metadata in

the Semantic Web. The metadata at this layer contains just the concepts of resource and properties (Lassila and Swick, 1999) in here. This layer is also known as the metadata layer in (Berners-lee, 1998).

- **RDF-S or RDF schema** is considered as a candidate schema layer language, since this layer is to define a hierarchical description of concepts (is-a hierarchy) and properties in the Semantic Web.
- **Ontology layer** is combination of the layers above RDF-S in Figure 3.5. This layer introduces the web ontology language that provides a richer set of modeling primitives that can be mapped to the well-known expressive Description Logics (Lassila and Swick, 1999). Currently, the most powerful web ontology language is OWL. Unifying logic is a pattern in this layer to convert the human knowledge or logic to a mahine readable terminology. Both Query and Rule in this layer are used to transform the data to the next layer.

The next two layers in the infrastructure of the Semantic Web (see Figure 3.4) is trust and proof. Recently only a few study are available about these layers. The purpose of these layers is to determine whether or not a statement is true for the users. These two layers are very important to the users. Recent applications on the Semantic Web at the moment generally depend upon context. How to judge the reliability of the Web context is still a issue.

However this research is only focus on the semantic part of the Semantic Web. The most important layer in this part is the ontology language that can formally describe the meaning of terminology used in Web documents for the Semantic Web. If machines are expected to perform useful reasoning tasks on these documents, the language must go beyond the basic semantics of RDF Schema (W3C, 2008). The next subsection describes the Ontology Web Language (OWL).

3.2.1 OWL

OWL is a web-based ontology language that is designed for use by applications that need to process the content of information instead of just presenting information to humans. It is a revision of the DAML+OIL web ontology language. In infrastructure of the Semantic Web, OWL is built on top of RDF, and is written in XML.

Comparing to RDF, OWL and RDF are much of the same thing, but OWL is a stronger language with greater machine interpretability than RDF. OWL adds more vocabulary for describing properties and classes: among others, relations between classes, cardinality, equality, richer typing of properties, characteristics of properties, and enumerated classes (W3C, 2008). The data described by an OWL ontology is interpreted as a set of "individuals" and a set of "object relations" which relate these individuals to each other. An OWL ontology also consists of a set of descriptions which place constraints on the classes and the relations between them. These descriptions provide semantics by allowing systems to infer additional information based on the data explicitly provided.

OWL provides three increasingly expressive sublanguages designed for use by specific communities of developers and users.

- OWL Lite supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It was hoped that it would be simpler to provide tool support for OWL Lite than its more expressive relatives, allowing quick migration path for systems utilizing thesauri and other taxonomies. However Owl Lite also has a lower formal complexity than OWL DL.

- OWL DL was designed to support the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable), and the availability of practical reasoning algorithms. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL.

- OWL Full is developed for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. It is based on a different semantics from OWL Lite or OWL DL. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any

reasoning software will be able to support complete reasoning for every feature of OWL Full.

3.3 Applications of Ontology-based System

In recent years, ontologies have been adopted in many businesses and scientific communities as a way to share, reuse, and process domain knowledge. Due to complex structure and most of undescribable relationship in context, the key role of ontologies in biomedical and gene expression studies has been led to the rapid development, such as medical ontology (Pisanelli, 2004), the Open Biomedical Ontologies (OBO), and the Gene Ontology (GO) (Ashbrner et al., 2000). This section reviewed two ontology-based developments, named Gene Ontology and Brain-Gene Ontology.

3.3.1 Gene Ontology

Gene Ontology (GO)is available online (http://www.geneontology.org/) and provides a controlled vocabulary to describe gene and gene product attributes in any organism. The knowledge structure of GO refers to a biological objective to which the gene or gene product contributions. GO can be broadly split into two parts: ontology and annotation.

The ontology of GO is actually composed of three ontologies: the molecular function, biological processes, and cellular components. Molecular function, biological process and cellular component are all attributes of genes and gene products. In the GO, these three categories are also named as:

- **Molecular Function ontology** presents molecular functions of a gene product are the jobs that it does or the "abilities" that it has.
- **Biological Process ontology** represents collections of processes as well as terms that represent a specific, entire process. The processes generally are represented as format of the relations in terms of ontology representations such as 'part_of'.

- **Cellular component ontology** describes locations, at the levels of subcellular structures and macromolecular complexes.

Please notice that GO is not the Semantic Web. It is only a online database to which an ontology presentation has been added. The ontology part is implemented by three categorised ontologies. The database or document part is used annotation. Annotation is the process of assigning GO terms to gene products. The annotation data in the GO database is contributed by members of the GO Consortium, and the Consortium is actively encouraging new groups to start contributing annotation.

3.3.2 The KEDRI's Brain-Gene Ontology (BGO)

Brain-Gene Ontology (BGO) is an newly developed ontology based system by KEDRI (Kasabov et al., 2007, 2008). It includes sonseptual and factual information about the brain and gene functions and their relationships as shown in Figure 3.6. The BGO describes the knowledge of brain functions and brain disease, and brain related genes and their activities. BGO is implemented in the Protégé ontology-building environment that is developed by the Medical Informatics Department of the Stanford University (http://protege.stanford.edu/index.html).

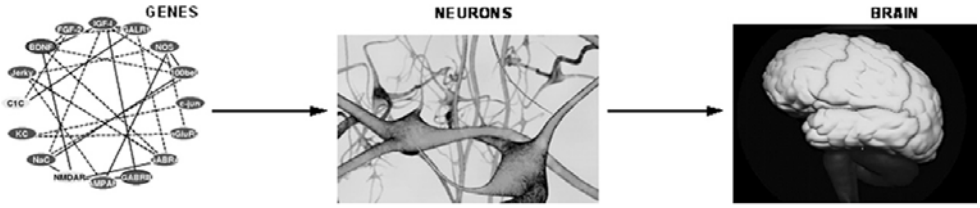


Figure 3.6: *Brain-Gene ontology is constructed based on the complex relationships between genes, their influence upon neurons and the brain (Kasabov et al., 2007 & 2008).*

Information structure of the BGO is comprised of three main parts: brain organization and function, gene regulatory network, and a simulation model.

- Brain organization and function is focused to build knowledge structure of neurons and inforamtion process of spike generation.

- Gene regulatory network structures neurogenetic processing gene expression regulation, protein synthesis. It also visually simulates the gene regulatory network .

- Simulation model describes computational neurogenetic modeling (CNGM) (Benuskova and Kasabov, 2007), evolutionary computation, evolving connectionist systems (ECOS) (Kasabov 2003), spiking neural network (Kasabov and Benuskova 2004) in a ontology-based presentation.

The information of BGO is based on Gene Ontology, Unified Medical Language System (UMLS) and the most used biological data sources.

One of the advantage of the BGO to the other recent ontology system is that data from the BGO can be used in simulation systems such as NeuCom, WEKA, and others. The outputted results can be added back to the BGO to visualise relationship information and further discoveries. The BGO allows users to navigate through the information space of brain diseases, brain related genes and their activities. But the discovery on interaction patterns between brain genes and brain diseases is still a gap in the BGO. This motivates the study of this thesis. In the next chapter, we develope an extension ontology framework for BGO system that is called “CNS Ontology”.

Chapter 4

A Methodology for Ontology-based Gene Expression Analysis and Personalised Modeling

In Chapter 2, I have reviewed the Pomeroy's work on CNS gene expression data. They have successfully developed a gene expression analysis based on CNS tumour gene expression data. Over half a decade, many researchers have undertaken similar experiment in order to develop more accurate modeling methods. But the interactions between genes and CNS cancer are still an open question in this domain. Our research addresses this issue and aims to develop an accurate model and a knowledge-based ontology for visualising the predicted results, see Figure 4.1.

For the knowledge modeling research, I have developed an one-vs.-all scheme SNR (OVA-SNR) method for gene selection as presented in Section 4.2. It is followed by two case studies for preforming the classification accuracy with OVA-SNR gene selection. For the classification, six algorithms have been investigated, including: multilayer perceptron, support vector machine, evolving classifier function, K-nearest neighbours, weighted K-nearest neighbours and weighted-weighted K-nearest neighbours. Except the last one, all of these algorithms have been discussed in Chapter 2. We will summarise the concept of WWKNN in Section 4.3. Since the original WWKNN is only suitable for two-class problem, I have discussed two approaches to solve multi-class problems. Finally the development of CNS gene expression ontology

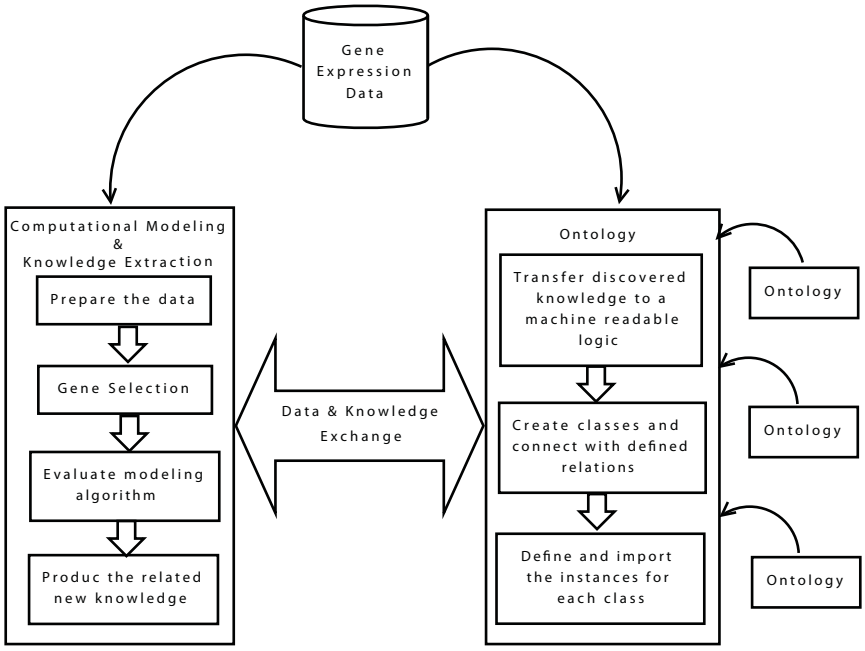


Figure 4.1: *General work flow of this research*

is discussed in Section 4.4.

4.1 Modeling Experiment

The experiment of modeling is built based on a supervised learning classification. The supervised learning is the method that involves training classifier to recognise distinctions among the defined classes, and testing accuracy of the classifier (Kotsiantis, 2007). The methodology can be summarised as follows:

- Define desired class labels based on morphology, tumour class or treatment outcome information. The class label must be meaningful to both researchers and computer softwares.
- Select the highest correlative genes with the target class.
- Build six classifier in leave one out cross-validation by removing one sample for testing and then used the rest as a training set.

- Several models are built with different parameters for each classifier and final chosen model is the one that minimises the total error in cross validation.
- Compare the chosen results from each classifier.

In this process, there are two steps that we would like to discuss in this chapter. One is the gene selection. The other one is classifier.

4.2 Gene Selection

In a gene expression analysis, the gene selection procedure is independent of the classification process, which is applied before the classifiers. It finds informative genes and remove the noise. Currently, there are many different approaches of gene selection available. Chapter 2 has reviewed two of the most popular methods, including T-test and SNR. However, I decided to use SNR as our main gene selection method in this thesis. There are two reasons for employing SNR, especially in CNS cancer diagnosis and treatment area. They are summarised as follows:

1. In the microarray data analysis, T-test can be only used under the assumptions that the experimental data is normally distributed, and the population variances are equal in two classes. In real cases of cancer datasets, these assumptions are difficult to be made. Genes are dynamic information that could performs quite different on every person, especially with cancer disease. This issue indicates that the selected genes by T-test are not reliable in terms of cancer diagnosis.
2. In the content of gene selection, there is always a open question to determine whether a selected gene is closely correlated with target class(Slonim et al., 2000). If the selected genes are correlated with more than one class, that may make a noise to the classifiers. The algorithm of SNR looks at the difference of the means in each of the classes scaled by the sum of the standard deviations as Equation 4.1. The algorithm of T-test is presented as Equation 4.2.

$$S_i = \left|\mu_i^{class1} - \mu_i^{class2}\right| / \left(\sigma_i^{class1} - \sigma_i^{class2}\right) \tag{4.1}$$

$$T_i = \left|\mu_i^{class1} - \mu_i^{class2}\right| / \sqrt{\left(\frac{1}{n_i^{class1} + n_i^{class2}}\right) \times \sigma_i} \tag{4.2}$$

Both of algorithms are quite same except the denominator. However the $\left(\sigma_i^{class1} - \sigma_i^{class2}\right)$ of SNR is always great than $\sqrt{\left(\frac{1}{n_i^{class1}+n_i^{class2}}\right) \times \sigma_i}$ of T-test. Thus, the Signal-to-Noise statistic penalises genes that have higher variance in each class more than those genes that have a high variance in one class and a low variance in another. This bias is perhaps useful for biological samples, e.g. in a case of tumour versus normal where in one class, the gene is working normally and regulated relatively strictly, and in the other class the gene is broken and varying more widely (Ross, 2003).

From the above discussion, we can see the benefits of SNR in terms of cancer diagnosis studies. However, the most gene selection methods are developed for the two-class problems (Leung et al., 2006). This lead us to develop a new SNR based gene selection method, since three multi-class problem are involved in this thesis. The following subsection describes this method in details.

4.2.1 One-Vs.-All based SNR Gene Selection Method

The multi-classes problems normally have more noise than two-class in gene selection (Chai and Domeniconi, 2004). Pomeroy (2002) has introduced a class separation based selection method, which separately selects genes based on the value for permutation tests of marker genes on one class. In fact, this method requires intensive computational power, and the final accuracy of classification is not so well. In this thesis, we introduce an one-vs.-all(OVA) scheme that is built on top of SNR gene selection method.

One-Vs.-All (OVA) is firstly introduced as a multi-class classification scheme. It built on top of real-valued binary classifiers is to train N different binary classifiers, each one trained to distinguish the examples in a single class from the examples in all remaining classes (Rifkin and Aldebaro , 2004). Based on the OVA scheme a complex multi-class problem is split to several simpler two-class problem. Generally the training single class is called “target class” and all remaining classes are called

"other class" in each training process. This scheme is also able to implement into gene selection methods. Suppose we have a multi-class problem with 6 classes. Under the OVA scheme, we take one class as "target" class, the "other" class will contain the remaining samples from other five classes. Representative genes of the "target" class with respect to the others are selected by applying a two-class gene selection method. This process is performed 6 rounds (same as number of classes). Each round will select the same number of genes to represent one of the six classes. For example, we select 6 genes in each round. A total of 36 genes are selected for further experiment.

In our thesis, we develop our SNR gene selection method based on OVA scheme. Generally, the gene selection of each dataset can be summarised as following steps:

1. Split the classes into 2 groups
2. Select same numbers of genes that up-regulate each target class samples.
3. Check the repeated feature genes between different classes, and delete the repeating ones.
4. Combine those selected genes as one selected gene subset.

This OVA scheme SNR is developed not only for the multi-class problem but also for two-class problem. In the two-class problem, the gene selection process is performed 2 times and each time we select the same number of genes to represent one class. The following case studies present how to apply OVA-SNR on both multi-class and two-class problem. It also presents the classification accuracy of the models that are built on the OVA-SNR selected genes.

4.2.2 Case Studies on Gene Selection

Two cases studies are discussed in this subsection. Pomeroy's dataset A (multi-class problem) is used in case one. Case two is based on Pomeroy's dataset B (two-class problem). The OVA scheme is programed in MATLAB. The SNR gene selection is provided by NeuCom (www.kedri.info). It applies the same algorithm as presented in Equation 4.1.

In order to determine whether OVA-SNR gene selection method is efficacious for microarray experiment, we build two models (SVM and kNN) based on the selected genes to present classification accuracies. Both of classification models are operated under leave one out cross validation with the linear normalised datasets. In case studies, we also use bio-clustering technique to identify correlationship between samples and genes.

Multi-class Gene Selection

Dataset A consists of 42 samples with 7,129 genes. This dataset is split into five classes. In this case, 50 genes (10 genes for each class) are selected using OVA-SNR. The correlation between genes and samples is shown in Figure 4.2. The dashed lines between columns are used to split different classes. The squares represent the correlarionship between genes and samples. The squares with darker color indicate a higher correlationship between the samples. The squares with lighter color indicate a lower correlationship between gene and sample. A gene has higher correlation with its "target" class, that is described as a signal in terms of "signal to noise ratio". A gene has higher correlation with "other" class, that is presented as a noise to classification.

From this figure, we can see that the samples have high correlated genes (darker color squares) of each single class are generally gathered together, and most of them are also indicated as lower correlation (lighter color) in other classes. This implies the corresponding genes of individual class can be clearly selected by OVA-SNR. However the

We built several models on 50 selected genes by using SVM and kNN. Each model is built with different parameter. Table 4.1 summarises the best results of SVM and kNN. Performance accuracy is calculated by the number of correctly classified samples over the total number of samples in the dataset.

	50 selected genes
SVM (linear)	92.86%
kNN (K=4)	90.48%

Table 4.1: *Performance accuracy of SVM and kNN that is built on dataset A with the 50 OVA-SNR ranked genes.*

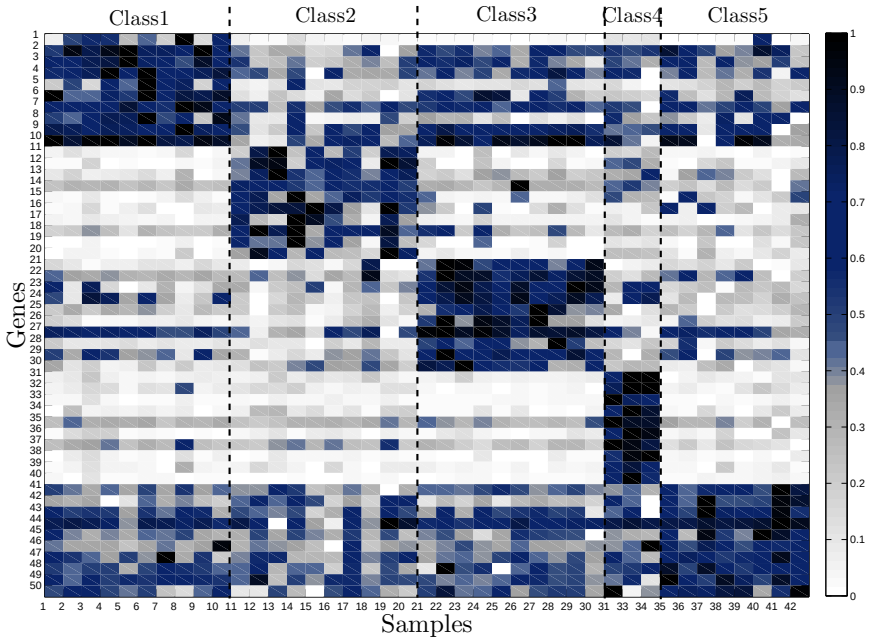


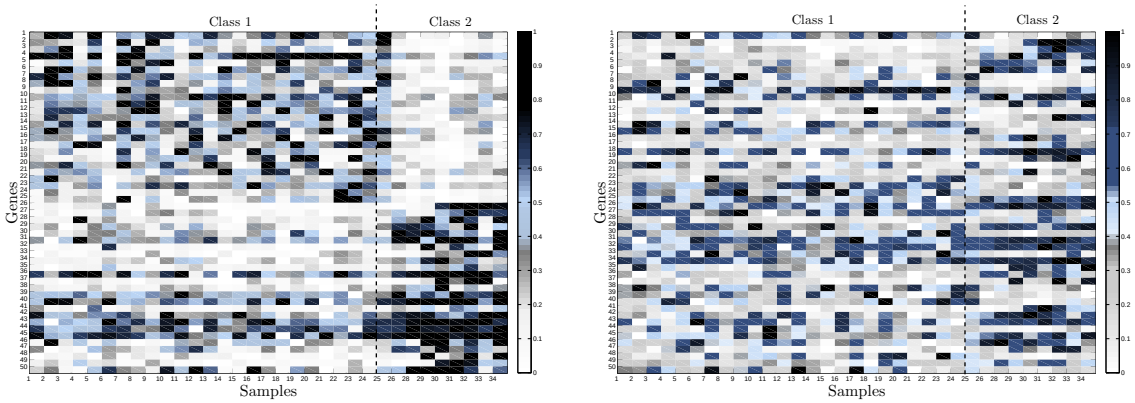
Figure 4.2: *The correlation between the genes and samples. Genes 1-10 are correlative genes to class 1. Genes 11-20 are correlative genes to class 2. Genes 21-30 are high correlative genes to class 3. Genes 31-40 are high correlated genes to class 4. Genes 41-50 are high correlated genes to class 5.*

This table indicates that both models achieve the quite high accuracy. This suggests that the 50 OVA-SNR selected genes are closely correlated with their target classes. The OVA-SNR gene selection method is efficacious for this multi-class problem.

Two-class Gene Selection

In this case study, our interest is not only the performance of OVA-SNR on two-class problem, but also the different performance between the OVA-SNR and original SNR. The dataset of this case is Pomeroy's dataset B which includes 34 samples with 7139 genes. In this case, we separately capture 50 genes by using two gene selection methods, including OVA-SNR and original SNR. For OVA-SNR, 25 genes are selected for each class. Figure 4.3 shows the different performance on correlations between OVA-SNR and original SNR. The subfigure (a) presents the 50 OVA-SNR selected genes. The subfigure (b) presents the 50 original SNR selected genes. By contrast the subfigure (a) and (b), the difference between the OVA-SNR selected

genes and original SNR selected gene is very clear. The genes that are selected by original SNR are evenly separated in the both class 1 and class 2. There is no any clear signals in subfigure (b). On the contrary, OVA-SNR selected genes perform more signals rather than noises. The correlative genes (genes 1-25) of class 1 preform lower correlative value in class 2. On the other hand, the correlative genes (genes 26-50) of class 2 preform lower correlative value in class 1. To compare the classification accuracy with different methods selected genes, we also apply SVM and kNN.



(a) (b)

Figure 4.3: *Comparison of OVA SNR and Normal SNR on dataset B. In the subfigure (a), genes 1-25 represent the correlative genes of class 1, and gene 26-50 represent the correlative genes of class 2. The signals of dataset B to each class are clearly presented in subfigure (a).*

Table 4.2 summarises the best results of SVM and kNN models on dataset B. The classification accuracies that obtains from the OVA-SNR are higher than original SNR selected genes. This result supports our OVA-SNR is better gene selection method in the two-class problem.

	OVA-SNR	**Original SNR**
SVM (linear)	94.12%	88.24%
kNN (K=5)	91.18%	88.24%

Table 4.2: *Performance accuracy of SVM and kNN that is built on dataset B with the 50 OVA-SNR ranked genes.*

4.2.3 Summary of OVA-SNR Gene Selection

The results of case studies suggest that OVA-SNR method is very efficacious in terms of correlated genes selection. This performs on the highly accurate models in both multi-class and two-class problems. Unlike to the original SNR, the OVA-SNR select the genes in different rounds, each round targets only one target class. This scheme ensures that each selected gene could highly correlate with only one target class. After gene selection, the next step of experiment is the classifier. The next section will discuss a newer personalised modeling algorithm, weighted-weighted k nearest neighbour.

4.3 Classifiers

Gene expression analysis has been proved efficacious in cancer diagnosis. A substantial number of methods and models have been implemented in which impressive results have been reported in different experiments. However, many evidences also suggest that the problem of patients to therapy is the highly variable response for individuals. Personalised modeling has been reported efficient for solving this problem, since it models the proposed problem based on the individual sample. This section describes weighted-weighted k nearest neighbours, a newly developed personalised modeling method that not only explores the appropriate solution of queried sample, but also ranks the important genes based on the queried patient.

4.3.1 Weighted-Weighted K Nearest Neighbours

Weighted-weighted K nearest neighbour (WWKNN) is a newly developed personalised model by KEDRI (Kasabov, 2006). The main idea behind WWKNN algorithm is: the K nearest neighbour vectors are weighted based on their distance to the new vector, and also the contribution of each variable is weighted according to their importance within the local area where the new vector belongs to (Song and Kasabov, 2006). It is assumed that the different variables have different importance to classify samples into different classes when the variables are ranks in terms of their discriminative power of class samples over the whole V-dimensional space. There-

fore, the variables probably have different ranking scores of the discriminative power of the same variables is measured for a local space of the problem space.

In WWKNN algorithm, the Euclidean distance d_j between a query vector x_i and a neighbour x_j is calculated as follows:

$$d_j = \sqrt{\sum_{i=1}^{V} c_{i,l} \left(x_{i,l} - x_{j,l}\right)^2} \tag{4.3}$$

where $c_{i,l}$ is the coefficient weight of variable x_l for the nearest neighbours x_j, and k is the number of the nearest neighbors. The coefficient weight is calculated by SNR as equation 4.1 that ranks each variable across all vectors in the local area:

$$c_i = (c_{i,1}, c_{i,2}, ..., c_{i,V}) \tag{4.4}$$

$$c_{i,l} = \left(1 - S_l / \sum_{l=1}^{V} S_l\right), (l = 1, 2, ..., V), where \tag{4.5}$$

where μ_l^{class1} and δ_l^{class1} represent the mean and standard deviation x_l for all vectors in neighbourhood set D_i that belong to class 1.

The final output y_i of x_i is calculated by using the Equation 4.6. In order to finally classify the queried vector x_i into one of classes, there has to be a probability threshold P_{thr} selected (Kasabov, 2007). In a two-class problem, there is only one probability that is normally denoted as P_{thr}. If output y_i is greater than P_{thr}, then the sample x_i is classified in class 2.

$$y_i = \sum_{1}^{K} \left((1 - d_j) \times y_j\right) / \sum_{1}^{K} (1 - d_j) \tag{4.6}$$

Comparing to other variants of classical kNN method, the new feature of WWKNN is the new distance: all variables are weighted according to their importance as discriminating factors in the neighbourhood area.

4.4 Multi-classes Classifications in WWKNN

In the beginning of WWKNN development, the algorithm was only considered to solve two-class problems. In other words, WWKNN was able to identify a sample either class 1 or class 2 only. But, Pomeroy's datasets involves three mulit-class and two two-class problems. This leads us to investigate the WWKNN in the multi-class problems. There are two approaches of WWKNN are developed. One uses multilayer threshold. The other approach is built on the One-Vs.-All (OVA) scheme.

4.4.1 Multilayered Threshold Approach

WWKNN is used to solve a classification problem. The calculated output y_i for a queried vector x_i is a "personalised probability". Then we need to compare the y_i with the probability threshold P_{thr}. Generally speaking, the P_{thr} in the two-class problem is a line that splits the samples into two classes. The samples that are under this line of P_{thr} (less than P_{thr}) is classified into class 1. Samples that is above the line of P_{thr} (greater than P_{thr}) is classified into class 2.

In two-class problem, we split the samples into two layers by using one threshold line. For the multi-class problem, the samples will be split into multi-layers that will need multi threshold lines to present layers. Based on this theory we develop a simple approach that calculates the probability threshold (P^j_{thr}) for different layers as follows:

$$P^j_{thr} = (c_j - 1) + P_{thr}, j = 1 : J \tag{4.7}$$

where c_j is the desired class label that is presented between 1 to J. The P_{thr} is chosen between 0 and 1. The bottom layer is under P^1_{thr}. The top layer is above P^J_{thr}

An output y_i that is greater than P^j_{thr} and less than P^{j+1}_{thr} is classified into class c_j. If output y_i is less than P^1_{thr} or greater than P^J_{thr}, then the queried vector x_i would be classified into class c_1 or c_J. The approach can be generated into the following steps:

1. Normalise input data and select a threshold that is between 0 to 1.

2. Identify total numbers of classes based on the original data.
3. Calculate layers of threshold by Equation 4.7.
4. Select the method of cross validation and the number of the nearest neighbour.
5. Input a new sample into the problem space.
6. Apply WWKNN function.
7. Calculate the output y_i by Equation 4.6.
8. Compare the output y_i with the layered thresholds P_{thr}^{j}.
9. Classify the output y_i into one of classes based on P_{thr}^{j}.

4.4.2 OVA Approach

The second approach of multi-class WWKNN is based on the OVA scheme. The similar technique is also used in linear SVM to solve the multi-classes problem (Joachims, 2006). We have described the OVA theory in Section 4.2. In the problem of classification, OVA approach is proposed to split a multi-class problem to several two-class problems.

Suppose we have been given a CNS dataset with 5 classes. These classes are named medulloblastomas, malignant gliomas, primitive neuroectodermal tumours (PNETs), atypical teratoid/rhabdoid tumours and normal human cerebellum. First, we split the multi-class problem into a series of 5 problems, and each split problem is proposed by one specific class classification (e.g. Medulloblastoma vs. other classes those are not Medulloblastoma). This process is described in Figure 4.4.

In this approach of classification, each new vector is presented sequentially to these 5 separated problems, each sequence will identify whether or not that sample belongs to the target class. The whole processes of classification can be simply summarised into the following steps:

1. Normalise input data and select a threshold that is between 0 to 1.
2. Identify the total number of classes based on the original data.

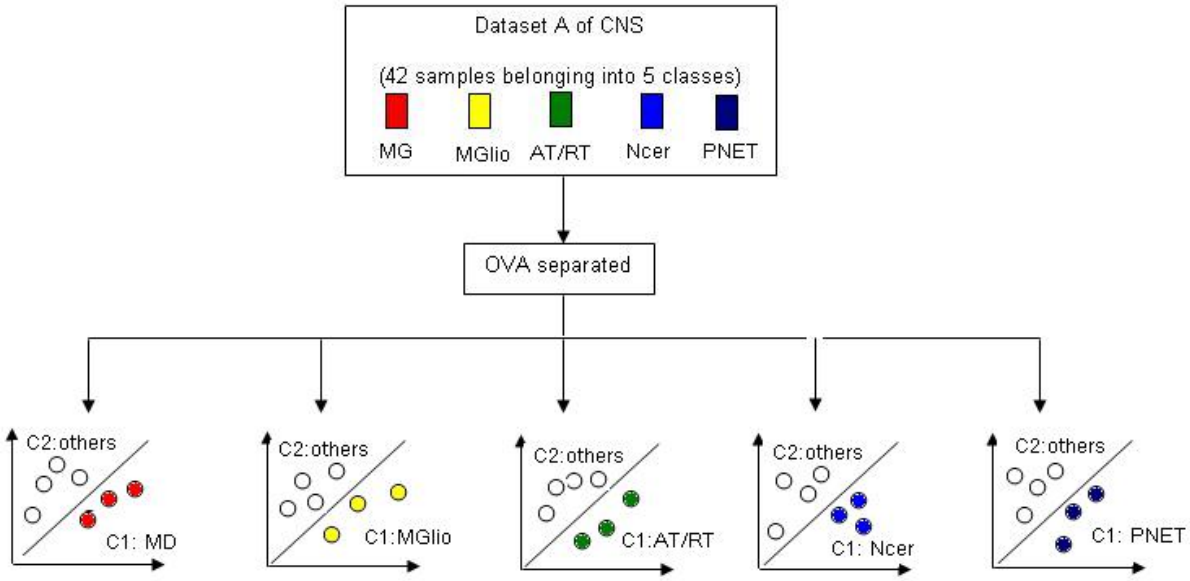


Figure 4.4: *OVA WWKNN*

3. Select the method of cross validation and the input number of the nearest neighbour that is going to be used.

4. Separate the original data into series of datasets that have the same number of classes. Each separated dataset has two classes: the target class and a class of the others.

5. Input a new sample into the problem space.

6. Apply WWKNN function to classify between the target class and a class of the others.

7. Sequentially repeat steps 5 and 6 for until all the separated datasets are used.

8. Compare their Euclidean distance with the thresholds.

In this approach, we do not need to calculate the different threshold layers, because we actually classify queried vector into two classes. In each round of OVA classification, we only record the accuracy of target class. The final accuracy of classification is based on the collection of target class accuracies. By contrast both approaches, here is a open question, which approach is more accurate? We provide the answer at the end of Chapter 5.

4.5 Knowledge Representation with Ontology

4.5.1 CNS Ontology

For the knowledge presentation and discovery, we construct an ontology-based framework to represent the knowledge based patterns in thr domain of CNS tumour. This ontology is called CNS ontology. In this thesis, all the information and knowledge discovery from Pomeroy's data are stored in this ontology based framework. CNS ontology is proposed to be an extension of Brain Gene Ontology (BGO) system, therefore it is built based on the knowledge content of BGO. Some of knowledge is acquired from the external ontology system (e.g. Gene Ontology) or from literature database such as National Center for Bio-technology Information (NCBI). The relationship between CNS ontology, BGO, GO and Pomeroy's data is shown in Figure 4.5.

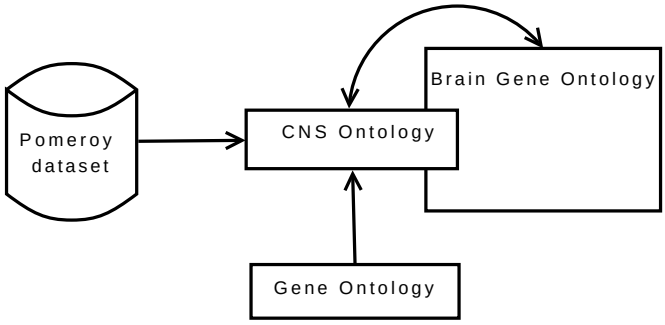


Figure 4.5: *Relationship between Pomeroy's data, BGO, GO and CNS ontology. The literature of Pomeroy provides a basic knowledge to structure the patterns in CNS knowledge domain. The structured knowledge of CNS can be exchange with the BGO. The GO provides the external knowledge to the CNS ontology*

CNS ontology is implemented by Protégé that-a free, open-source ontology-building environment that provides users to construct domain models and knowledge-based applications(http://protege.stanford.edu/). At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualisation, and manipulation of ontology systems in various representation formats. The Protégé supports two main formats of ontology:

- The Protégé-Frames editor enables users to build and populate ontologies that are frame-based, in accordance with the Open Knowledge Base Connectivity

protocol (OKBC). In this model, an ontology consists of a set of classes organised in a subsumption hierarchy to represent a domain's salient concepts, a set of slots associated to classes to describe their properties and relationships, and a set of instances of those classes - individual exemplars of the concepts that hold specific values for their properties.

- The Protégé-OWL editor enables users to build ontology for the Semantic Web, in particular in the W3C's Web Ontology Language (OWL). "An OWL ontology may include descriptions of classes, properties and their instances. Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailment may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms".

CNS ontology is constructed using Protégé-OWL editor, since OWL provides more wide range of external knowledge exchange throughout Internet. For internal knowledge exchange, we convert the frame-based BGO ontology to OWL format by using an ontology format conversion tool box in Protégé.

4.5.2 The Factors of CNS Ontology

Chapter 3 has reviewed the factors of ontology-based knowledge framework. This subsection describes three important factors of CNS ontology that are classes, relations and individuals.

Classes

Based on the content of Pomeroy's data, CNS ontology consists of three superclasses. These are called "Samples", "Gene_bank" and "Clinical_attributes". These classes are in the top level of the hierarchical ontology structure as shown in Figure 4.6

The class "Clinical attributes" has three subclasses that are called "Common_information_of_sampl "Treatment_information" and "Types_of_CNS". Each one of them is subsumed by other subclasses. "Common_information_of_samples" class contains by "Gender",

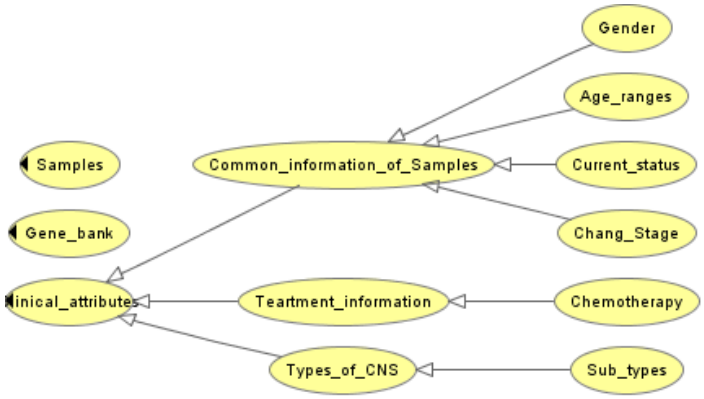


Figure 4.6: *Snapshot of CNS ontology showing hierarchical structure in CNS knowledge domain.*

"Age_ranges", "Current_status" and "Chang_stage". "Treatment_information" has a subclass "Chemotherapy". "Types of CNS" also has one subclass "Subtype".

Relations

As presented in Chapter 3, two types of relations can be defined in ontology including parent-ship relations and object relations. The parent-ship relations have already been described in above paragraph. In CNS ontology, the "Samples" class and "Common_information_of_samples" class are connected through an object relation that is described as the word "has". Since the class "Common_information_of_samples" has six subclasses, its subclasses are inherited the "has" relation to connect with Samples as shown in Figure 4.7

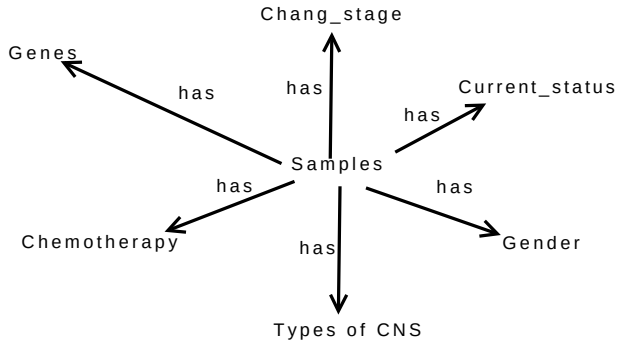


Figure 4.7: *The objects relations between the Samples class and subclass of the Common information of samples class.*

Individuals

The individual presents an instance of each class. In CNS ontology, the individuals are defined from the related literature and supplement information of Pomeory's study (2002). Figure 4.3 shows the list of individuals in CNS ontology

Classes	Individuals
Genes	The discriminative genes that are discovered by modeling methods
Samples	99 samples
Current status	Alive, Dead
Gender	Male, Female
Chang stage	M0, M1, M2, M3, M4, T1, T2, T3, T3b, T4
Chemotherapy	Carboplatin, CCNU, Cisplatin, Cytoxan Etoposide, Methotrexate, Procarbazine Thiotepa, Vincristine
Types of CNS	AT/RT, Malignant Glioma, Medulloblastoma Normal cerebellum, PNET
Subtypes	Classic, Desmoplastic

Table 4.3: *List of individual in CNS ontology.*

The individuals of class Genes are identified by using their gene accession numbers. These gene accession number are also meaningful to online gene banks such as Entrez and PubGene.

4.5.3 The Connection between the CNS Ontology and BGO

One purpose of using ontology is to support the sharing and reuse of knowledge by making it possible for ontology to import another ontology. To achieve this purpose, we need to build knowledge pathways from one ontology to others. When the ontology systems are connected, all of the classes, relations and individuals of imported ontology are available to define and use in the importing ontology. As mentioned in the Chapter 2, OWL-based ontology systems are identified as URIs by using metadatas, such as http://www.owl-ontology.com/the_CNS_ontology.owl.

In order to connect CNS ontology and BGO system, we firstly import the CNS document into BGO as shown in Figure 4.8. OWL imports statement is presented as

the form of (owl:imports rdf:resource="http://www.owl-ontology.com/brain_gene_ontology"), where resource is the URIs (metadata) of BGO system.

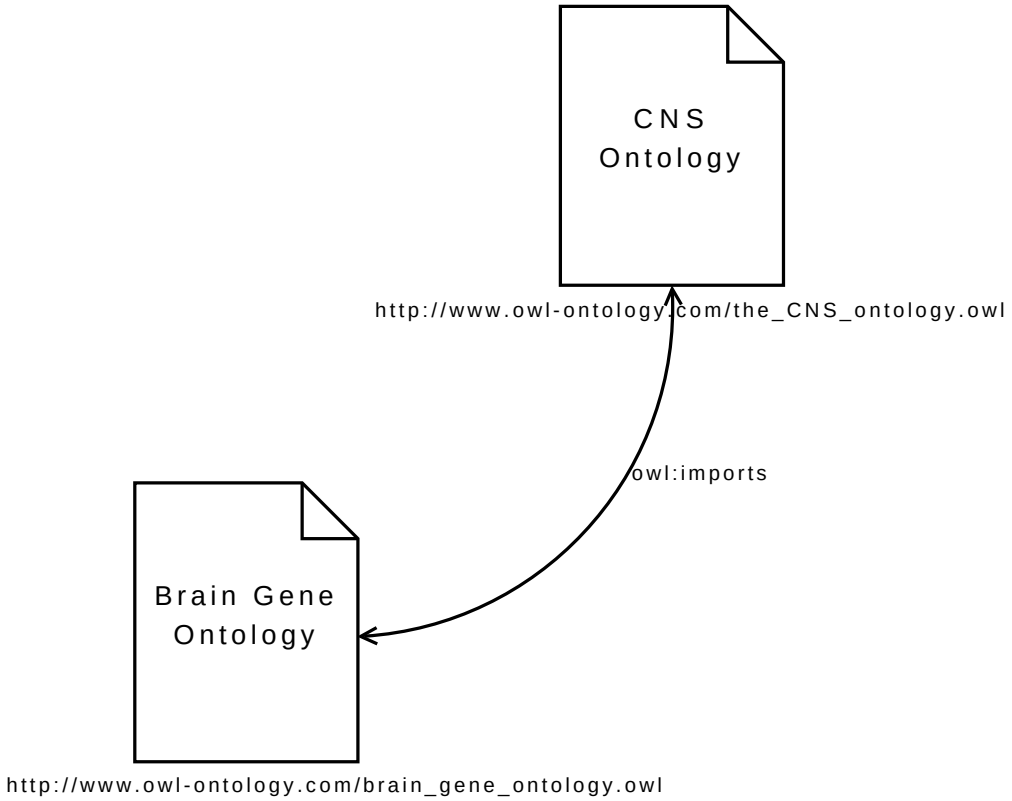


Figure 4.8: *The connection between CNS ontology and BGO.*

Above operations describe that CNS ontology and BGO are under same knowledge domain. After CNS ontology is imported to BGO, we need to build a relation between CNS ontology and BGO. This relation is the main pathway for the knowledge exchange between CNS ontology and BGO. We created an object relation, "has_BGO_document" that connects from the "Gene_bank" class of CNS ontology to the "Genes" class in BGO. The ontology representation in this relation is "Gene_bank → has_BGO_document → Genes". This relation can also be reversed back as "Genes → has_BGO_document → Gene_Bank". Throughout this knowledge pathway, the instance genes from CNS ontology can be fund related gene document in BGO, or follows the object relation that is between Gene class and other classes to find other level relations such as interactive proteins. The gene from BGO can also be defined the related CNS disease in CNS ontology.

CNS ontology is also able to be self-learned and consequently extracts certain hidden information and knowledge from the existed information in the knowledge structure. However, before we operate CNS ontology for research. We have to model those thousands of genes first, and transfer these accurate models and into an ontology presentation. The next chapter is focused on the gene expression experiment that would contribute to the discriminant genes discovery.

Chapter 5

Experimental Results on the CNS Case Study Problem

5.1 Datasets

The proposed experiment in this thesis is designed on a well-known benchmark CNS cancer micorarray data (available at http://www-genome.wi.mit.eud/mpr/CNS/) (Pomeroy et al., 2002). This data contains the expression levels of 7,129 genes across 99 samples that consists of 67 children with medulloblastomas, 10 young adults with malignant gliomas, 5 children with AT/RT, 5 with renal/extrarenal rhabdiod tumours, 8 children with supratentorial PNETs and 4 normal cerebellum samples. This data were organised into five datasets. Table 5.1 summarises these five datasets used for classification in the experiment.

Dataset name	Type of problem	Training Samples	Ref.
Dataset A	Multi-class problem	42	1
Dataset A1	Multi-class problem	40	2
Dataset A2	Multi-class problem	90	3
Dataset B	Two-class problem	34	4
Dataset C	Two-class problem	60	5

Table 5.1: *Summary of microarray datasets used for experiment*

1. **Dataset A** has 42 samples in CNS tumour patients, that consists of 10 medulloblastomas (class 1), 10 malignant gliomas (class 2), 10 atypical teratoid/

rhabdoid tumours (class 3), 4 normal cerebellum (class 4), and 8 PNETs (class 5).

2. **Dataset A1** contains 40 samples across 5 classes of CNS tumours. This dataset is removed two pineoblastoma samples from the dataset A. Dataset A1 consists of 10 medulloblastomas (class 1), 10 malignant gliomas (class 2), 10 atypical teratoid/ rhabdoid tumours (class 3), 4 normal cerebella (class 4), and 6 PNETs (class 5).

3. **Dataset A2** contains 90 samples which consist of 60 medulloblastomas (class 1), 10 malignant gliomas (class 2), 10 atypical teratoid/ rhabdoid tumours (class 3), 4 normal cerebella (class 4), and 8 PNETs (class 5).

4. **Dataset B** contains 34 samples, 25 classic medulloblastomas (class 1) vs. 9 desmoplastic medulloblastomas (class 2).

5. **Dataset C** contains 60 samples in medulloblastoma patients. Among them 21 samples are failures (class 1) while 39 samples are survivors (class 2). Survivors represents the patients who are alive after the treatment. Failures are those who succumb to the central nervous system cancer.

5.2 Experimental Purposes

This CNS dataset has been so far extensively studied, and many models and approaches have been developed for its classification task. However, most papers are focused on the computational accuracy of the performance. In this experiment we are more interested in what knowledge can be discovered from these models and whether the knowledge can be reused for cancer diagnosis.

As suggested in literature (Pomeroy et al., 2003), each dataset in CNS cancer microarray data was collected with a particular research purpose as following:

- Dataset A, A1 and A2 were used to determine whether the CNS multi tumour classes are separable based on gene expression.
- Dataset B was used to describe the classifications of classic and desmoplastic in medulloblastoma morphology.

- Dataset C was proposed to present medulloblastoma treatment outcome by using gene expression.

Our experiment will carry two more purposes for this study. The first purpose is to make a comparison of global, local and personalised modeling methods for CNS cancer diagnosis and prognosis. A variety of methods based on global, local and personalised modeling, including Multilayer perceptron (MLP), Support Vector Machine (SVM), Evolving Classifier Function (ECF), k-Nearest Neighbours (kNN), Weighted K-Nearest Neighbours (WKNN) and Weighted-Weighted K-Nearest Neighbours (WWKNN) are investigated. The second purpose of our experiment is to discover a reusable knowledge pattern for cancer diagnosis and prognosis in terms of gene expression. The output of this purpose is discovered based on the achievement of the first research purpose. The following sections describes the experiment process of this research.

5.3 Experiment Setup

Hardware and Software

The experiments are implemented under a MATLAB Development Environment on a computer with 3.2 GHZ Pentium 4 and 1024 Mb RAM. Relevant software and implements used for classification and modeling in the experiments are summarised in Table 5.2.

Software/Algorithm	Note	Availability
NeuCom	Nero-computing decision support system	www.kedri.info
PCA	Principal component analysis	NeuCom
SNR Algorithm	For gene selection	NeuCom
MLP Algorithm	For microarray dataset classification	NeuCom
SVM Algorithm	For microarray dataset classification	NeuCom
ECF Algorithm	For microarray dataset classification	NeuCom
KNN Algorithm	For microarray dataset classification	NeuCom
WKNN Algorithm	For microarray dataset classification	MATLAB code
WWKNN Algorithm	For microarray dataset classification	MATLAB code

Table 5.2: *Relevant software used for experiment*

5.3.1 Principal Component Analysis (PCA)

Datasets of large dimensionality are in general difficult to visualise due to the intrinsic difficulty of reducing and projecting the dataset to a small number of dimensions where standard visualisation techniques are applicable. Principal component analysis (PCA) is a commonly used technique to reduce multidimensional data sets to lower dimensions for analysis. This module is designed to capture the variables (principle components) which could explain all of the variance in the original dataset (Jolliffe, 2002).

In this thesis, we employ PCA analysis to visualise the experimental datasets with selected genes. The datasets are plotted based on top 2 principal components, which demonstrate the classification of different classes from the most important parts of the dataset.

5.3.2 Microarray Diagnosis Setup

The first step of the experiment is to normalise the data, which is completed by using linear normalised approach. Each of the models used in our experiment is validated through leave-one-out cross validation (LOOCV) that is described in Chapter 2.

5.3.3 Parameters Setup for Relevant Algorithms

The parameters setting for the proposed gene selection algorithm and six classifiers are summarised as follows:

1. SNR gene selection algorithm:
   An OVA scheme is built on top of SNR.

2. MLP classifier:
   N (Number of hidden neurons): 4-6.
   Number of training cycles: 500.

3. SVM classifier:
   Type of kernel: Linear kernel.

4. ECF classifier:
   Number of epochs: 4.

5. kNN classifier:
   K(K-the number of nearest neighbours): 3-8.

6. WKNN classifier:
   K(K-the number of nearest neighbours): 3-8.
   θ (Threshold): 0.5.

7. Two-class WWKNN classifier:
   K(K-the number of nearest neighbours): 5-15.
   θ (Threshold): 0.5.

8. Layered threshold WWKNN classifier (multi-class):
   K(K-the number of nearest neighbours): 2-7.
   θ (Threshold): 0.2-0.8.

9. OVA classifier (multi-class):
   K(K-the number of nearest neighbours): 2-7.
   θ (Threshold): 0.5.

5.3.4 Selected Genes

Our aim is to use different computational models to analyse the datasets and discover the CNS cancer related knowledge. Selecting fewer genes may limited the discovered knowledge, and often make the operation unreliable, whereas selecting too many genes is usually involve too much noise that may confuse inductive algorithms. Previous studies indicate that a few dozen to a few hundred top-ranked genes can efficiently classify the different disease patterns in most microarray experiments (Li and Yang, 2002). The following words of this subsection describe the process of the selected genes from each dataset.

Dataset A

To compare performances of different classifiers on dataset A, the number of selected gene by OVA-SNR is set as 35. 7 top-ranked genes are selected for each class. The

ranks of top genes are based on their SNR values as shown in Table 5.3

Class Name	Each class	SNR values	Max SNR values
Class 1	7	$\geqslant 0.85$	0.97
Class 2	7	$\geqslant 1.15$	1.30
Class 3	7	$\geqslant 1.27$	1.55
Class 4	7	$\geqslant 3.30$	4.29
Class 5	7	$\geqslant 0.70$	0.83

Table 5.3: *The number of top-ranked genes to be selected in dataset A*

The highest SNR values is produced from class 4 (normal cerebella). This suggests a clearly separable difference of gene expression between cancer patients and normal samples. The lowest SNR threshold is calculated from class 5 (PNETs). PNET stands for a group of tumours since the cells of these tumours look similar under a microscope. Based on this table, we expect a high classification accuracy on class 4, but a lower accuracy on class 5. The list of selected genes are presented in Appendix B.

Dataset A1 and A2

The same selected genes from dataset A are used to do classification over dataset A1 and A2 to compare the different results between three different datasets which are in the same category.

Dataset B

For the experiment of dataset B, 18 genes are selected for each class. The SNR values of selected genes are shown in Table 5.4. The list of selected genes are presented in Appendix B.

Class Name	Each class	SNR values	Max SNR values
Class 1	18	$\geqslant 0.70$	0.99
Class 2	18	$\geqslant 0.72$	0.99

Table 5.4: *The number of top-ranked genes to be selected in dataset B*

Dataset C

Regarding to the genes selected from dataset, their SNR values are quite low in either class 1 or class 2. This implies that the samples of class 1 and samples of class 2 have similiar variance on gene expressions. Due to this issue, the selected genes may easily involve noise variables, which can impact our experimental accuracy. Therefore we selected different 4 sets of top-ranked genes for experimental classification. Then the knowledge discovery is based on the most accurate result. The number of selected gene sets are 30, 40, 50 and 60. The number of genes on each class and their SNR values are shown in Table 5.5.

Number of selected genes	Class Name	Each class	SNR values
30	Class 1	15	$\geqslant 0.36$
	Class 2	15	$\geqslant 0.44$
40	Class 1	20	$\geqslant 0.35$
	Class 2	20	$\geqslant 0.42$
50	Class 1	25	$\geqslant 0.34$
	Class 2	25	$\geqslant 0.40$
60	Class 1	30	$\geqslant 0.33$
	Class 2	30	$\geqslant 0.39$

Table 5.5: *The number of top-ranked genes to be selected in dataset C*

5.4 Experimental Results

5.4.1 Dataset A

The top 7 ranked genes per class are selected as described in the above section. Figure 5.1 indicates the dataset A with 35 top-ranked genes in PCA. It shows that samples of normal cerebellum are well separated from the four types of CNS samples. All of malignant gliomas samples (class 2) are clearly separable in the figure. This is not surprising because the malignant gliomas reflect the derivation of gliomas from cells of non-neuronal origin, which is a significant difference to other types of CNS tumour (Avgeropoulos and Batchelor, 1999). The other three classes are presented quite close to each other.

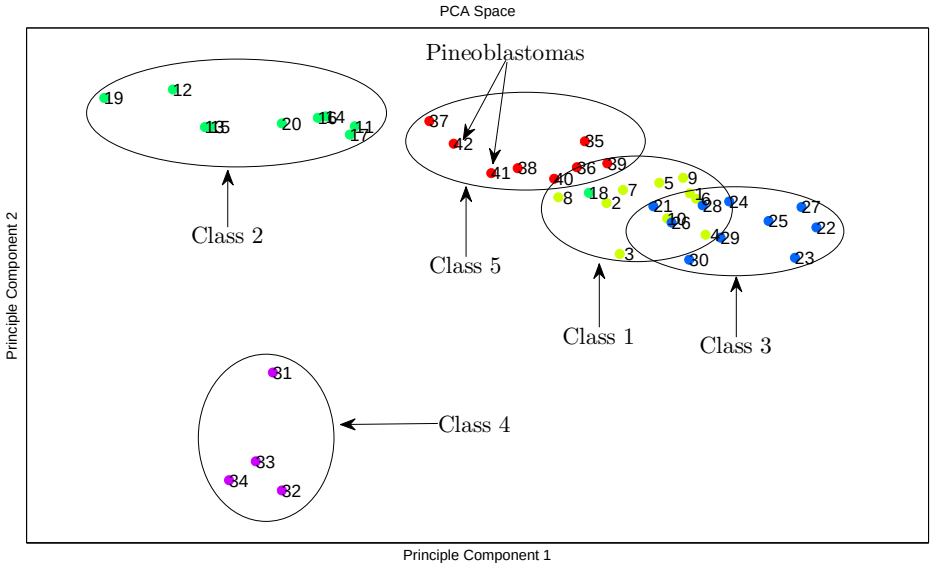


Figure 5.1: *PCA using 35 selected genes that are associated with each tumour type in dataset A.*

In the clinical diagnosis of CNS, medulloblastomas, AT/RTs and PNETs are quite difficult distinguished from each other, since they all occur in the area of cerebellum (Rorke et al., 1995). However, most of samples from these three CNS tumour classes are still separable from each other in PCA figure. This implies that classifiers are able to separate the classes with a low rate of error.

Table 5.6 presents the best experimental results on dataset A, which obtained from each model with a particular parameter. The best classification accuracy (97.62%) achieved on Pomeroy's data is from OVA-WWKNN – 41 out of 42 samples are successfully classified. SVM, kNN and WKNN performs better accuracy than layered-WWKNN, MLP and ECF, which produced 95.24% accuracy. The lowest accuracy is produced by MLP and ECF, that is 88.1%.

From these results, we can see that the accuracy of classification between class 1, class 2, class 3 and class 4 are very balanced across all the classifiers. Because of the close relationship between medulloblastomas and PNETs, six different classifiers excluding OVA-WWKNN produce more errors in class 5. In Pomeroy's work, the best accuracy they achieved is 83.33% through a kNN algorithm with 50 features, which is lower than the result from our experiment.

	MLP	**SVM**	**ECF**	**kNN**	**WKNN**	**Layered WWKNN**	**OVA WWKNN**
	N:4	**linear**	**E:4**	**K:5**	**K:2, θ:0.5**	**K:2, θ:0.8**	**K:8, θ:0.7**
Class 1	90%	100%	100%	100%	90%	90%	100%
Class 2	90%	90%	90%	90%	90%	90%	90%
Class 3	90%	100%	100%	100%	100%	100%	100%
Class 4	75%	100%	100%	100%	100%	100%	100%
Class 5	87.5%	87.5%	50%	87.5%	75%	75%	100%
Total	88.1%	95.24%	88.1%	95.24%	95.24%	90.48%	**97.62%**

Table 5.6: *The best classification result of every applied modeling method on dataset A.*

5.4.2 Dataset A1

PNETs of the CNS are grossly divided into supratentorial PNET and infratentorial PNET. The infratentorial PNET includes medulloblastoma, which occurs in the cerebellum. The supratentorial PNET includes pineoblastoma, which occurs in the pineal region. Dataset A1 is an additional variant of dataset A in Pomeroy's data. The dataset A1 repeats the dataset A excluding 2 pineoblastomas (Sample 41 and 42) from the class 5 (PNETs). Figure 5.2 presents PCA analysis on dataset A1 with 35 selected top-ranked genes. Comparing the results of dataset A and A1, we can not define the differences between two datasets excluding that two PNET samples are missed in the dataset A1.

For the classification of dataset A1, we repeated our algorithms with the same parameter on dataset A1. The results of each classifier on dataset A1 is presented in Table 5.7. This table indicates that OVA-WWKNN achieved the best classification accuracy (97.5%). SVM and kNN produced the better accuracy (95%) than layered-WWKNN (90%), WKNN (85.71%) and MLP (82.5%). The lowest accuracy (80%)is produced by ECF. Overall, the results of dataset A1 are very similar as before. This implies that pineoblastoma is indepentend to infratentorial. The models will not be impacted without them.

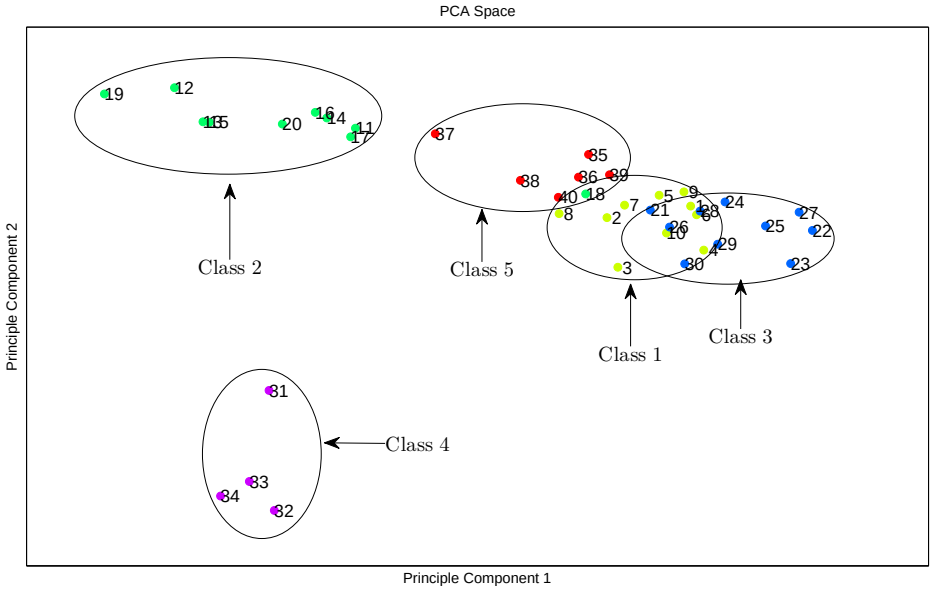


Figure 5.2: *PCA using selected 35 genes to describe the dataset A1.*

	MLP	**SVM**	**ECF**	**kNN**	**WKNN**	**Layered WWKNN**	**OVA WWKNN**
	N:4	**linear**	**E:4**	**K:5**	**K:2, θ:0.5**	**K:2, θ:0.8**	**K:8, θ:0.7**
Class 1	70%	100%	100%	100%	90%	90%	100%
Class 2	90%	90%	80%	90%	90%	90%	90%
Class 3	90%	100%	90%	100%	100%	100%	100%
Class 4	100%	100%	75%	100%	100%	100%	100%
Class 5	66.67%	83.33%	33.3%	83.3%	66.67%	66.67%	100%
Total	82.5%	95%	80%	95%	85.71%	90%	**97.5%**

Table 5.7: *The best classification result of every applied modeling method on dataset A1*

5.4.3 Dataset A2

Dataset A2 is another additional variant of dataset A in Pomeroy's data. It was proposed to test whether inclusion of a larger number of medulloblastomas (class 1) might lessen the destinations noted in dataset A, other 50 medulloblastoma samples were added. From PCA analysis as shown in Figure 5.3, we can see that a big set of medulloblastomas samples overlap the area of PNETs. This implies that classifiers are likely to produce a high probability of error on class 5.

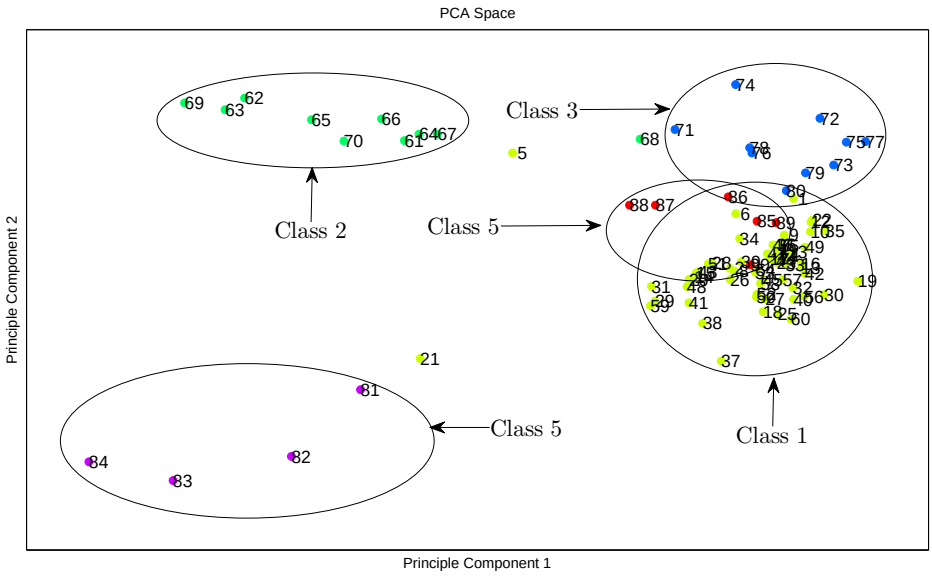


Figure 5.3: *PCA using selected 35 genes to describe the dataset A2.*

For the classification, we apply same classifiers with similar parameters as before. Table 5.8 shows the classification accuracy in % from seven different classifiers. Again the best accuracy (92.22%) is achieved by OVA-WWKNN. The second high accuracy (90%) is produced by the personalised modeling method (kNN), and followed by the accuracy of SVM (88.89%).

Many class 1 samples overlap class 5 samples in the problem space as shown in Figure 5.3. This causes that all algorithms performed extremely poor in the classification experiment of class 5 samples, but all performed well in the classification of class 1. This finding suggests a close relationship between the medulloblastomas and

	MLP	SVM	ECF	kNN	WKNN	Layered WWKNN	OVA WWKNN
	N:7	**linear**	**E:4**	**K:2**	**K:2, θ:0.5**	**K:2, θ:0.8**	**K:6, θ:0.5**
Class 1	96.66%	97%	90%	86.67%	88.33%	90%	98.33%
Class 2	80%	90%	90%	90%	90%	90%	90%
Class 3	70%	100%	100%	90%	90%	100%	90%
Class 4	75%	75%	100%	75%	75%	75%	100%
Class 5	16.67%	0%	0%	0%	0%	16.67%	33.33%
Total	85.56%	88.89%	85.56%	90%	82.22%	85.56%	**92.22%**

Table 5.8: *The best classification result of every applied modeling method on dataset A2*

PNETs. Inclusion of a larger the number of medulloblastomas lessens the classification accuracy in both medulloblastomas (class 1) and PNETs (class 5)

5.4.4 Dataset B

Figure 5.4 shows that classic and desmoplastic classes are well separated. Based on this PCA figure, we expect higher accurate results from the classification of every applied algorithms. The next paragraph summarised the accuracies of classification experiment on dataset B.

Table 5.9 presents the experimental result obtained by applied classifiers on CNS dataset B. The best classification accuracy (97.05%) achieved is from WWKNN model – 33 out of 34 samples are successfully classified. The personalised models (kNN and WKNN), a local model (ECF) and a global model (SVM) outperform the global model – (MLP). In the Pomeroy's work, the best accuracy they achieved is 97.05% using kNN with 400 selected genes based on cross-validation testing, which is the same result from our WWKNN model.

	MLP	SVM	ECF	kNN	WKNN	WWKNN
Parameter	**N:5**	**linear**	**E:4**	**K:3**	**K:3, θ:0.5**	**K:5, θ:0.5**
Class 1	92%	96%	96%	96%	96%	100%
Class 2	88.89%	88.89%	88.89%	88.89%	88.89%	88.89%
Total	91.18%	94.12%	94.12%	94.12%	94.12%	**97.05%**

Table 5.9: *The best classification result of every applied modeling method on dataset B*

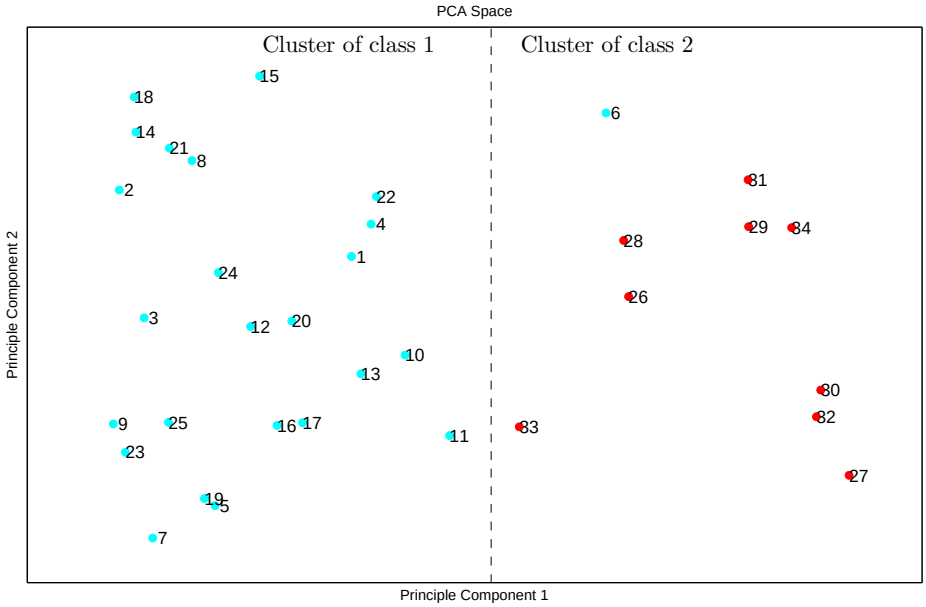


Figure 5.4: *PCA using selected 36 genes to describe the dataset B. The most samples of class 1 and class 2 can be linearly separated.*

From the error sample list, No.33 sample has been predicted as error vector in every classifiers. Sample No.33 of class 2 is located very close to class 1 as shown in Figure 5.4. It is easy to be confused with the samples of class 1 in the classification.

5.4.5 Dataset C

30 Top-ranked Gene Set

The experiment of dataset C is started with the 30 selected genes. Figure 5.5 shows the PCA analysis of dataset C with 30 selected genes. We can see that the samples of class 2 evenly separate in PCA visualisation, which covers a large area in this figure. But the samples of class 1 cluster on left hand side of Figure 5.5. Some of the samples of class 1 are also overlapped by the samples of class 2.

Table 5.10 summarises the best classification accuracies achieved by each applied algorithm with 30 selected gene. It shows that MLP and ECF perform the best accuracy (85%) on 30 selected genes. The lowest accuracy of performance is produced

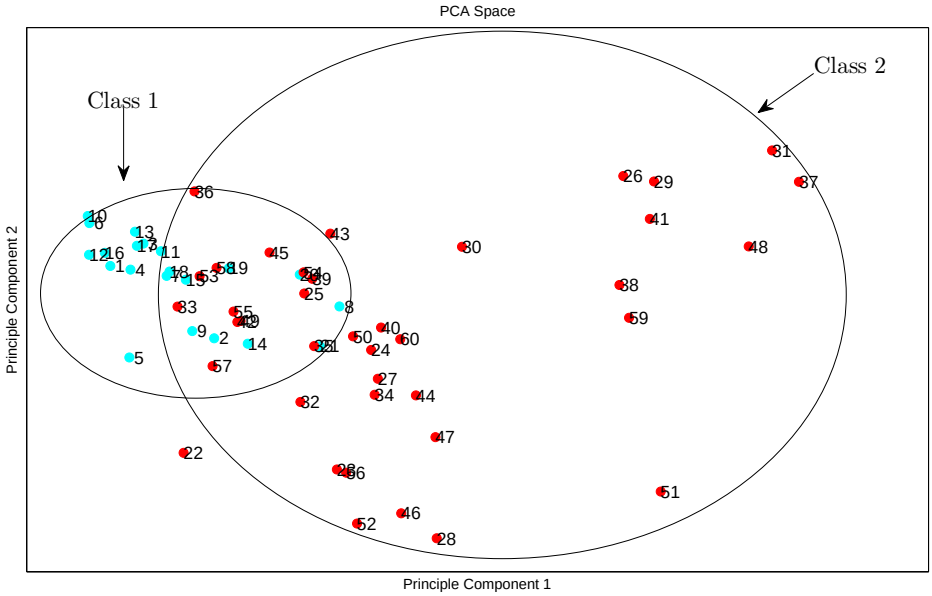


Figure 5.5: *PCA using selected 30 genes to describe the dataset C.*

by WKNN and SVM. However, WWKNN and kNN achieved the accuracy of 81.67% with a balanced classification. The difference between class 1 and class 2 is only 1.1%. This result is also acceptable in terms of clinical problem of disease diagnosis.

	MLP	**SVM**	**ECF**	**kNN**	**WKNN**	**WWKNN**
Parameter	**N:6**	**linear**	**E:4**	**K:5**	**K:3, θ:0.5**	**K:9, θ:0.5**
Class 1	95.24%	71.43%	71.43%	80.95%	95.24%	80.95%
Class 2	79.49%	83.05%	92.31%	82.05%	69.23%	82.05%
Difference	15.75%	11.62%	20.88%	1.1%	26.01%	1.1%
Total	**85%**	78.33%	**85%**	81.67%	78.33%	81.67%

Table 5.10: *The best classification result of every applied modeling method on dataset C with 30 top-ranked genes*

40 Top-ranked Gene Set

Figure 5.6 presents the PCA analysis of dataset C with 40 selected genes. The result is very similar to using 30 selected genes, which has a big overlapped area between

class 1 and class 2. Samples of class 2 cover a larger space in Figure 5.6.

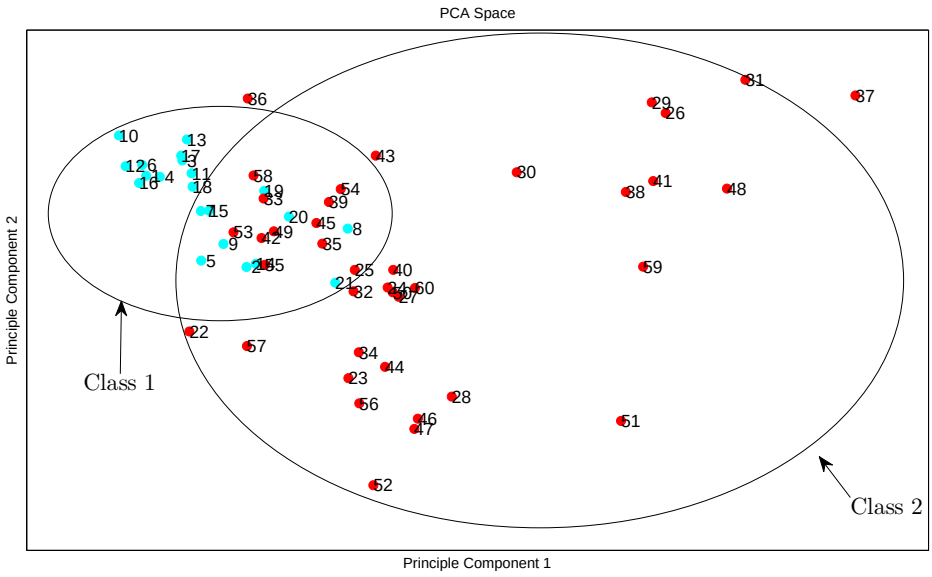


Figure 5.6: *PCA using selected 40 genes to describe the dataset C.*

Table 5.11 records the best accuracy of each algorithm on 40 selected gene. Both of SVM and WWKNN can achieve the best accuracy (86.67%). WWKNN condusts us a more balanced result with 5.86% difference between the accuracy of class 1 and class 2. WKNN performs the lowest accuracy with 40 top-ranked gene. Overall the

	MLP	**SVM**	**ECF**	**kNN**	**WKNN**	**WWKNN**
Parameter	**N:6**	**linear**	**E:4**	**K:5**	**K:3, θ:0.5**	**K:5, θ:0.5**
Class 1	95.24%	76.19%	80.95%	95.24%	95.24%	90.48%
Class 2	79.49%	84.62%	89.74%	74.36%	71.79%	84.62%
Difference	15.75%	8.43%	8.79%	20.88%	23.45%	5.86%
Total	85%	81.67%	**86.67%**	81.67%	80%	**86.67%**

Table 5.11: *The best classification result of every applied modeling method on dataset C with 40 top-ranked genes*

classifiers with 40 genes provide more accurate model than 30 genes.

50 Top-ranked Gene Set

Figure 5.7 presents the PCA analysis of dataset C with 50 selected genes. This figure clearly shows that class 1 is more separable from the class 2 than the classifications with the 30 and 40 selected genes. It indicates a more accurate classification that could be produced with 50 selected genes.

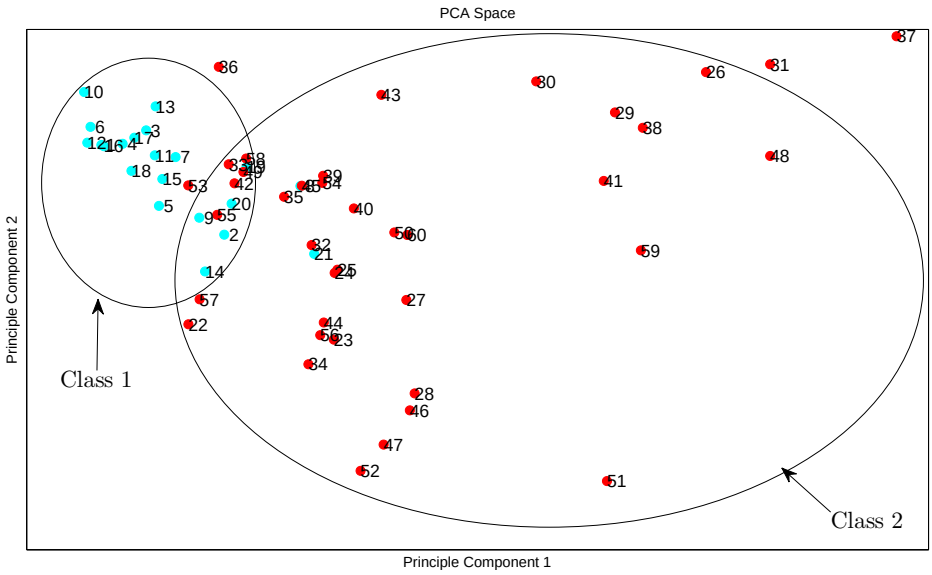


Figure 5.7: *PCA using selected 50 genes to describe the dataset C.*

Table 5.12 presents classification result of every applied modeling method on dataset C with 50 top-ranked genes. The best accuracy is manifested by WWKNN. Its prognostic accuracy is 88.33%. The lowest accuracy again is produced by WKNN. The most balanced result is from MLP with only 1.1% difference between class 1 and class 2. The models that built on 50 top-ranked genes have provided the best accuracy so far.

60 Top-ranked Gene Set

Figure 5.8 presents the PCA analysis of dataset C with 60 selected genes. The classification of PCA is actually the worst in this experiment. The samples of class 1 and the samples of class 2 overlap a bigger area than 30, 40 and 50 selected genes.

	MLP	SVM	ECF	kNN	WKNN	WWKNN
Parameter	**N:6**	**linear**	**E:4**	**K:3**	**K:3, θ:0.5**	**K:9, θ:0.5**
Class 1	80.95%	76.19%	71.43%	95.24%	71.43%	95.24%
Class 2	82.05%	87.74%	92.31%	79.49%	84.62%	84.62%
Difference	1.1%	11.55%	20.44%	15.75%	13.19%	10.62%
Total	81.67%	85%	85%	85%	80%	**88.33%**

Table 5.12: *The best classification result of every applied modeling method on dataset C with 50 top-ranked genes*

It indicates that many noise genes are involved in the selected genes, which would reduce our classification accuracy.

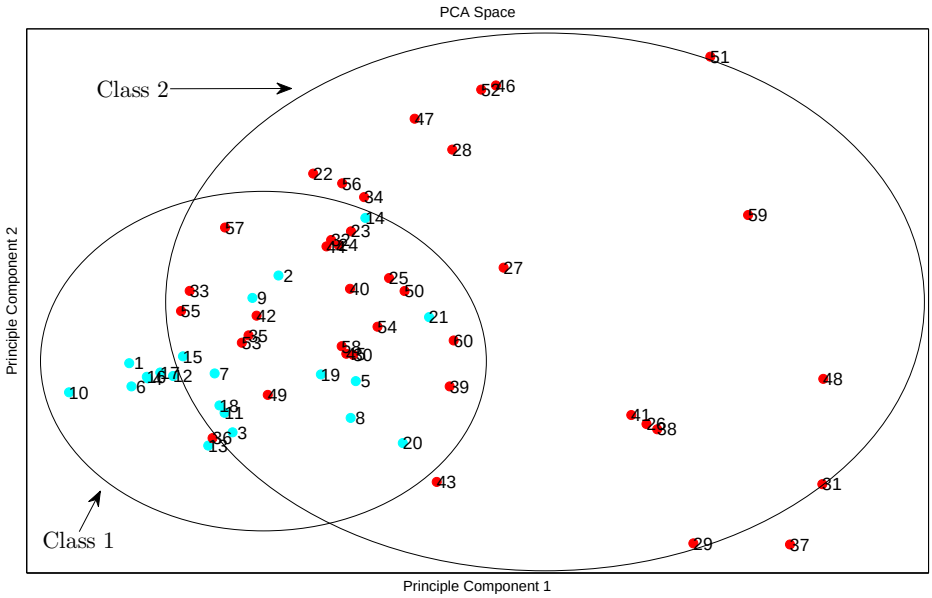


Figure 5.8: *PCA using selected 60 genes to describe the dataset C.*

Table 5.13 shows the classification results of applied modeling methods on 60 selected genes. As show in table, the best classification accuracy (83.33%) is manifested by MLP with 4 hidden neurons. WWKNN performs better result (80%) than other models (SVM, ECF, KNN and WKNN). The most balanced accuracy is produced by ECF with 10.16% difference.

	MLP	**SVM**	**ECF**	**kNN**	**WKNN**	**WWKNN**
Parameter	**N:4**	**linear**	**E:4**	**K:3**	**K:3, θ:0.5**	**K:9, θ:0.5**
Class 1	95.24%	71.43%	66.67%	95.24%	90.49%	95.24%
Class 2	76.92%	82.05%	76.92%	69.23%	71.09%	71.79%
Difference	18.32%	10.62%	10.16%	26.01%	19.40%	23.45%
Total	**83.33**%	78.33%	73.33%	78.33%	78.33%	80%

Table 5.13: *The best classification result of every applied modeling method on dataset C with 60 top-ranked genes*

Summary on Classification of Dataset C

Above results have elucidated that the most accurate result is produced with 50 top-ranked genes using WWKNN (88.33%) in terms of classification on medulloblastoma treatment outcome. The lowest accurate results is produced using 60 top-ranked genes. Comparing the Pomeroy's work (78.33%) using a kNN algorithm with 8 genes, our classification accuracy obtained by WWKNN (88.33%) is much better. The models that built on 50 top-ranked genes will be considered in the knowledge discovery. The knowledge discovery of this dataset will be described in next chapter.

5.5 Comparison of Two Approaches of WWKNN on Multi-class Classification

We have applied two approaches of WWKNN to solve multi-class problem on Pomeroy's data. One is multilayered threshold (layered-WWKNN), the other one is the "one-vs.-all" (OVA-WWKNN). Interestingly, OVA-WWKNN outperforms in every classification problem (i.e. dataset A, A1 and A2). In this section, we compare both layered-WWKNN and the OVA-WWKNN with different parameter on three multi-class problems.

K denotes the selected nearest neighbour in personalised models. (see Chapter 3). We now define K as an integral number between 2 and 11. To evaluate each solution, we have set up an invariable threshold 0.5 for both approaches.

100 top ranked genes are used for this classification. The result obtained for dataset A, A1 and A2 are recorded in Table 5.14.

Dataset A	**K:2**	**K:3**	**K:4**	**K:5**	**K:6**	**K:7**	**K:8**
Layered	90.48%	76.19%	71.43%	73.81%	69.05%	54.76%	52.38%
OVA	90.48%	90.48%	92.86%	92.86%	97.62%	95.42%	97.62%
	K:9	**K:10**	**K:11**				
Layered	40.48%	35.71%	28.57%				
OVA	97.62%	97.62%	97.62%				
Dataset A1	**K:2**	**K:3**	**K:4**	**K:5**	**K:6**	**K:7**	**K:8**
Layered	90.00%	77.50%	67.50%	67.50%	67.50%	57.50%	52.50%
OVA	90.00%	90.00%	90.00%	92.50%	97.50%	95.00%	95.00%
	K:9	**K:10**	**K:11**				
Layered	42.50%	37.50%	32.50%				
OVA	97.50%	97.50%	97.50%				
Dataset A2	**K:2**	**K:3**	**K:4**	**K:5**	**K:6**	**K:7**	**K:8**
Layered	85.56%	81.11%	75.56%	74.44%	70.00%	68.89%	68.89%
OVA	85.56%	90.00%	91.11%	92.22%	92.22%	91.11%	91.11%
	K:9	**K:10**	**K:11**				
Layered	72.22%	75.56%	72.22%				
OVA	91.11%	92.22%	92.22%				

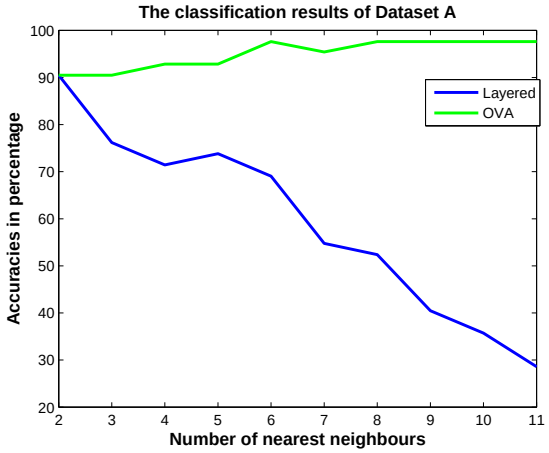
Table 5.14: *Results comparison of two approaches*

Based on these results, we find that layered-WWKNN usually achieve better accuracy with small number of neighbours (K). OVA-WWKNN obtains evenly high accuracy across every neighbour. For a better visualisation, we demonstrate accuracy changes in 2D linear diagrams as shown in Figure 6.3.

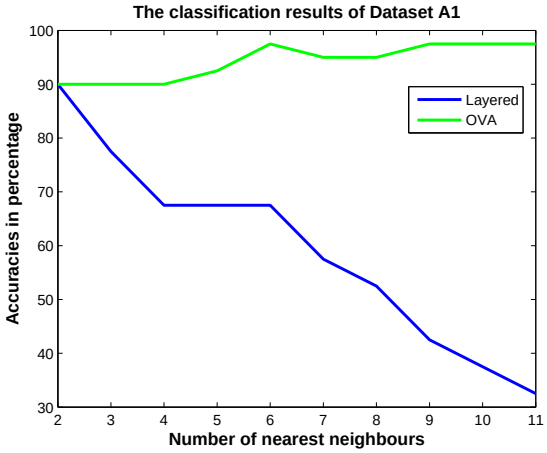
Notice that both approaches are applied WWKNN as the main classification function. Only difference between two approaches is that OVA firstly transformed a multi-class problem to several two-class problems. In WWKNN approach, the output is calculated by Equation 5.1. In order to finally classify the queried vector x_i into one of classes, there has to be a probability threshold P_{thr} selected (Kasabov, 2007). The predicted output of query vector is based on the comparison between the output and probability threshold.

$$y_i = \sum \left((1 - d_j) \times y_j \right) / \sum (1 - d_j) \tag{5.1}$$

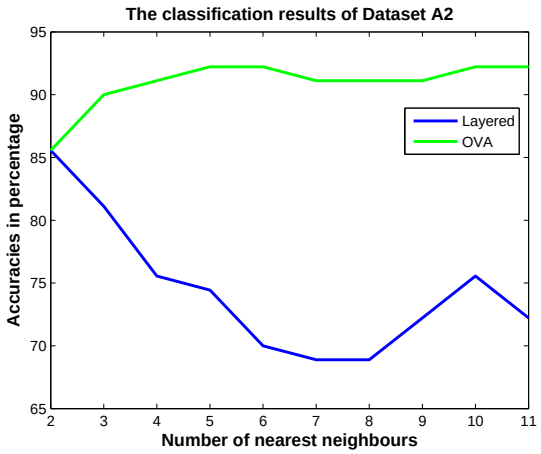
where y_i denotes the output of class label for new vector, d_j is the measured distance from the inputted vector to nearest neighbours. The y_j is representing the class



(a) Dataset A



(b) Dataset A1



(c) Dataset A2

Figure 5.9: *Two multi-class classification of WWKNN on dataset A, A1 and A2*

labels of measured nearest neighbours.

Unlike to machine learning algorithms (Alpaydim, 2004), WWKNN is not able to learn the difference between the classes from existed data. WWKNN is a statistic analysing algorithm, which classifies the samples by calculating the numbers. To solve two-class problem, y_j is only represent two numbers: 1 or 2, which is quite simple for the classification since there are only two options. In multi-class classification, the problem is much more complex. More different classes are involved in training data that would increase the errors of classification. In Pomeroy's data, three classes of CNS tumour (class 1, class 2 and class 3) are quite close to each other, which actually increase more error probability.

Overall, the layered-WWKNN has a significant weakness in multi-class classification. The algorithm could be easily confused by more than two class labels. However the OVA approach simplifies the multi-class problem to several two-class problems with class label 1 and 2, which overcomes this problem.

5.6 Summary of Modeling Experiment

This experiment shows that the personalised modeling method outperforms other global and local modeling methods in terms of classification accuracy on Pomeroy's data. WWKNN is the best algorithm to implement the modeling in every datasets. For the multi-class problems, we have developed two approaches of WWKNN classification (multilayered threshold and OVA). With multilayered threshold WWKNN, the better classification accuracy only occurs with a small number of K such as 2. This leads to a weakness of layered-WWKNN. But the OVA approach overcomes this weakness in the classification.

However, this chapter only presents one of our experimental purpose. The next chapter will focus on knowledge discovery and ontology representation to do further investigation.

Chapter 6

Ontology-based Modeling and Knowledge Discovery Illustrated on the Case Study Problem

The last chapter has offered a comparative study of major global, local and personalised models, including MLP, SVM, ECF, kNN, WKNN and WWKNN on benchmarked Pomeroy's CNS cancer data. The personalised model, WWKNN performs the best classification accuracy. This result implies the potential of personalised diagnostics and treatment for clinical decision-making.

As mentioned in Chapter 5, we have two experimental purposes. One is the comparative study of global, local and personalised modeling. This chapter focuses on the second purpose that is to discover the reusable knowledge for cancer diagnosis and prognosis based on the WWKNN results. The ontology based knowledge system will be used for analysis in this chapter. Section 6.1 describes how we use WWKNN and the CNS ontology to discover the important cancer related information. In Section 6.2, we implement the knowledge discovery process on dataset A (multi-class CNS tumours). We then present the knowledge discovery on two subclasses of medulloblastomas in Section 6.3. Section 6.4 presents the knowledge discovery in terms of medulloblastoma treatment outcome, and gene reaction after treatment.

6.1 Knowledge Discovery Method

In this thesis, our interest is to discover the discriminant genes that are able to represent one category of samples in the entire problem space. We design our knowledge discovery process as following:

1. Capture and record the personalised important genes for each sample.
2. Import these recorded genes into related ontology class.
3. Create ontology based relations between the individual samples and their personalised important genes.
4. Apply the ontology query tool to define the most discriminative genes in one experimental class.
5. Use statistical analysis approach to evaluate the discriminant genes.

This knowledge discovery process will be contributed by two implementations including WWKNN and CNS ontology. The WWKNN modeling method is used to rank the importance of experimental genes for each single sample. CNS ontology is applied to extract the discriminant genes for each experimental class. Notice that we also apply several different data analysis approaches to evaluate the captured genes in this thesis. The following subsections separately describe the details of WWKNN and CNS ontology in knowledge discovery.

6.1.1 Knowledge Discovery with WWKNN Modeling

We have described the weighted-weighted K nearest neighbour (WWKNN) model (Kasabov, 2007). In the our experiment, WWKNN achieved the best classification accuracy on each dataset of Pomeroy's data. In order to explore the important knowledge, WWKNN is also capable of discovering certain important information and knowledge specialised for the individual query sample. An example for personalised data sample from dataset B analysis is demonstrated in Table 6.1.

Sample:	**1**	**Sample:**	**27**
Gene ID	Importance	Gene ID	Importance
G6815	1.00	G4941	1.00
G5275	0.82	**G6815**	0.90
G4247	0.75	**G4423**	0.88
G4423	0.70	**G5275**	0.67
G226	0.61	G5328	0.65
G5957	0.54	G4463	0.60

Table 6.1: *An example for personalised data sample analysis using WWKNN on Pomeroy's dataset B with 6 genes.*

Table 6.1 shows that the importance of each gene for cancer data sample 1 and 27 of CNS dataset B is significantly different. The top-ranked gene of sample 1, G6815 is the second important gene for sample 27 in terms of classification performance.

In this thesis, we record top ten ranked genes for each sample based on their personalised gene ranking. Then we import these recorded genes into class "Gene_ bank" of the CNS ontology. Notice that all genes are recorded as their accession number. In the ontology knowledge environment, every individual instance must be unique. For these genes that have more than one related sample, the CNS ontology only record them once, but each gene has multi ontology-based relations with its different related samples. The next subsection explains how we use CNS ontology to extract and export the discriminant genes based on the individual samples.

6.1.2 Knowledge Discovery with the CNS Ontology

The developed CNS ontology provides conceptual links between samples, CNS diseases and personalised information. All the information of each sample in the CNS ontology are traceable through an ontology-based query tool. Once the personalised important genes on each sample are imported into CNS ontology, we create a new relation that is called "has_important_gene_of" between the class "Samples" and class "Gene_banks". This relation is proposed to connect personalised important genes and each particular sample in this ontology. When the sample and important genes are completely connected, CNS ontology will allow us to easily identify and export the most discriminative genes for each experimental class in one problem space (a knowledge domain), see Figure 6.1.

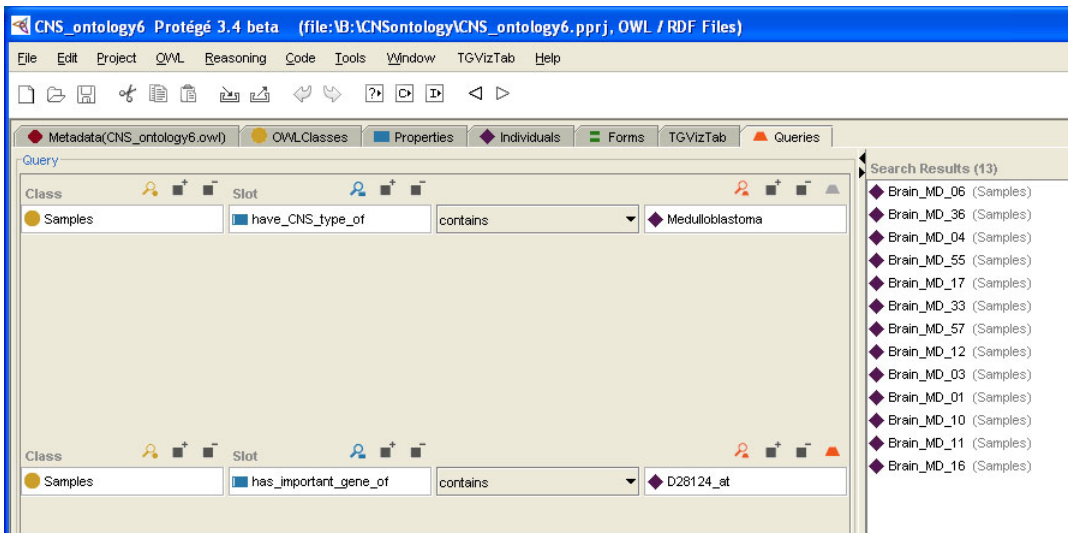


Figure 6.1: *Snapshot of CNS ontology detail showing query research system looking for medulloblastoma samples that closely correlate to gene D20124 (gene accession No.) as an example.*

The following sections of this chapter describe the process of knowledge discovery by the combination of WWKNN and the CNS ontology, and the detailed information of the discovered discriminant genes. Based on the problem categories of experimental datasets, the discriminant genes discovery focuses on three main knowledge domains including multiple CNS tumour class, medulloblastoma morphology and medulloblastoma treatment outcomes.

6.2 Knowledge Discovery in Multi-class Problem

We firstly focus on the discriminant genes discovery for different CNS tumour class. This knowledge discovery is based on the original dataset A that has five different classes. Class 1, class 2, class 3 and class 5 represents embryonal CNS tumours medulloblastomas, malignant gliomas, primitive neuroectodermal tumours, and atypical teratoid / rhabdoid tumours with sample size 10, 10,10 and 8. Class 4 represents 4 samples that have normal human cerebella. In the classification experiment, we have selected 35 top ranked genes by using OVA SNR. Normal brain is easily classified with the 100% accuracy (see chapter 5), since gene expression values of normal human cerebella are extremely difference to tumour classes. In this thesis, we are in-

terested in the discovery of discriminant genes on four different CNS tumour classes, since a mistake on the difference cancer diagnosis will increase the risk of patients.

Based on the personalised gene ranking analysis of WWKNN, every sample has personalised ranks on 35 experimental genes. To capture the common knowledge across the samples in same class, we firstly select 10 top ranked genes from each sample to import into CNS ontology. Then based on CNS ontology knowledge framework, we extract three genes for each CNS tumour class. These genes are described in the following related subsections.

To evaluate each discriminant gene, we have applied the statistical analysis approach boxplot. The statistics toolbox of MATLAB is used to carry out the analysis. For some cases that are not able be evaluated by boxplot, we have employed a Two-Sample t-test. For further explanation of t-test see (Wild and Seber, 2000). The two dimensional and three dimensional diagrams are used for visualising of discriminant genes.

6.2.1 Analysis on Medulloblastomas

The experimental result on dataset A indicates that the samples of medulloblastomas can be clearly separated from other tumour samples. Based on the discovered personalised important genes, we obtain three top ranked genes from samples of medulloblastoma. These genes are recorded as their accession numbers: M93119, X06617 and U05012_s in the CNS ontology. Table 6.2 presents the details of these three genes.

Gene No.	Accession No.	Description
G2365	M93119	INSM1 Insulinoma-associated 1 (symbol provisional)
G4092	X06617	RPS11 Ribosomal protein S11
G6435	U05012_ s	NTRK3 Neurotrophic tyrosine kinase, receptor, type 3 (TrkC)

Table 6.2: *Three defined discriminant genes of Medulloblastomas.*

Figure 6.2 presents expression value variances of three discriminant genes cross 42 experimental samples. Notice that sample no. 1-10 represents the samples of medulloblastoma. In this figure, the gene M93119 and X06617 present more variable gene

expression value on medulloblastoma samples. But the samples perform very small variance on gene U05012_s. To evaluate these discriminant genes, we apply the boxplot.

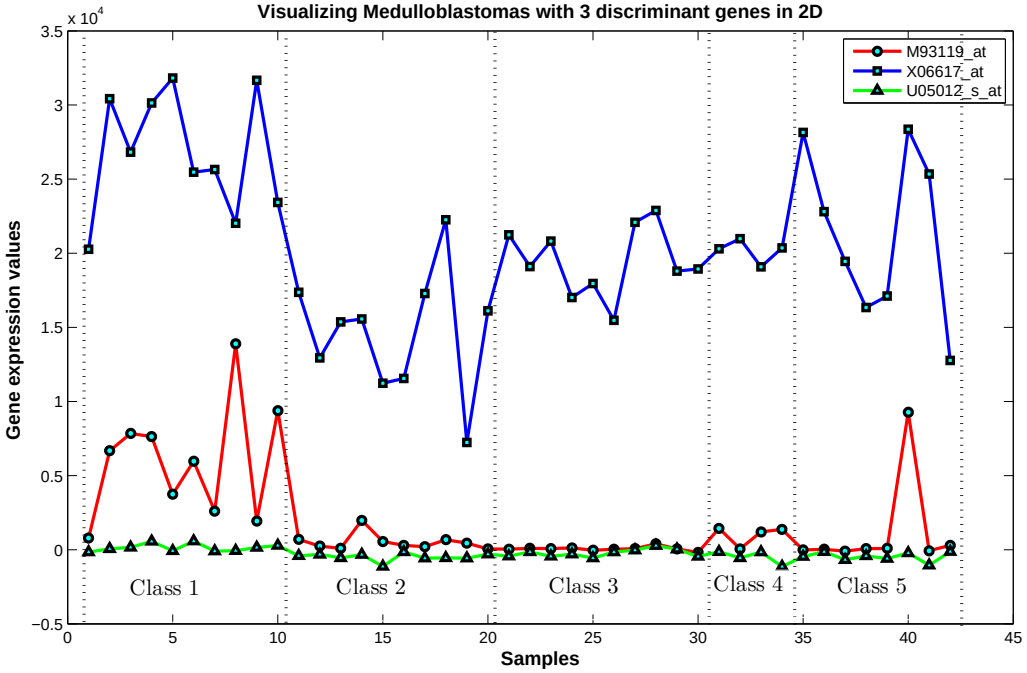
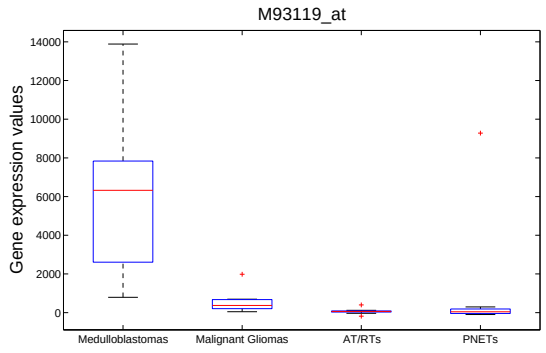
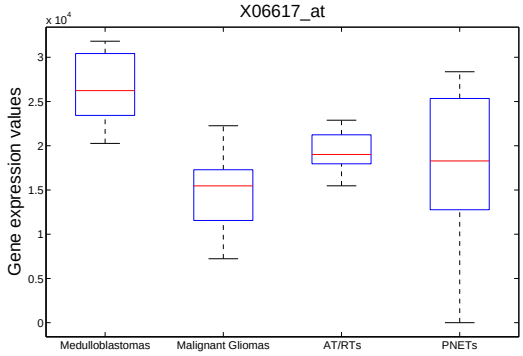


Figure 6.2: *Visualising medulloblastomas with 3 discriminant genes in 2D. Sample no. 1-10 represents the samples who has medulloblastoma. M93119 indicates the most variant values in samples of class 1.*

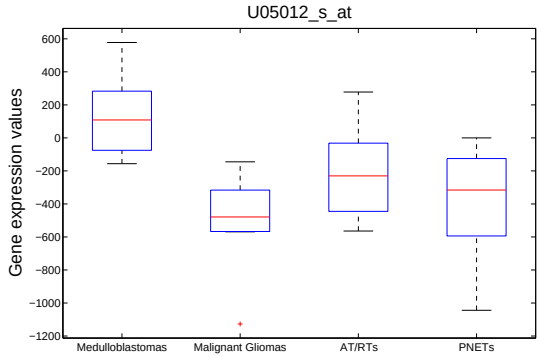
The boxplots of genes M93119, X06617 and U05012_s for each class are shown in Figure 6.3. It gives a better visualisation of medulloblastomas related discriminant genes in gene expression values. In this figure, the boxplots on the right hand side represent the samples of medulloblastomas, and each of subfigure represents one discriminant gene. We can see that the median line (median) of the medulloblastoma boxes are higher than other three boxes (other classes of CNS tumours) in each discriminant gene. The longer boxplot on gene M93119 suggests that samples of medulloblastoma have wider expression value distribution. We also identify that the medulloblastomas can be clearly discriminated in the boxplot of gene U05012_s. These distinctive difference between the medulloblastomas and other CNS samples in the three discriminant genes suggests a strong evidence that most of medulloblastomas could be separated based on these three discriminant genes.



(a) M93119



(b) X06617



(c) U05012_ s

Figure 6.3: *Boxplots of three discriminant genes in Medulloblastomas*

Figure 6.4 shows a 3D visualisation of multi CNS class samples with three medulloblastoma based on discriminant genes. In this figure, the samples of medulloblas-

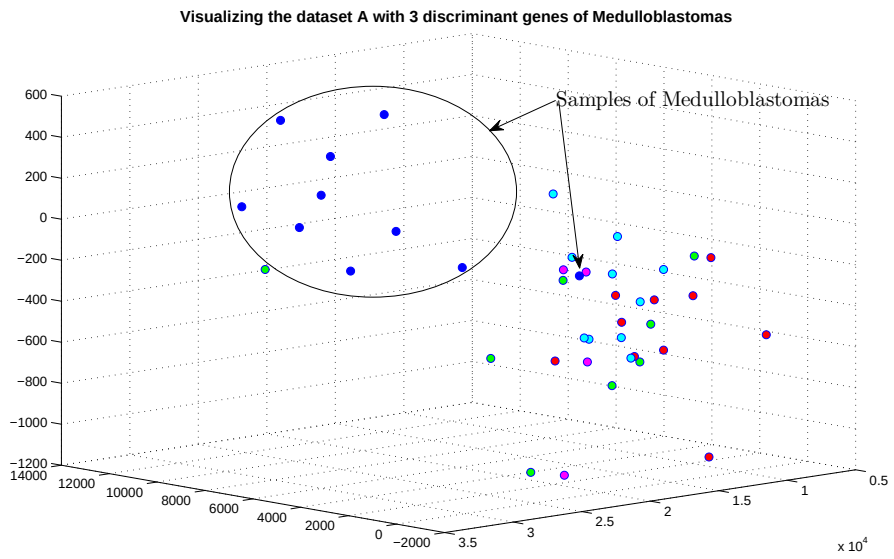


Figure 6.4: *Visualising medulloblastomas with 3 discriminant genes.*

toma is indicated highly separable that 9 out 10 samples can be visually separated.

6.2.2 Analysis on Malignant Gliomas

Malignant gliomas are relatively common primary brain tumours, deriving from cells of non-neuronal origin, which is clearly separable from other three neuronal tumour classes in CNS (Gromeier and Wimmer, 2001). The accurate classification from our experiment also supports that malignant gliomas is highly separable tumour class in the CNS. Based on the personalised importance of genes produced by WWKNN, CNS ontology extracts gene X86693, U45955 and Z31560_s as the discriminant genes for malignant gliomas. Table 6.3 presents the details of these three genes.

Gene No.	Accession No.	Description
G4741	X86693	High endothelial venule
G3239	U45955	Neuronal membrane glycoprotein M6b mRNA, partial cds
G6035	Z31560_ s	SOX2 SRY (sex determining region Y)-box 2

Table 6.3: *Three defined discriminant genes of malignant gliomas.*

Figure 6.5 presents gene expression values of these genes across the 42 samples in dataset A. In this figure, sample no.11-20 represents the samples that are in malignant glioma class. The samples of malignant glioma present significantly high gene expression value on these three discriminant genes. They are also evaluated by using the boxplots as shown in Figure 6.6.

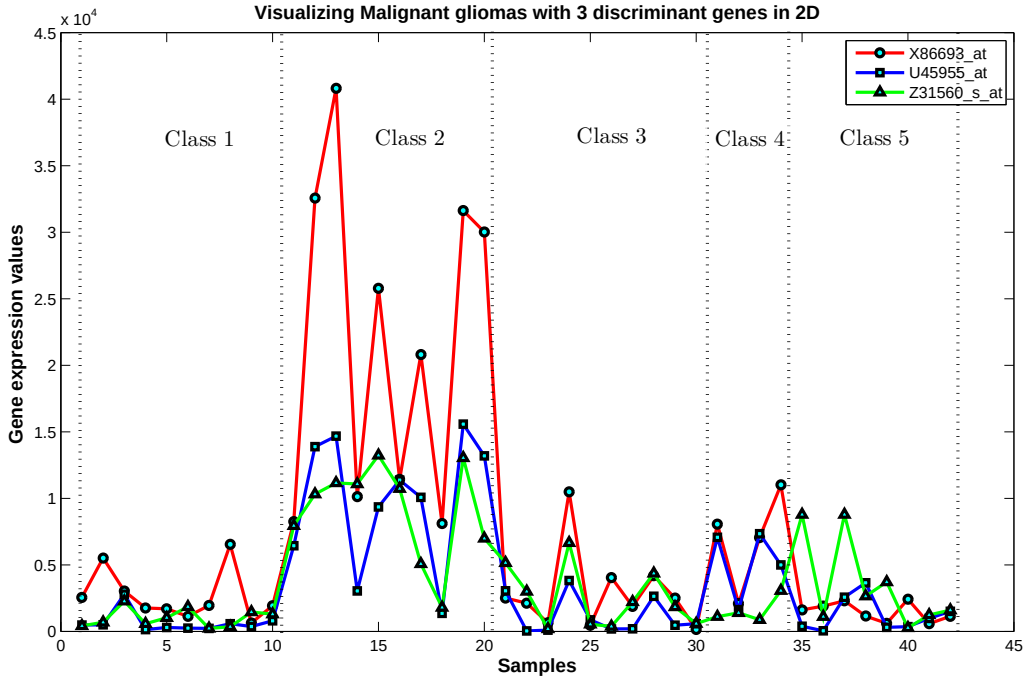
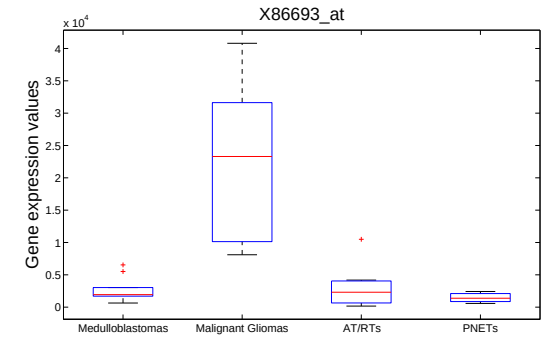


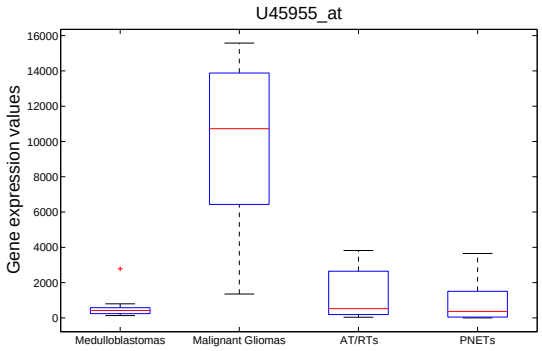
Figure 6.5: *Visualising malignant gliomas with 3 discriminant genes in 2D*

In the figure, the left second box of each subfigure represents the samples of malignant gliomas. The median lines of malignant gliomas are much higher than other classes. In gene X86693 and U45955, the malignant glioma boxes have longer interquartile range (the distance between the top and bottom of the box) that implies a widely separated gene expression value distribution on analysed discriminant genes. The findings from boxplots suggest that samples of malignant glioma have significant high and more variable gene expression value on gene X86693, U45955 and Z31560_s.

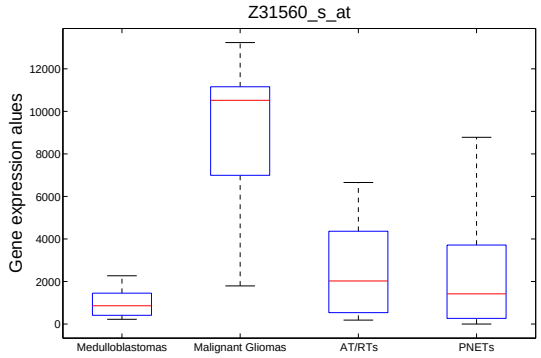
Figure 6.7 shows a 3D visualisation of multi CNS class samples with three malignant gliomas related discriminant genes. The samples of malignant gliomas are represented as red circles, which can be visually separated from other tumours. The most of samples from other classes cluster together in this space. This implies that the



(a) X86693



(b) U45955



(c) Z31560

Figure 6.6: *Boxplots of three discriminant genes in Malignant gliomas. Gene X86693 and U45955, the malignant glioma boxes have longer interquartile range (the distance between the top and bottom of the box) that implies a widely separated value distribution of gene expression on analysed discriminant genes.*

samples of other CNS classes presents similar expression values on gene X86693, U45955 and Z31560_s.

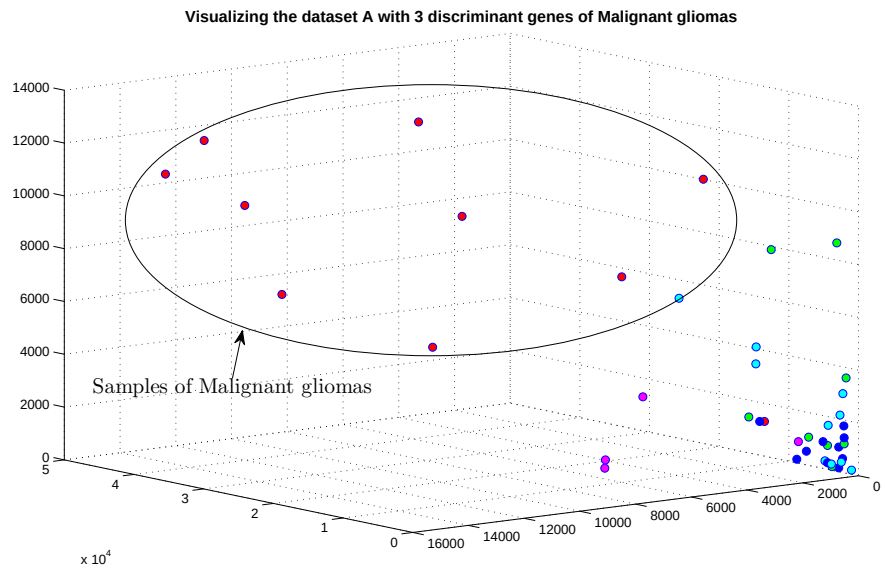


Figure 6.7: *Visualising malignant gliomas with 3 discriminant genes in 3D*

6.2.3 Analysis on Atypical Teratoid/Rhabdoid Tumours (AT/RTs)

We next analyse the AT/RT tumours. It is one of the most common childhood solid tumours (Lefkowitz et al., 1988). An atypical tertoid rhaboid tumour can be mistaken for a medullonlastoma and primitive neuroectodermal tumour, since the current prognosis is significantly poor in clinical knowledge. However our experimental results suggest that AT/RTs are clearly separable. All global local and personalised models perform very accurate classification on AT/RTs.

Based on the personalised gene ranking scores in CNS ontology, we also define three most discriminative genes in AT/RTs. These genes are recorded as their accession numbers: D83735, L38969 and D83174_s. Table 6.4 presents the details of these three genes.

Gene No.	Accession No.	Description
G618	D83735	Adult heart mRNA for neutral calponin
G1553	L38969	Thrombospondin 3 (THBS3) gene
G6724	D83174_ s	CBP1 Collagen-binding protein 1

Table 6.4: *Three defined discriminant genes of AT/RTs.*

Figure 6.8 presents gene expression value variance of these discriminant genes across

the samples of dataset A. The samples of AT/RTs are presented in between sample 21 and 30 in the figure. The gene expression values of D83174_s and D83735 have significantly variable values on the AT/RT patients in the Pomeroy's dataset A. Comparing these two discriminant genes, variance of L38969 is not very discriminative on samples of AT/RT.

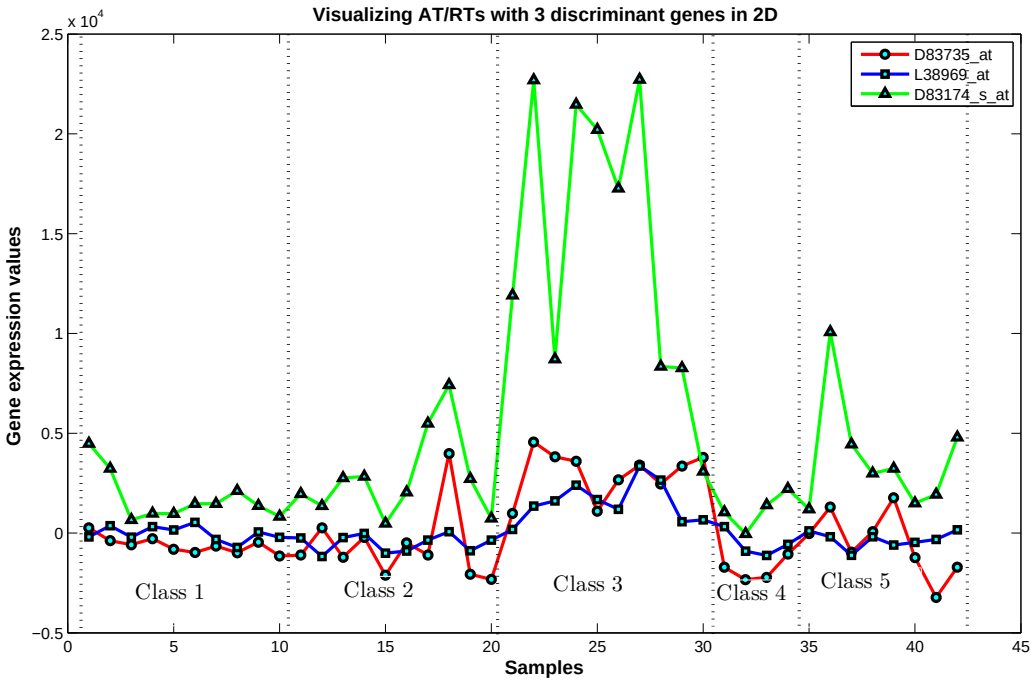
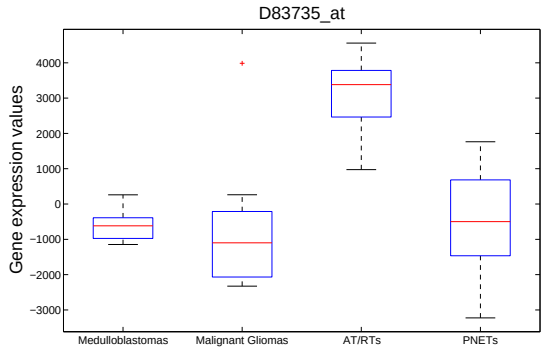
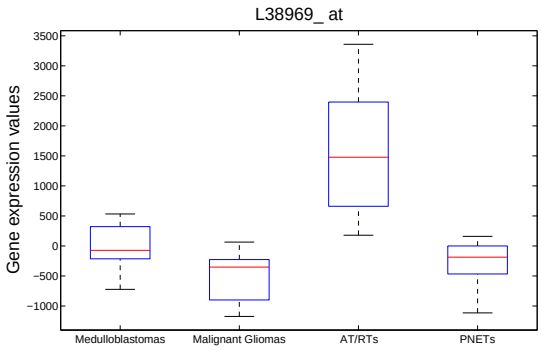


Figure 6.8: *Visualising atypical teratoid/rhabdoid tumours with 3 discriminant genes in 2D*

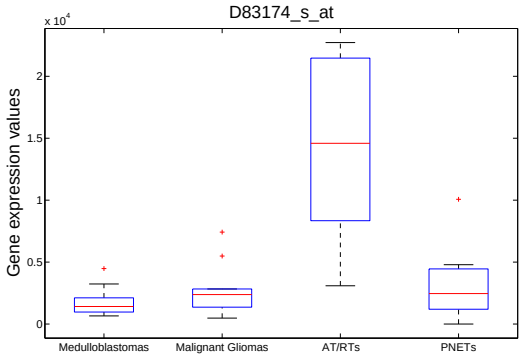
For the statistical analysis, we use boxplot as shown in Figure 6.9. The third box (from the left hand side) in each subfigure present the gene expression values of AT/RTs on each discriminant gene. In general, the AT/RTs appear higher median lines and separable value distribution on gene expressions. This suggests a close relationship between three discriminant genes and AT/RT tumour class. Three discriminant genes are able to represent the main gene mutations in terms of AT/RT tumour.



(a) D83735



(b) L38969



(c) D83174

Figure 6.9: *Boxplots of three discriminant genes in Atypical teratoid/rhabdoid tumoursthe. AT/RTs appear higher median line and separable distribution of gene expression values.*

The 3D visualisation of samples from dataset A is shown in Figure 6.10. All ten samples of AT/RTs are indicated visually separable in this space.

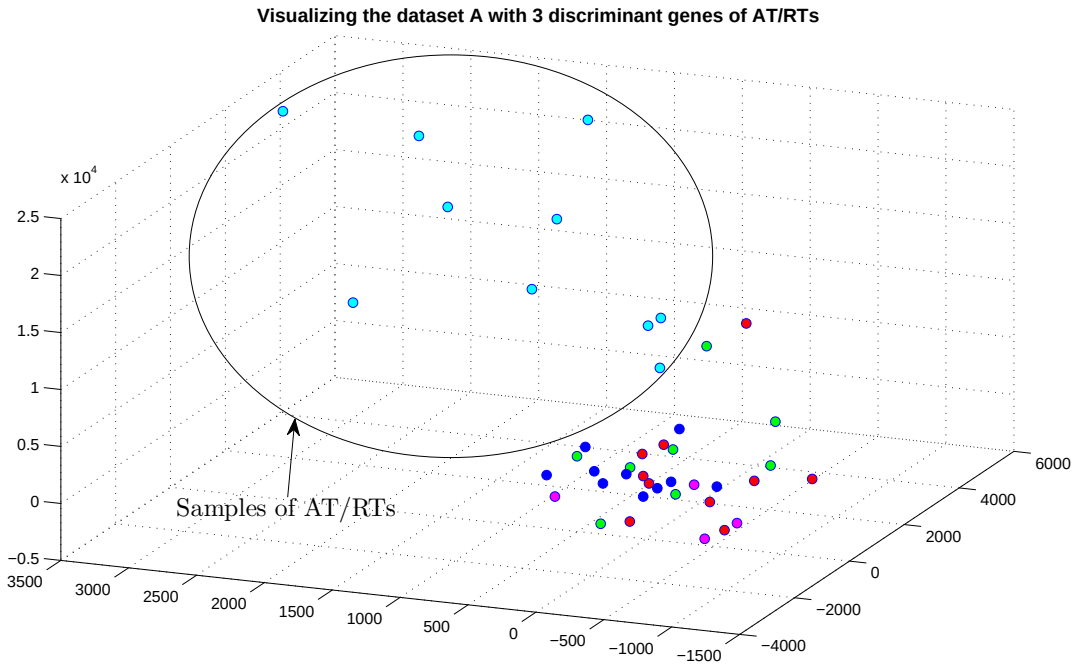


Figure 6.10: *Visualising atypical teratoid/rhabdoid tumours with 3 discriminant genes in 3D*

6.2.4 Analysis on Primitive Neuroectodermal Tumours

Primitive neuroectodermal tumour (PNET) is a rare tumour, which is now known as ewing family tumours including medulloblastomas. In the experimental dataset A, PNET tumour class has 8 samples, including 6 infratentorial PNET and 2 supratentorial PNET. Both PNETs are brain tumour occurs in different area of brain (see Chapter 5). Considering the difference between two PNETs, this thesis also analysed dataset A1 that excludes these 2 pineoblastomas. However, the classification result indicates that these 2 pineoblastomas do not impact accuracy of classification on infratentortial PNET. Since both two types of PNET arise in CNS, we still use dataset A to extract the discriminant genes for PNETs. CNS ontology conducted us to obtain three discriminant genes of PNETs. Table 6.5 presents the details of these three genes.

Figure 6.11 shows 2D visualisation of these three genes across the samples in dataset A. The samples of PNETs are represented as samples no. 35-42 in Figure 6.11. The

Gene No.	Accession No.	Description
G6368	M80397_ s_ at	POLD1 Polymerase (DNA directed), delta 1, catalytic subunit (125kD)
G982	HG4178-HT4448_ at	Af-17
G4152	X14830_ at	CHRNB1 Cholinergic receptor, nicotinic, beta polypeptide 1 (muscle)

Table 6.5: *Three defined discriminant genes of PNET.*

figure indicates that the variable value of discriminant genes is only focused on few samples of PNETs. We can see that gene HG4178-HT4448 just has two samples with significant high gene expression. This is also appeared in gene X14830. The other samples of PNETs are not clearly separable in this figure. For better visualisation and analysis, we also apply a boxplot technique.

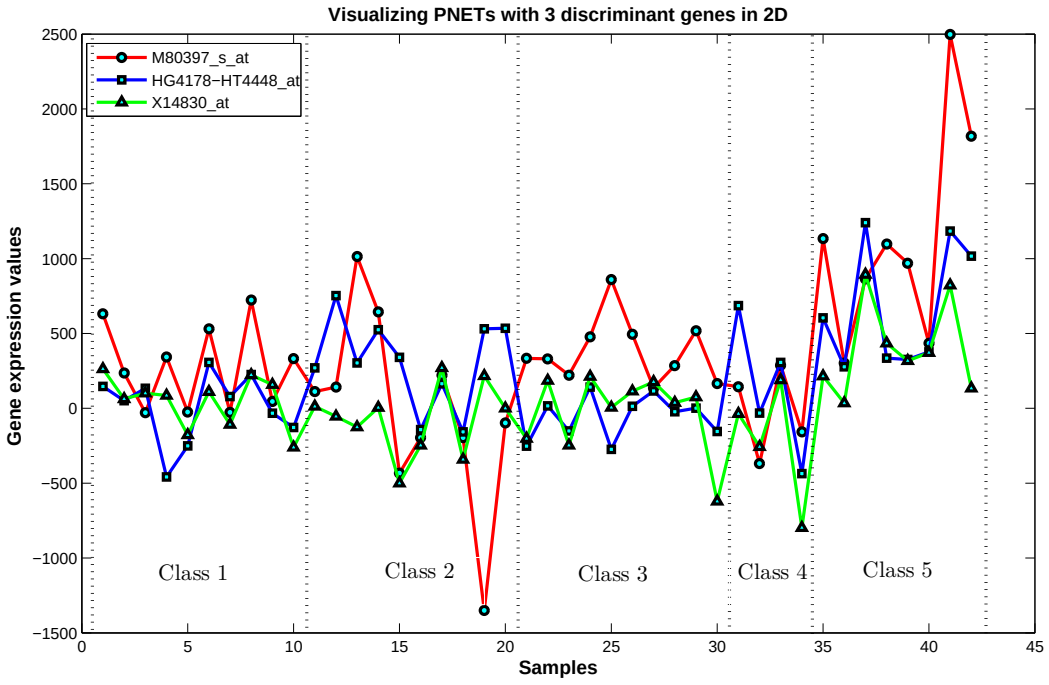
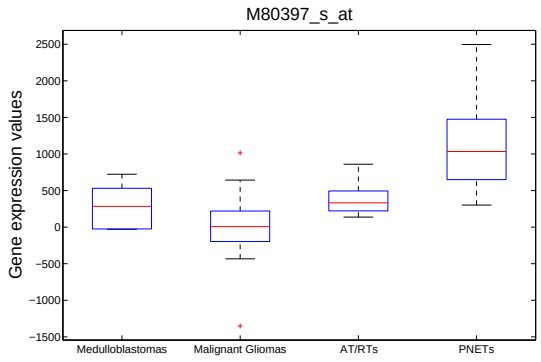


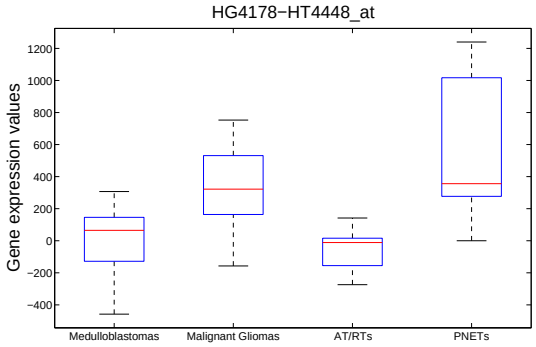
Figure 6.11: *Visualising Primitive neuroectodermal tumours with 3 discriminant genes in 2D*

Figure 6.12 describes these three discriminant genes in boxplot. Notice that the fourth boxes of subfigures represents the samples of PNET, and each of subfigure represent one of the selected discriminant gene. In the figure, gene M80397_s indicates

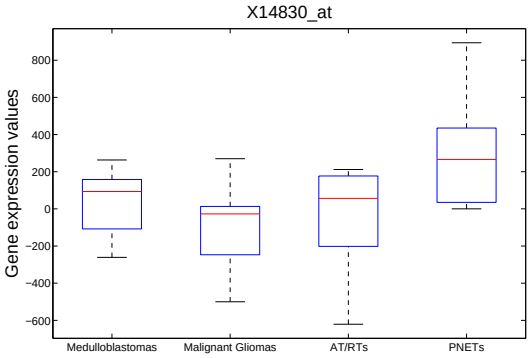
higher median line with wider gene expression value distribution. HG4178-HT4448 indicates that the PNETs and malignant glimas have very similar median of boxplots. The gene expression value distribution of PNET samples and AT/RT samples are performed similar on gene X14830. These findings suggest a further analyse on the gene HG4178-HT4448 and X14830. To further analyse these two genes we conduct a Two sample t-test. The null hypothesis is that the data in two classes is independent



(a) M80397



(b) HG4178-HT4448



(c) X14830

Figure 6.12: *Boxplots of three discriminant genes in Primitive neuroectodermal tumours. Means of class PNETs and means of class malignant glimas with gene HG4178-HT4448. The samples of PNETs and samples of AT/RTs has a significant overlapped gene expression values with gene X14830.*

random samples from normal distributions with equal means but unknown variances, against the alternative that the means are not equal. This analysis is carried out by using statistical toolbox from MATLAB. The p-value for each comparison of two sample t-test are listed in Table 6.6.

Two Sample t-test is conducted the small p-values (e.g. less than 0.05) of every test.

Gene Accession No.		**P-value**
X14830_ at	PNET - MD	0.0096
	PNET - MGlio	0.0171
	PNET - AT/RT	5.16E-05
HG4178-HT4448	PNET - MD	5.20E-04
	PNET - MGlio	0.0474
	PNET - AT/RT	8.26E-05

Table 6.6: *Results of two samples t-test from MATLAB*

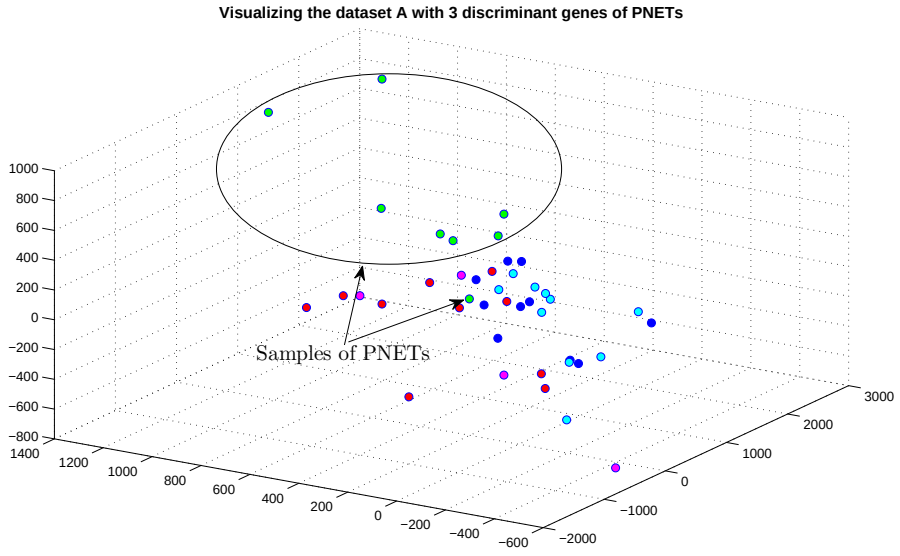


Figure 6.13: *Visualising Primitive neuroectodermal tumours with 3 discriminant genes in 3D*

This suggests a strong evidence that PNETs and the other classes have different means and value distributions. Based on this result, we could prove that PNETs have higher means than samples of other CNS classes in this dataset. But these results has no evidence to support the samples of PNETs are clearly separable from others in gene X14830. In this thesis, we cannot solve this problem, since sample size of experimental data is too small. However, the clear separation on gene M80397 and wide distribution of gene expression value on gene HG4178-HT4448 are different enough to discriminate the samples of PNET in this dataset. Figure 6.13 indicates a 3D visualisation of multi CNS class samples with PNETs based discriminant genes. 7 of 8 PNETs are visually separable.

6.3 Knowledge on Principal Histological Subclass of Medulloblastomas

Pomeroy et al. (2002) have defined that two subtypes of medulloblastoma (desmoplastic and classic) are clearly distinguishable by gene expression. They also suggested that SHH dysregulation and PTCH (Versteege, 1998) highly correlated to the pathogenesis of medulloblastoma. For further information please see (Pomeroy et al, 2002). In our experiment, samples were classified with high accuracy (above 90%) by using every applied algorithm. The best classification accuracy (97.3%) achieved on dataset B is from personalised WWKNN model, which is the same result as Pomeroy's work.

To discover new knowledge, we use the same method as we have used in multi tumour problems. Firstly we import the personalised top 10 important genes into CNS ontology. These genes are ranked by WWKNN. Then we use ontology-based query tool to extract the discriminant genes for each experimental class. Three different genes are extracted for each experimental class based on their personalised importance in terms of medulloblastoma subclasses.

To evaluate our finding, we compare of means and standard deviation of two samples with different discriminant genes. The statistic toolbox of MATLAB is used to carry out the calculation and comparison. We cannot use boxplot or other similar statistical methods to compare and analyse the expression values of discriminant genes, since the samples size of these two classes (25 vs. 9) is distinctive difference.

6.3.1 Discriminant Gene Discovery on Classic Medulloblastomas

The discriminant genes of classic medulloblastoma are recorded as their accession number: HG1980-HT2023, U63842 and X67951. Table 6.7 presents the detail of these genes.

Figure 6.14 indicates the variance of three discriminant genes across the 34 samples in dataset B. The dashed line is used to separate the samples of classic and desmplastic of medulloblastomas. We can see that gene expression value variance of the

Gene No.	Accession No.	Descriptions
G6818	HG1980-HT2023	Tubulin, Beta 2
G3541	U63842	Neurogenic basic-helix-loop-helix protein (neuroD3) gene
G4466	X67951	PAGA Proliferation-associated gene A

Table 6.7: *Discriminant genes for subclass of medulloblastomas*

discriminant genes perform quite different in two different classes. Gene HG1980-HT2023, U63842 and X67951 perform more variable across the samples of classic medulloblastomas. This figure suggests the classic medulloblastoma related gene mutation occur on gene HG1980-HT2023, U63842 and X67951. This can be also proved in statistical analysis.

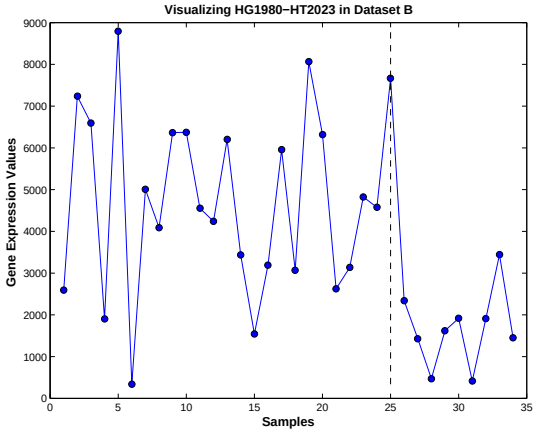
Table 6.8 and 6.9 present the means and standard deviations of two medulloblastoma subclasses on gene HG1980-HT2023, U63842 and X67951. These two tables clearly indicate that samples of classic medulloblastomas have higher means and standard deviation on gene HG1980-HT2023, U63842 and X67951 than samples of desmoplastic. This strongly suggests that gene mutations of HG1980-HT2023, U63842 and X67951 occur differently in classic and desmoplastic medulloblastomas, and higher variance of gene expression values discriminant classic medulloblastomas from desmoplastic medulloblastomas.

Gene accession No.	Classic	Desmplastic
HG1980-HT2023	4747.4	1665.7
U63842	1523.7	90.44
X67951	2813.4	903.22

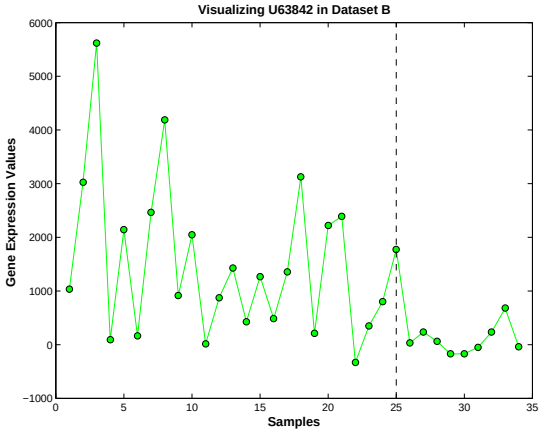
Table 6.8: *The means of gene expression values on gene HG1980-HT2023, U63842 and X67951 across samples of dataset B*

Gene accession No.	Classic	Desmplastic
HG1980-HT2023	2180.6	923.63
U63842	1417	266.05
X67951	1876.7	355.47

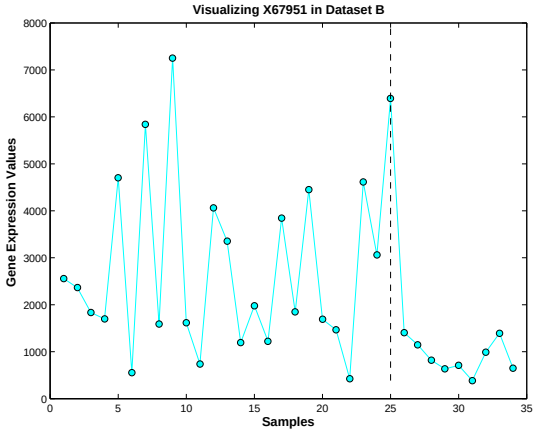
Table 6.9: *The standard deviation of gene expression values on gene HG1980-HT2023, U63842 and X67951 across samples of dataset B*



(a) HG1980-HT2023



(b) U63842



(c) X67951

Figure 6.14: *Three discriminant genes of classic Medulloblastomas*

6.3.2 Discriminant Gene Discovery on Desmoplastic Medulloblastomas

The discriminant genes of desmoplastic medulloblastoma are recorded as their accession number: HG3543-HT3739, X53331 and X65724 as shown in Table 6.10.

Figure 6.15 indicates the variable gene expression values of HG3543-HT3739, X53331 and X65724 across the 34 samples. In the subfigures, the dashed line is used to separate the samples of classic and desomplastic of medulloblastomas. The figure indicates that gene HG3543-HT3739, X53331 and X65724 perform higher variance in the class of desmoplastic medulloblastomas. We also applied the statistical analysis tool to test this finding. Table 6.8 and 6.9 present the means and standard devia-

Gene No.	Accession No.	Descriptions
G5278	HG3543-HT3739	Insulin-Like Growth Factor 2
G4250	X53331	MGP Matrix protein gla
G4426	X65724	NDP Norrie disease (pseudoglioma) protein

Table 6.10: *Discriminant genes for desmoplastic medulloblastomas*

tions of gene expression values on gene HG3543-HT3739, X53331 and X65724 across samples of dataset B.

Samples of desmoplastic present higher means and standard deviations with gene HG3543-HT3739, X53331 and X65724. This finding suggests desmoplastic medulloblastoma is more likely to occur gene mutation on gene HG3543-HT3739, X53331 and X65724, and these three genes are discriminant genes of desmoplastic medulloblastomas.

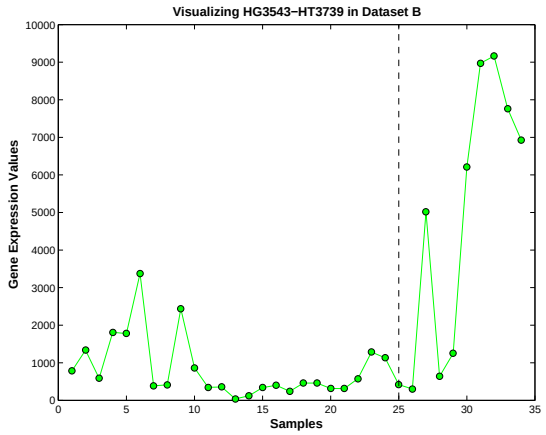
Gene accession No.	Classic	Desmplastic
HG3543-HT3739	822.96	5138.6
X53331	1345.2	4605.6
X65724	70.96	371.55

Table 6.11: *Means of gene expression values on gene HG3543-HT3739, X53331 and X65724 across samples of dataset B*

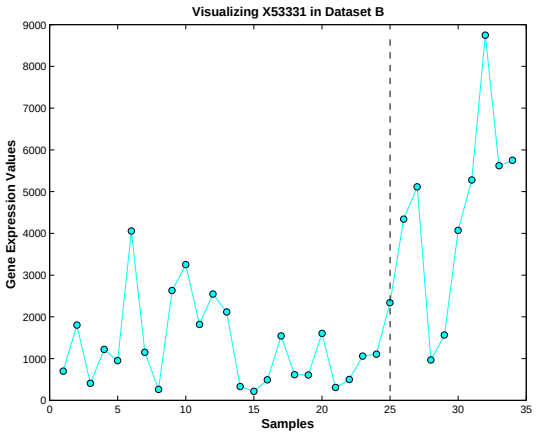
Gene accession No.	Classic	Desmplastic
HG3543-HT3739	798.70	3551.6
X53331	1018.6	2319.9
X65724	117.39	195.03

Table 6.12: *Standard deviation of gene expression values on gene HG3543-HT3739, X53331 and X65724 across samples of dataset B*

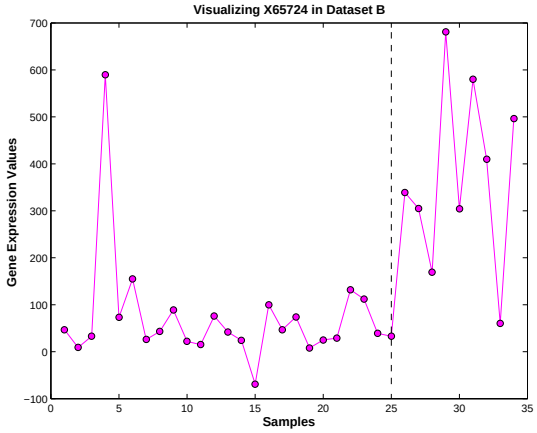
The finding from the comparison of means and standard deviation supports six discriminant genes in terms of principal histological subclass of medulloblastomas. Notice that the prediction of six discriminant genes is not a substitute for traditional diagnostics, since some of cancer cases still depend on individual patient. We only try to find out knowledge that would help to make decision for clinical diagnostics.



(a) HG3543-HT3739



(b) X53331



(c) X65724

Figure 6.15: *Three discriminant genes of desmoplastic medulloblastomas*

6.4 Analysis on Clinical Outcome of Medulloblastomas

6.4.1 Discriminant Gene Discovery for Clinical Outcome Prediction of Medulloblastomas

A clinical challenge concerning medulloblastoma is the highly variable response of patients to therapy. This issue is also indicated in our experiment of dataset C that includes 60 samples (39 medulloblastoma survivors and 21 treatment failures) with 7,129 genes. In the experiment, we have defined that the SNR values of gene selection

are very low (less than 0.5) for both of classes. This implies the variance between the classes may be quite similar, which is much more difficult to extract the common discriminant genes between the samples. In Section 6.2, we have discovered three genes that are able to discriminate the medulloblastomas from other samples. For the knowledge discovery on clinical outcome of medulloblastomas, we want to find out how these genes discriminate different classes. We firstly compared the mean and standard deviation of survivors and failures. Table 6.13 presents the mean of three defined medulloblastoma discriminant genes. Table 6.14 indicates the standard deviation of these genes in two classes of treatment outcomes of medulloblastomas.

Gene accession No.	**Class 1**	**Class 2**
M93119	2024.66	2620.97
X06617	14693.5	14410.8
U05012_ s	-15.0476	**113.378**

Table 6.13: *Mean of gene expression values on gene M93119, X06617 and U05012_ s across samples of dataset C.*

Gene accession No.	**Class 1**	**Class 2**
M93119_ at	2414.17	2460.18
X06617_ at	4372.01	3807.16
U05012_ s_ at	297.507	**421.784**

Table 6.14: *Standard deviation of gene expression values on gene M93119, X06617 and U05012_ s across samples of dataset C.*

Table shows that failures and survivors of medulloblastomas have the most significant difference on Gene U05012_s. TrkC is one of the gene symbols for U05012_s, which is suggested as a molecular basis for the variability of medulloblastoma outcome in several related researches (John et al., 1999 and Grotzer et al., 2000). In literature, the low expression value of TrkC has been suggested as unfavourable to medulloblastoma patients, but it does not appear in this study. Figure 6.16 shows the gene expression values of TrkC across the samples of dataset C.

In this figure, TrkC performs more variable value in the class of survivors. Some of survivors even presents lower expression values than failures. The most of failures perform in a short value distribution. All these findings suggest that survivors of medulloblastoma have very diverse presentation on TrkC, against the failures have

quite similar variance. Overall the significant difference on TrkC implies that TrkC is favourable to discriminate medulloblastoma outcome.

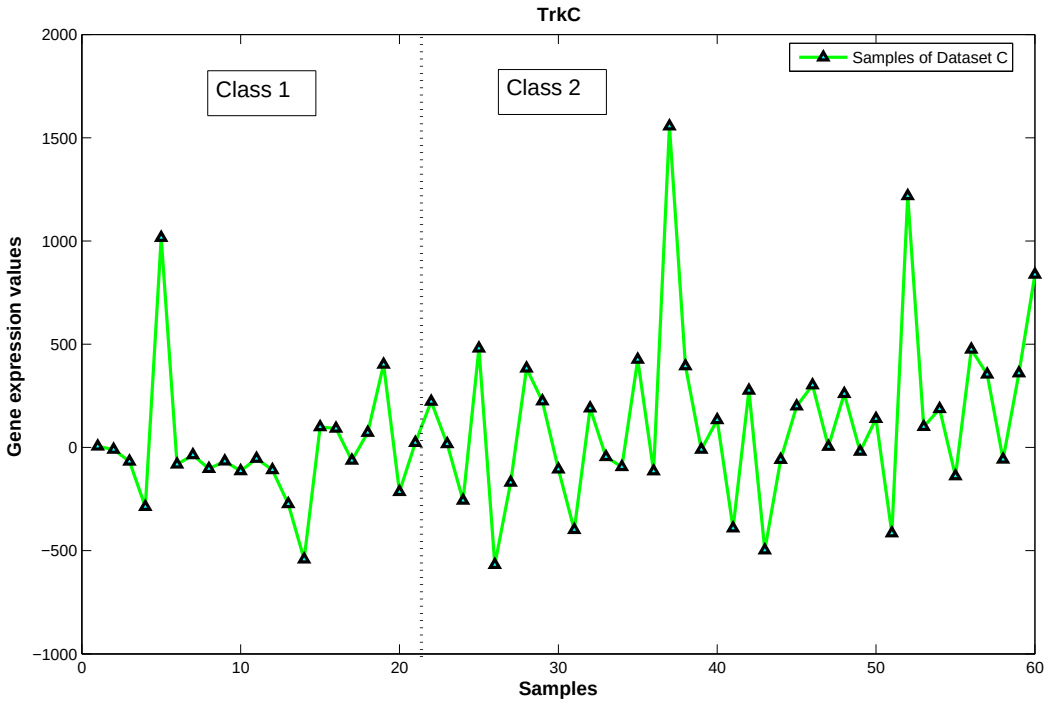


Figure 6.16: *Gene expression values of TrkC across the samples of dataset C.*

6.4.2 Knowledge Discovery on Interaction between Genes and Drugs

In this research, we are more interested in the relationship between the chemotherapy and patient responses (alive or dead). This could be presented in the gene expression value. In the Pomeroy's data, nine chemotherapy drugs have been recorded including carboplatin, CCNU, cisplatin, cytoxan, etoposide, methotrexate, thiotepa and vincristine. All the drugs are antineoplastic medicationmt that interferes with the growth of cancer cells and slows their growth and spread in the body. For the further information see http://www.drugs .com/. Due to the interaction between the drugs and limited sample size, we only analyse the samples that have only been treated by "cisplation", "cytoxan" and "vincristine" during in their treatment period.

We use ontology-based query tool to capture the 36 samples that have been treated by "cisplation", "cytoxan" and "vincristine" from dataset C. 9 samples are selected from failures (class 1). 27 samples are selected from survivors (class 2). Based on their personalised top-ranked genes, we generate three discriminant genes for both class 1 and class 2 by using CNS ontology. Table 6.15 describes the details of these three genes.

Gene No.	Accession No.	Descriptions
G3185	L17131_ rna1	High mobility group protein (HMG-I(Y)) gene exons 1-8
G2996	M96739	NSCL-1 mRNA sequence
G844	D14686	AMT Glycine cleavage system

Table 6.15: *Discriminant genes for outcomes of medulloblastomas*

To test our findings, we applied WWKNN to analyse these 36 samples with three discovered discriminant genes. Table 6.16 presents the WWKNN classification results. The result indicates 35 out of 36 samples can be successfully classified. This implies these three genes have a very close relationship with drug of "cisplation", "cytoxan" and "vincristine".

Number of K	Class 1	Class 2	Total
2	100.00%	96.30%	97.22%
3	100.00%	96.30%	97.22%
4	100.00%	96.30%	97.22%
5	100.00%	96.30%	97.22%

Table 6.16: *WWKNN results on drugs*

Figure 6.17 indicates the these 36 samples in a 3D space based on the three discriminant genes. We can see that samples of two classes are clearly separable.

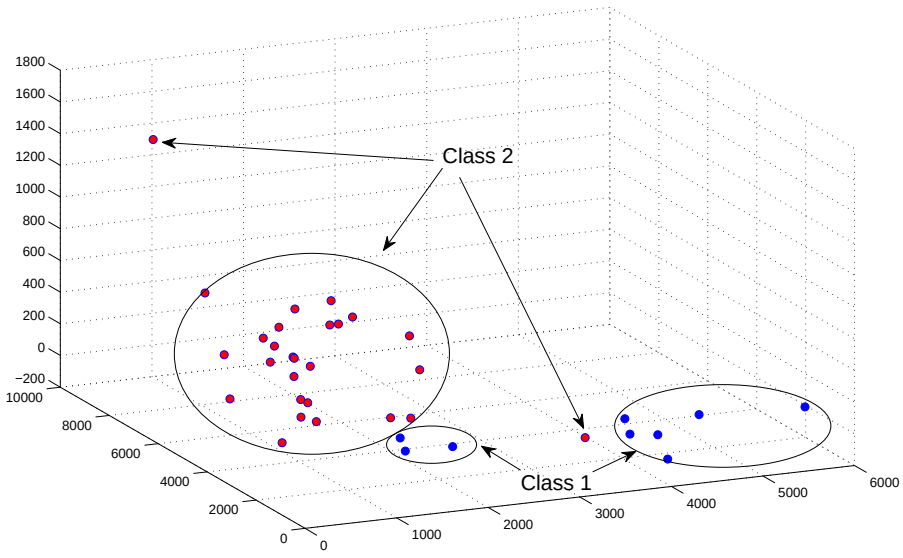


Figure 6.17: *Visualising 3 discriminant genes in 3D*

6.5 Upgrading BGO with the newly Discovered Knowledge

As part of BGO system development, the discovered discriminant genes from this thesis will be imported into information space of BGO. We have explained how to import the discovered knowledge from CNS ontology to BGO in Chapter 4. CNS ontology is an extension of BGO system. However, they still have slightly difference in terms of ontology based framework. In CNS ontology, we identify each gene by using the gene accession number. But genes are presented as their official symbols in BGO. To import knowledge of CNS ontology into BGO system, we have to define the official symbols for these discriminant genes. Table 6.17 describes the official symbols for these discriminant genes. The detailed information of these gene is explored from online biological data source, Gene ontology and NCBI. The upgraded version of BGO could be download from KEDRI's website.

Gene No.	CNS ontology ID	BGO ID
G2365	M93119	INSM1
G4092	X06617	RPS11
G6435	U05012_s	NTRK3
G4741	X86693	SPARCL1
G3239	U45955	GPM6B
G6035	Z31560_s	SOX2
G618	D83735	CNN2
G1553	L38969	THBS3
G6724	D83174_s	CBP1
G6368	M80397_s	POLD1
G982	HG4178-HT4448	Af-17
G4152	X14830	CHRNB1
G6818	HG1980-HT2023	Undefined symbol
G3541	U63842	NEUROG1
G4466	X67951	PRDX1
G5278	HG3543-HT3739	IGFBP2
G4250	X53331	MGP
G4426	X65724	NDP
G1352	L17131	HMGA1
G2404	M96739	NHLH1
G237	D14686	AMT

Table 6.17: *The official symbols for these discriminant genes*

6.6 Conclusion

For knowledge discovery, each personalised data presents a single case of cancer diagnosis. From a collection of individual cases, we could extract a few genes that have the most common connections with the particular disease. In this thesis we call these genes are discriminant genes. These discriminant genes cannot represent gene mutation for entire patients, But the knowledge of these genes could provide a very important adjunct for the future cancer diagnosis.

Above analysis presents the idea of using both modeling method and ontology method for knowledge discovery in the field of cancer diagnosis. Based on the above results, we can conclude that the different classes of CNS tumour could be separated by a few discriminant genes. For the every experimental problem, we have identified three discriminant genes for each tumour class. All of discovered discriminant

genes have been evaluated by a statistical approach. There are strong evidences that discriminant genes are able to represent each class. We also develop an analysis that is proposed to discover gene reactions after use treatment drug. This anaylsis predict three genes that have significant reaction with “cisplation”, “cytoxan” and “vincristine”. This result is tested by using WWKNN.

The CNS system as an extension of BGO presents the relationship between CNS cancer disease and discriminant genes. All of genes that are recorded as discriminant genes in CNS ontology have been imported into BGO with more detailed informations. BGO system presents the detailed information of these genes and the connection between these genes and different brain functions.

Chapter 7

Conclusion

7.1 Summary

We have presented an ontology-based knowledge discovery in CNS tumour diagnosis using gene expression data in this thesis. The general idea behind this research is to use both modeling approaches and ontology systems to define the discriminant genes for CNS diagnosis. The models and ontology-based knowledge framework are built based on the analysis of benchmark Pomeroy's data. It contains 99 samples with 7,129 genes. This thesis is also proposed as a part of the brain gene ontology (BGO) system development. All of discovered knowledge from this thesis have been imported into BGO with particular structure.

To achieve the goal of finding the discriminant genes for cancer diagnosis, we have designed a two-stage research procedure. The first stage is computational modeling experiment. It includes four steps that are data normalisation, gene selection, cross validation and applying classifiers. Chapter 2 reviewed the major techniques for each step and five common classifiers in microarray studies. The second stage is the discriminant genes discovery. In this stage, we have imported the modelled genes into ontology. Based on the certain knowledge framework, we extracts the discriminant genes. Every discovered discriminant gene has been analysed and evaluated by using statistical techniques. All results have supported our findings in terms of discriminant genes for cancer diagnosis.

In the computational modeling experiment, the first challenge is the gene selection. It

would impacts the final classificational accuracy. Two major gene selection methods (T-test and Signal-to-Noise Ratio) have been described in Chapter 2. In Chapter 4, we compared these two methods. The solution suggested that Signal-to-Noise Ratio (SNR) is more suitable to this study. However, the traditional SNR algorithm is only created for two-class problems. In our experiment, three multi-class problems are involved. Due to this issue, we developed an One-Vs.-All (OVA) scheme. It is used to simplify a multi-class problem to several two-class problems. We then built the OVA scheme on top of SNR. To determine whether the OVA-SNR is efficacious in this study, we applied two case studies including a multi-class problem and a two-class problem. Two classifiers have been applied to evaluate the selected genes. The results suggested that OVA-SNR is highly efficacious not only in multi-class problem, but also in two-class problem.

In Chapter 4, we have introduced the weighted-weighted k nearest neighbour (WWKNN). It is personalised modeling algorithm that is also capable of discovering the important information for each individual experimental sample. However, the previous version of WWKNN only can be applied on the two-class problem. Due to this limitation, we have developed two approaches of WWKNN to solve multi-class problem. One is called multilayer threshold WWKNN, which is to multiply the probability threshold for the final classification of WWKNN. The other approach is called OVA-WWKNN. This approach uses similar strategy in OVA-SNR, that simplify a multi-class problem to several two-class problems. At end of this chapter, we described the prototype of the CNS ontology system.

Chapter 5 described the computational experiment setup and experimental result. Six discussed classifiers have been applied. We recorded the best results of every classifier on each problem. In terms of the classification accuracy, WWKNN significantly outperforms other five classifiers. Its accuracy is also much better than Pomeroy's work. In the multi-class classification, both developed WWKNN approaches have been used. Interestingly the OVA-WWKNN produced more accurate result than multilayer threshold WWKNN. We investigate this finding by comparing the performance of two approaches in multi-class classification. The outcomes suggest the downside of multilayer threshold WWKNN that produces more classificational errors with more nearest neighbours.

The knowledge discovery is considered in Chapter 6. Both WWKNN and CNS ontol-

ogy have been applied to extract the discriminant genes. Based on the personalised ranks of each gene, we have extracted three discriminant genes for four types of discussed CNS tumours and two subclasses of medulloblastomas. For the clinical outcomes of medulloblastomas, this method is probably impractical since individual patient has different response on treatments. In this problem, our focus is to discover the gene response after treatment. Since the interaction between different drugs, only 36 samples have been analysed,which have been treated with same three drugs. The CNS ontology conducts us to find the three genes with significant importance to every sample. To test this finding, we have built a WWKNN model on these 36 samples with three important genes. WWKNN achieved a nearly 100% accuracy. This outcome strongly supported that performance of these three genes are capable to discriminate samples from survivors and failures.

In general, we could conclude that this thesis has achieved the general purpose of our study. There are three major contributions that have been delivered.

1. This thesis offered a comparative study of major modeling to the area of CNS tumour data analysis. This comparison involves six classifiers from global, local and personalised modeling approaches. The final result indicates that personalised modeling is more efficient for this particular problem.

2. Several discriminant genes have been discovered. These genes cannot represent gene mutation for entire patients, But these genes could provide a very important adjunct for the future cancer diagnosis.

3. All of discriminant genes have been imported to both CNS ontology and BGO system. The reusable knowledge from BGO and CNS ontology will contribute more related researches in the future.

7.2 Future Work

If time allowed there are numerous developments we would like to attempt. The most immediate of there are listed below:

- In this study, we have defined several genes that relate to CNS cancer disease. Due to the limited sample size we cannot prove our findings on other real

samples. In the future work, we could experiment with larger sample size datasets to test our findings.

- The approaches used to extract discriminant genes for Pomeroy's data could be implemented and tested on other brain related microarray datasets.
- For ontology knowledge construction, our focus is to build the knowledge bridge between CNS ontology and BGO system. In the future work, we intend to connect more online available ontology systems in the area of microarray study. This will involve not only brain related ontology, but also some other biological ontology.
- A journal publication regarding the findings of this thesis is in progress.

Appendix A

MATLAB code of One-Vs.-All scheme and WWKNN

MATLAB code of One-VS.-All scheme

```
% ============================================
% OVA --- One-vs.-All scheme for WWKNN
% Author: Yuepeng Wang, Nov, 2007
% ============================================
function outputRes = crossOVA(infile, thr0)
clc;

%load files
eval(sprintf('load %s.txt; dat1=%s; clear %s', infile, infile, infile));
[num, dim] = size(dat1);

% Parameter Setting
thr= double (1:2);
for i= 1:2
    thr(i)= thr(i)+thr0; % the threshold for determining the classlabel
end

% --- choose validation mode ---
ansCross = questdlg('Select Validation Mode','Mode Selection', 'K-Fold Cross-validation', ...
    'Leave-one-out Cross-validation', 'Leave-one-out Cross-validation');
if strcmp(ansCross, 'K-Fold Cross-validation')
    ans1 = questdlg('How many folds you want to use?','K-fold selection', '3','5','10','5');
    numFold = str2num(ans1);
    disp(sprintf('\n --- fold %d--- \n', numFold));
elseif strcmp(ansCross, 'Leave-one-out Cross-validation')
    numFold = num;
    disp(sprintf('\n --- Leave-one-out Cross Validation --- \n'));
end
```

```
% ---------- Model parameter setting -------------
% --- WWKNN model parameters ---

prompt={'Enter the number of K', 'Enter the number of features '};
name='Input parameters for WWKNN model';
numlines=1;
ansPara = inputdlg(prompt, name, numlines, defaultanswer);
nbr = str2num(ansPara{1});
nfeat = str2num(ansPara{2});
if nfeat > (dim-1)
    nfeat = dim-1;
end

datNorm1=dat1;
clsLbl = unique(datNorm1(:, end));
dimLbl = length (clsLbl);
overallAcc = [];
clsnum=zeros(1,dimLbl);
allAcc=zeros(1,dimLbl);
clsAcc=zeros(1,dimLbl);

%--------OVA Scheme---------
record=zeros(nfeat,numFold);
for set=1:dimLbl % split a multi-class problem to several two-class
    datNorm=zeros(num,dim);

    i=0;
    for j=1:num
        if datNorm1(j,dim)==set
            i=i+1;
            for m=1:dim-1
                datNorm(i,m)=datNorm1(j,m);
                datNorm(i,dim)=1; % set the target class as 1
            end
        else
            i=i+1;
            for m=1:dim-1
                datNorm(i,m)=datNorm1(j,m);
                datNorm(i,dim)=2;  % set other class as 2
            end
        end
    end

    for m=1:num
        if datNorm(m,dim)==1;
            clsnum(set)=clsnum(set)+1;
        end
    end

    r1 = fix(num/numFold);
```

```
    for k = 1:numFold
        if k < numFold
            tstD = datNorm((k-1)*r1+1:k*r1, :);
            if k == 1
                trnD = datNorm(k*r1+1:end, :);
            else
                trnD = [datNorm(1:(k-1)*r1, :); datNorm(k*r1+1:end, :)];
            end
        else
            tstD = datNorm((k-1)*r1+1:end, :);
            trnD = datNorm(1:(k-1)*r1, :);
        end

        tstClasslabel = tstD(:, end);

        % --------- Call WWKNN model ------------
        res = wwknn(trnD, tstD, nbr, nfeat);

        for j=1:dimLbl
            if res.Output <= thr(1)
                res.Output=1;
            else
                res.Output=2;
            end
        end

        if strcmp(ansCross, 'Leave-one-out Cross-validation')
            if tstClasslabel == 1
                for i=1:nfeat
                    record(i,k)=res.featureID(i);
                end
                if res.Output == tstClasslabel
                    allAcc(set)=allAcc(set)+1;
                end
            end
            clsAcc(set)=(allAcc(set)/clsnum(set))*100;
        end

        if (datNorm(:,end)==1)
            % --------- Output actual and predicted -----------------
            if tstClasslabel == res.Output
                disp(sprintf('        %d                    %d
                            %d', k, tstClasslabel, res.Output));
            else
                disp(sprintf('        %d                    %d
                            %d     *', k, tstClasslabel, res.Output));
            end
        end
    end
end
save  record_c.txt record  -ASCII -tabs
```

```
% -----------Output Overall Accuracy ------------------
for set=1:dimLbl
    if strcmp(ansCross, 'K-Fold Cross-validation')
        disp('*******************************************************\n');
        disp(sprintf('*** Class 1 Overall Accuracy of %d folds Crossvalidation:
                    %f%% ***', numFold, mean(overallClass1)));
        disp(sprintf('*** Class 2 Overall Accuracy of %d folds
                    Crossvalidation: %f%% ***', numFold, mean(overallClass2)));
    elseif strcmp(ansCross, 'Leave-one-out Cross-validation')
        disp(sprintf('       Class %d Overall Accuracy of LOO
                    Crossvalidation: %4.2f%% ', set, clsAcc(set)));
    end

end
overallAcc = (sum(allAcc))/num*100;
disp(sprintf('*** Overall Accuracy of LOO Crossvalidation: %4.2f%% ***', overallAcc));
```

MATLAB code of weighted-weighted k nearest neighbours

```
% ==========================================
% WWKNN --- Weighted distance weighted variables  K-nearest neighbours
% parameter:   Train: Training dataset
%                     test: test dataset
%                     nbr: Number of neighbours
%                     nfeat: Number of features
% Output:         res
% Note:             This WKNN model is used in unbiased model
% last modified:    Raphael, Oct, 2007
% ==========================================
function res = wwknn(train, test, nbr, nfeat)
[ntr, mtr] = size(train);
[nte] = size(test, 1);

% unbiased FEATURE Selection on the training set
[fid, snrout] = snrV2(train,mtr-1,1);
% Feature selection on the training set.
fid = fid(1:nfeat);
%snrout: snrvalues for each variable
%fid: feature id ranked by snr-values
clear dataout;
clear snrout;
rcoef = zeros(1,nfeat);
rifid = zeros(1,nfeat);

for t = 1:nte
    distance = ndist(train(:,fid), test(t,fid), 'euclidean distance');
    strain = [distance train];
    strain = sortrows(strain, 1);
    strain = strain(1:nbr, 2:end);          % nbr neighbours
    try
```

```
% === Weighted VARIABLE(selected features) based on SNR ranking ===
       [ifid, snrout] = snrV2(strain, size(strain, 2) -1, 1);
% calculate the SNR value for nfeature genes(Weighted Variable)
       snrout(find(isnan(snrout))) = 0;
% set nan snr value to 0 since it means the column value's std  = 0;
        coef = (snrout - min(snrout))/(max(snrout) - min(snrout));
    catch
        ifid = fid;
        snrout = ones(size(ifid,1),1);
        snrout(find(isnan(snrout))) = 0;
% set nan snr value to 0 since it means the column value's std  = 0;
        coef = snrout;
    end
    clear dataout;

    for i = 1:nbr
        d(i) = sqrt(sum( ((test(t, ifid)-strain(i, ifid)).*coef').^2)) /
        length(ifid);
    end
    d = d/max(d);
    output(t) = sum((1-d)' .* strain(:,end))/sum(1-d);
    rcoef = coef';
    rifid = ifid';
end

res.featureID = rifid;
res.Coef = [fid coef];
% the coef matrix of weighted feature (selected genes) with gene IDs
res.Output = output;

% ================================================
% dataout:   the reduced set of data set with descending SNR
% geneout:   the list of gene corresponding to the ranked SNR
% Author: Liang Goh (7/01/03) - coded amidst Monet, Bach and Dvorak!
% ================================================
function [geneout, snrout] = snrV2(datain,rankno,TestType)
if nargin<3
    TestType=1
end

data = datain;
[row, col] = size(datain);
% sort the matrix in ascending order based on last column (i.e. classes)
datain = sortrows(datain,col);
% exclude the feature haveing same value in both class1 and class2
remFeat = find(var(datain(:, 1:end-1)) == 0);
% count the number in each classes
%    classmat = countclass(datain(:,col));
%    [classrow,classcol] = size(classmat);
% split the data into the classes
geneout   = [];
snrout    = [];
```

```
allsnrmat = [];
classrow=length(unique(datain(:,col)));

if (classrow <= 2)
    LoopCounter=1;
else
    LoopCounter=classrow;
end

for j=1:LoopCounter
    t1=find(datain(:,col)==j);
    t2=find(datain(:,col)~=j);
    mat1 = datain(t1,:);
    mat2 = datain(t2,:);
    % Calculate the snr for each gene within each class
    snrmat        = calsnr(mat1,mat2, remFeat, TestType);

    % rank snr in ascending order
    [snr1, gene1] = sort(snrmat, 2);

    % merge results
    snrout        = [snrout; snr1(:,col-rankno:col-1)];
    geneout       = [geneout; gene1(:,col-rankno:col-1)];
    allsnrmat     = [allsnrmat; snrmat];
end

%change to descending order ranking
snrout = flipud(snrout');
geneout = flipud(geneout');

% when output classes > 2, there will be a gene list for each class, so
% need to merge the genes for each class into one.
if (classrow > 2)
% Take top half of genes identified for each class.  Need to ensure
% that there is no overlap of genes.
    [geneout,snrout] = mergegene(geneout,snrout);
    dataout = datain(:,geneout(:,1));
else
    % 2 or less output classes
    geneout = geneout(:,1);
    dataout = datain(:,geneout(:,1));
end

% dataout = [dataout,datain(:,col)];
dataout=[data(:,geneout), data(:,end)];

function [snrmat] = calsnr(mat1, mat2,remFeat, TestType)
% Calcuate snr for each column based on both matrices mat1 and mat2
% Note last column is output classes, so is not calculated.
% Author: Liang Goh (16/01/03)

[row, col] =size(mat1);
```

```
snrmat = [];

gmean=mean(mat1(:,1:col-1));
omean=mean(mat2(:,1:col-1));
gstd = std(mat1(:,1:col-1));
ostd = std(mat2(:,1:col-1));
onevectorind=find(gstd==0 & ostd==0);
otherind=find(gstd | ostd);
if TestType==1 %testing for any difference in genes between classes
    if ~isempty(onevectorind)
        snrmat(onevectorind)=abs(gmean(onevectorind) -
        omean(onevectorind))./(gmean(onevectorind)
        +omean(onevectorind));
        if ~isempty(remFeat)
             snrmat(remFeat) = 0;
        end
    end
    if ~isempty(otherind)
        snrmat(otherind) = abs(gmean(otherind) -
        omean(otherind))./(gstd(otherind)
        + ostd(otherind));
    end
elseif TestType==2
%testing to see if class 1 is upregulated with respect to class 2
    if ~isempty(onevectorind)
        snrmat(onevectorind)=(gmean(onevectorind) -
        omean(onevectorind))./(gmean(onevectorind)
        +omean(onevectorind));
    end
    if ~isempty(otherind)
        snrmat(otherind) = (gmean(otherind) - omean(otherind))
        ./(gstd(otherind)
        + ostd(otherind));
    end
elseif TestType==3
%testing to see if class 2 is upregulated with respect to class 1
    if ~isempty(onevectorind)
        snrmat(onevectorind)= -(gmean(onevectorind)-
        omean(onevectorind))./(gmean(onevectorind)
        +omean(onevectorind));
    end
    if ~isempty(otherind)
        snrmat(otherind) = -(gmean(otherind) - omean(otherind))
        ./(gstd(otherind)+ostd(otherind));
    end
end

%=====================================================
% Merge the selected genes into one gene list
% geneout:    the merged gene list
% genein:       the mutliple columns of gene list
% Author: Liang Goh (03/02/03) - with the accompaniment of Enya!
```

```
% ====================================================
function [geneout,snrout] = mergegene(genein, snrin)

[row,col]=size(genein);
geneout=genein(:,1);
snrout=snrin(:,1);

for i=2:col
    testmat = genein(:,i);
    snrmat = snrin(:,i);
    result = checkdata(geneout,testmat);
    [r,c]=size(result);
    % common genes found, so should grab those that are not common, so set
    % common ones to empty matrix.
    testmat(result(:,2),:)=[];
    snrmat(result(:,2),:)=[];
    geneout=[geneout; testmat];
    snrout=[snrout;snrmat];
end
return;

% ===================================
% checkdata:     check if data is repeated in both data sets
% ===================================
function [result] = checkdata(gdata, rdata)
[r1, c1] = size(gdata);
[r2, c2] = size(rdata);

cmin = min(c1,c2);

result = [];

for i=1:r1
    for j=1:r2
        if gdata(i,1:cmin) == rdata(j,1:cmin)
            result = [result; i, j];
        end
    end
end

% ===================================
% ndist:
% thedata:  training data
% sample :   testing data
% ================================
function distance = ndist(thedata, sample, type)
if nargin < 3
    type = 'Euclidean Distance'
end

switch lower(type)
    case 'euclidean distance'
```

```
        distance = euc(sample, thedata)';
    case 'manhattan distance'
        distance = sum(abs(ones(size(thedata,1), 1)*sample - thedata)')';
end
return

% ====================================
% euc:  ensure the 2 input matrices can be calculated by dist
% ====================================
function d = euc(a, b)
if size(a,2) ~= size(b,2)
    error('A and B should have the same number of features');
end

if size(a, 1) == 1 & size(b, 1) ~= 1;  % if a is a 1 dimenstion vector
    a = ones(size(b, 1), 1)*a;
    d = dist(a', b');
d = d(1, :);
elseif size(b, 1) == 1 & size(a, 1) ~= 1;
    b = ones(size(a,1),1)*b;
d = dist(a',b');
d = d(:,1);
elseif size(a) == size(b)
    d = dist(a', b');
else
    error('A and B should have the same number of samples or have a single
          sample in A and many samples in B');
end

% ================================
% dist:  calculate the Euclidean distance
% ================================
function d = dist(a,b)
% DISTANCE - computes Euclidean distance matrix
% E = distance(A,B)
%    A - (DxM) matrix
%    B - (DxN) matrix
% Returns:
%    E - (MxN) Euclidean distances between vectors in A and B
% Description :
%    This fully vectorized (VERY FAST!) m-file computes the
%    Euclidean distance between two vectors by:
%                ||A-B|| = sqrt ( ||A||^2 + ||B||^2 - 2*A.B )
% Example :
%    A = rand(400,100); B = rand(400,200);
%    d = distance(A,B);
% Author   : Roland Bunschoten
%            University of Amsterdam
%            Intelligent Autonomous Systems (IAS) group
%            Kruislaan 403  1098 SJ Amsterdam
%            tel.(+31)20-5257524
%            bunschot@wins.uva.nl
```

```
% Last Rev : Oct 29 16:35:48 MET DST 1999
% Tested   : PC Matlab v5.2 and Solaris Matlab v5.3
% Thanx    : Nikos Vlassis
% Copyright notice: You are free to modify, extend and distribute
%    this code granted that the author of the original code is
%    mentioned as the original author of the code.

if (nargin ~= 2)
   error('Not enough input arguments');
end

if (size(a,1) ~= size(b,1))
   error('A and B should be of same dimensionality');
end

aa=sum(a.*a,1); bb=sum(b.*b,1); ab=a'*b;   % sum the data alogn the colomn
d=sqrt(abs(repmat(aa',[1 size(bb,2)]) + repmat(bb,[size(aa,2) 1]) - 2*ab));
```

Appendix B

Selected genes

Dataset A, A1 and A2

Gene No.	Accession No.	Description
G3311	U49837	LIM protein MLP mRNA
G5589	X91653_s	DNA for exon encoding for N-acetylglucosaminyltransferase V (340 bp)
G1221	L06419	PLOD Lysyl hydroxylase
G3718	U77718	Desmosome associated protein pinin mRNA
G3595	U67191	Multiple exostosis-like protein (EXTL) mRNA
G2683	U08015	NF-ATc mRNA
G4510	X71428	RNA-BINDING PROTEIN FUS/TLS
G6476	U72935_cds3_s	ATRX gene (putative DNA dependent ATPase and helicase) extracted from Human putative DNA dependent ATPase and helicase (ATRX) gene
G4529	X73608	Testican
G4709	X84003	TAFII18 mRNA for transcription factor TFIID
G5653	Z75190_s	Apolipoprotein E receptor 2
G842	HG2415-HT2511	Transcription Factor E2f-2
G6964	L35594	Autotaxin mRNA

Continued on Next Page...

Table B.1 – Continued

Gene No.	Accession No.	Description
G4578	X76383	HE3(alpha)
G2306	M86699	TTK TTK protein kinase
G744	D87684	KIAA0242 gene, partial cds
G3880	U85992	Clone IMAGE:35527 unknown protein mRNA, partial cds
G3928	U90426	Nuclear RNA helicase
G2129	M63391_rna1	Desmin gene
G732	D87462	KIAA0272 gene, partial cds
G2184	M69023	Globin gene
G1764	M17754	BN51T BN51 (BHK21) temperature sensitivity complementing
G2033	M55621	MGAT1 N-acetylglucosaminyltransferase I
G6893	U24685	Anti-B cell autoantibody IgM heavy chain variable V-D-J region (VH4) gene, clone E11, VH4-63 non-productive rearrangement
G2267	M81933	CDC25A Cell division cycle 25A
G370	D31884	KIAA0063 gene
G2070	M59979	PTGS1 Prostaglandin-endoperoxide synthase 1 (prostaglandin G/H synthase and cyclooxygenase)
G873	HG2755-HT2862	T-Plastin
G2267	M81933	CDC25A Cell division cycle 25A
G2309	M86752	TRANSFORMATION-SENSITIVE PROTEIN IEF SSP 3521
G3382	U52969	BRAIN SPECIFIC POLYPEPTIDE PEP-19
G3404	U55209	Myosin VIIa transcript 2 mRNA
G4779	X90857	-14 gene, containing globin regulatory element
G1532	L37936	MITOCHONDRIAL ELONGATION FACTOR TS PRECURSOR
G648	D85527	LIM domain, partial cds

Dataset B

Gene No.	Accession No.	Description
G6815	HG1980-HT2023	Tubulin, Beta 2
G4463	X67951	PAGA Proliferation-associated gene A (natural killer-enhancing factor A)
G3538	U63842	Neurogenic basic-helix-loop-helix protein (neuroD3) gene
G4406	X64330	ATP-citrate lyase
G5957	J03241_s	TGFB3 Transforming growth factor, beta 3
G3234	U44839	Putative ubiquitin C-terminal hydrolase (UHX1) mRNA
G5103	Z27113	DNA-DIRECTED RNA POLYMERASE II 14.4 KD POLYPEPTIDE
G4116	X12447	ALDOA Aldolase A
G4158	X15183	60S RIBOSOMAL PROTEIN L13
G3678	U73328	DLX7 Distal-less homeobox 7
G3500	U61263	Acetolactate synthase homolog mRNA
G3168	U40391_rna1	Serotonin N-acetyltransferase gene
G3072	U33839	No description available for U33839
G4667	X81817	6C6-Ag mRNA
G3470	U59913	SMAD5 (Smad5) mRNA
G4218	X51804	PUTATIVE RECEPTOR PROTEIN
G4306	X57398	NME1 Non-metastatic cells 1, protein (NM23A) expressed in
G4028	X02152	LDHA Lactate dehydrogenase A
G5275	HG3543-HT3739	Insulin-Like Growth Factor 2
G4247	X53331	MGP Matrix protein gla
G4423	X65724	NDP Norrie disease (pseudoglioma) protein
G226	D14530	40S RIBOSOMAL PROTEIN S23
G2953	U25789	Ribosomal protein L21 mRNA

Continued on Next Page...

Table B.2 – Continued

Gene No.	Accession No.	Description
G4941	Y00757	SGNE1 Secretory granule, neuroendocrine protein 1 (7B2 protein)
G4701	X83543	APXL Apical protein (Xenopus laevis-like)
G1443	L27560	Insulin-like growth factor binding protein 5 (IGFBP5) mRNA
G4246	X52966	RPL35A Ribosomal protein L35a
G5552	L06797_s	PROBABLE G PROTEIN-COUPLED RECEPTOR LCR1 HOMOLOG
G1591	L41066	NF-AT3 mRNA
G5843	HG3431-HT3616_s	Decorin, Alt. Splice 1
G568	D79205	Ribosomal protein L39
G5328	M14745	BCL2 B cell lymphoma protein 2
G6343	X53595_s	APOH Apolipoprotein H
G603	D82345	NB thymosin beta
G2117	M62843	PARANEOPLASTIC ENCEPHALOMYELITIS ANTIGEN HUD
G1783	M19720_rna2	L-myc gene (L-myc protein) extracted from Human L-myc protein gene

Dataset C

Gene No.	Accession No.	Description
G2695	X69150	Ribosomal protein S18
G1352	M36072	RPL7A Ribosomal protein L7a
G1771	X13293	MYBL2 V-myb avian myeloblastosis viral oncogene homolog-like 2

Continued on Next Page...

Table B.3 – Continued

Gene No.	Accession No.	Description
G6531	U14972	Ribosomal protein S10 mRNA
G6064	K03189_f	Chorionic gonadotropin (hcg) beta subunit mRNA
G3185	L17131_rna1	High mobility group protein (HMG-I(Y)) gene exons 1-8
G3028	X13482	U2 SMALL NUCLEAR RIBONUCLEOPROTEIN
G18	L12711_s	TKT Transketolase (Wernicke-Korsakoff syndrome)
G8	L19711	Dystroglycan (DAG1) mRNA
G4130	X04741	UBIQUITIN CARBOXYL-TERMINAL HYDROLASE ISOZYME L1
G5508	U12404	HSPB1 Heat shock 27kD protein 1
G4546	U15008	SnRNP core protein Sm D2 mRNA
G4951	U81375	Placental equilibrative nucleoside transporter 1 (hENT1) mRNA
G3420	X13794_rna1	Lactate dehydrogenase B gene exon 1 and 2 (EC 1.1.1.27) (and joined CDS)
G572	Z49148_s	Enhancer of rudimentary homolog mRNA
G2671	U39318	AF-4 mRNA
G3834	X67247_rna1	RpS8 gene for ribosomal protein S8
G3746	U14968	Ribosomal protein L27a mRNA
G5528	HG613-HT613	Ribosomal Protein S12
G4509	D63880	KIAA0159 gene
G1159	Y07604	Nucleoside-diphosphate kinase
G1806	J04823_rna1	Cytochrome c oxidase subunit VIII (COX8) mRNA
G5433	M13934_cds2	RPS14 gene (ribosomal protein S14) extracted from Human ribosomal protein S14 gene
G752	U30872	CENP-F kinetochore protein mRNA
G4338	M81757	40S RIBOSOMAL PROTEIN S19
G1320	L06419	PLOD Lysyl hydroxylase
G2496	J02611	APOD Apolipoprotein D

Continued on Next Page...

Table B.3 – Continued

Gene No.	Accession No.	Description
G348	D86974	KIAA0220 gene, partial cds
G327	U37673	Neuron-specific vesicle coat protein and cerebellar degeneration antigen (beta-NAP) mRNA
G2196	U28963	Gps2 (GPS2) mRNA
G3320	X69636	mRNA sequence (15q11-13)
G5812	U18018	ETV4 Ets variant gene 4 (E1A enhancer-binding protein, E1AF)
G2032	M97287	SATB1 Special AT-rich sequence binding protein 1 (binds to nuclear matrix/scaffold-associating)
G1478	U78180	Sodium channel 2 (hBNaC2) mRNA alternatively spliced
G1054	S76475	NTRK3 Neurotrophic tyrosine kinase, receptor, type 3 (TrkC)
G3531	D28124	Unknown product
G4173	U70867	Prostaglandin transporter hPGT mRNA
G4484	M17733	Thymosin beta-4 mRNA
G3645	L10333_s	Neuroendocrine-specific protein A (NSP) mRNA
G844	D14686	AMT Glycine cleavage system protein T (aminomethyltransferase)
G6252	S66541_s	B-50=neural phosphoprotein [human, Genomic, 778 nt, segment 3 of 3]
G6810	AC002045_xpt2	A-589H1.2 from Homo sapiens Chromosome 16 BAC clone CIT987-SKA complete genomic sequence, complete sequence.
G2996	M96739	NSCL-1 mRNA sequence
G851	D86963	PTB Ribosomal protein L26
G588	U40271_s	PTK7 Protein-tyrosine kinase 7
G5458	L09229_s	FACL1 Long chain fatty acid acyl-coA ligase
G237	D78012	CRMP1 Collapsin response mediator protein 1
G3485	M74715_s	IDUA Iduronidase, alpha-L-

Continued on Next Page...

Table B.3 – Continued

Gene No.	Accession No.	Description
G654	HG2525-HT2621	Helix-Loop-Helix Protein Delta Max, Alt. Splice 1
G3731	L32164	Zinc finger protein mRNA, 3' end

References

Alpaydim, E. (2004). *Introduction to Machine learning*: MIT Press.

Anthea, M., Hopkins, J., McLaughlin, C. W., Johnson, S., Warner, M. Q., LaHart, D., et al. (1993). *Human Biology and Health*. New Jersey: Prentice Hall.

Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, & Cherry, J. M. (2000). Gene ontology: Tool for the unification of biology. The gene ontology consortium. *Nature Genetics, 25*, 25-29.

Avgeropoulos, N., & Batchelor, T. (1999). New Treatment Strategies for Malignant Gliomas. *The Oncologist, 4*, 209-224.

Ayers, M., Symmans, W. F., Stec, J., Damokosh, A. I., Clark, E., Hess, K., et al. (2004). Gene Expression Profiles Predict Complete Pathologic Response to Neoadjuvant Paclitaxel and Fluorouracil, Doxorubicin, and Cyclophosphamide Chemotherapy in Breast Cancer. *Journal of Clinical Oncology.*

Bartons, C. (2008, January 26). Gene genie. *New Zealand Herald.*

Benuskova, L., & Kasabov, N. (2007). *Computational Neurogenetic Modeling.* New York: Springer.

Berman, D., M, Karhadkar, S., Hallahan, A., & Pritchard, J. (2002). Medulloblastoma Growth Inhibition by Hedgehog Pathway Blockade. *Science, 297*(5586), 1559-1561.

Berners-lee, T. (1998). *Semantic web road map.* from http://www.w3.org/Design Issues/Semantic.html.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American.* Retrieved from http://www.cs.aue.auc.dk/ legind/F8S_IR_kursus/ IRobligatorisk/semantic_web.pdf

Black, P. (2004). *"Euclidean distance", in Dictionary of Algorithms and Data Structures.* from http://www.nist.gov/dads/HTML/euclidndstnc.html

Breiman, L., & Spector, P. (1992). Submodel selection and evaluation in regression: The X-random case60. *International Statistical Review, 60*, 291-319.

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

Chai, H., & Domeniconi, C. (2004). *An Evaluation of Gene Selection Mthods for Multi-class Microarray Data Classification.* Paper presented at the The Second European Workshop on Data Mining and Text Mining for Bioinformatics, Berlin, Germany.

Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What Are Ontologies, and Why Do We Need Them? . 14, 1, 20-26.

Chen, J. (2007). Key aspects of analyzing microarray gene-expression dat. *Pharmacogenomics,* 8(5), 473-482.

D'Haeseleer, P., Liang, S., & DeRisi, J. (2000). Genetic network inference: From co-expression clustering to reverse engineering. *Bioinformatics,* 16(8), 707-716.

Galanis, E., Buckner, J., Schomberg, P., Hammack, J., Raffel, C., & Scheithauer, B. (1997). Effective chemotherapy for advanced CNS embryonal tumors in adults. *Clinical Oncology 15,* 2939-2944.

Gollub, T. R., Ball, C. A., Goldberg, D. E., & Joachims, T. (2003). The Standard microarray database: Dataset access and quality assessment tools. *Nucl. Acids Res., 31*(1), 94-96.

Greer, J., Erickson, J., Baldwin, J., & Varney, M. (1994). Application of the Three-Dimensional Structures of Protein Target Molecules in Structure-Based Drug Design. *Medicinal Chemisty 37*(8), 1035-1054.

Gromeier, M., & Wimmer, E. (2001). Viruses for the treatment of malignant glioma. *Curr. Opin. Mol. Ther., 3*(5), 503-508.

Grotzer. M. A, Janss. A. J., K.-M. Fung, J. A. Biegel, L. N. Sutton, L. B. Rorke, et al. (2000). TrkC Expression Predicts Good Clinical Outcome in Primitive Neuroectodermal Brain Tumors. *Journal of Clinical Oncology, 18*(5).

Gruber, T. R. (1993). *Toward Principle for the Design of Ontologies Used for Knowledge Sharing.* Paper presented at the International Workshop on Formal Ontology, Padova, Italy.

Gruber, T. R. (1993). A translation approach to portable ontologies *Knowledge Acquisition, 5*(2), 199-220.

Gruber, T. R. (2008). Ontology. In L. Liu & T. zsu (Eds.), *Encyclopedia of Database Systems:* Springer-Verlag.

Hartigan, J. A., & Wong, M. (1976). A K-Means Clustering Algorithm. *Applied Statistics, 28*(1), 100-108.

Herta, E. B., & Duval, B. (2006). A hybird GA/SVM approach for gene selection and classification of microarray data.

Howard, Y., Chang, J., Sneddon, A. A., Alizadeh, R. S., & Rob, B., (2004). Gene Expression Signature of Fibroblast Serum Response Predicts Human Cancer Progression: Similarities between Tumors and Wounds.

Huang, L., Song, Q., & Kasabov, N. (2008). Evolving Connectionist System Based Role Allocation for Robotic Soccer. *International Journal of Advanced Robotic Systems, 5*(1), 59-62.

Jacco, O., Lynda, H., & Rutledge. (2002). Hypermedia and the Semanticc Web: A Research Agenda. *Digital information, 3(1).*

Joachims, T. (2006). *Training linear SVMs in linear time.* Paper presented at the Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.

John. K, Mary E. Sutton, Diane J. Lu, Tracey A. Cho, Liliana C. Goumnerova, Lyuda Goritchenko, et al. (1999). Activation of Neurotrophin-3 Receptor TrkC Induces Apoptosis in Medulloblastomas. *Cancer Research, 59,* 711-719.

Jolliffe. (2002). Principal Component Analysis. In P. Bickel, P. J. Diggle, S. Fienberg, U. Gather, I. Olkin & S. Zeger (Eds.), *Statistics* (Vol. 487). NY: Springer.

Kanellopoulos, I. (1997). *Neuro-computation in Remote Sensing Data analysis.*

Kasabov, N. (1998). Evolving Fuzzy Neural Networks - Algorithms, Applications and Biological Motivation.

Kasabov, N. (2001). Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 31(6), 902-918.

Kasabov, N. (2002). *Evolving connectionist systems. methods and applications in bioinformatics, brain study and intelligent machines.* London: Springer-Verlag.

Kasabov, N. (2007). *Evolving connectionist systems. the knowledge engineering approach* (2nd ed.). New York: Springer.

Kasabov, N. (2007). Global, local and personalied modeling and pattern discovery in bioinformatics: An interated approach. *Pattern recognition Letters, 28,* 673-685.

Kasabov, N., Vishal, J., Gottgtroy, P. C. M., Benuskova, L., & Joseph, F. (2007). Brain gene ontology and simulation system (BGOS) for a better understanding of the brain. *Cybernetics and Systems,, 38*(5), 495 - 508.

Kasabov, N., Vishal, J., & Benuskova, L. (2008). Integrating evolving brain-gene ontology and connectionist-based system for modeling and knowledge discovery. *Neural networks*, 21(2-3), 266-275.

KEDRI. (2002). *NeuCom.* from www.theneucom.com

Kirwin, C. (1995). *'Reasoning', in Ted Honderich* Oxford:Oxfors University Press.

Kleihues, P., Burger, P. C., & Scheithauer, B. W. (1993). *Histological typing of tumours of the central nervous system.* New York: Springer - Verlog.

Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection.* Paper presented at the Inter national Joint Conference on Artifical Intelligence (IJCAI), Quebec, Canada.

Kotsiantis, S. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica Journal*, 248-268.

Kozak, K., M., & Kozak, K. (2004). Weighted k-Nearest Neighbour Techinques for High Throuhput Screening Data. *International Journal of Biomedical Sciences 1* (3), 155-160.

Lassila, O., & Swick, R. (1999). *Resource Description Framework (RDF) Model and Syntax Specification.* W3C. from http://www.w3.org/TR/REC-rdf-syntax.

Lefkowitz, J., Rorke, L., & Packer, R. (1988). Atypical teratoid tumours of infancy: definition of an entity. *Ann Neural*(22), 448-489.

Leung, Y., Chang, C., Hung, Y., C, & Fung, P. (2006). *Gene Selection for Brain Cancer Classification.* Paper presented at the EMBS Annual International Conference, New York , USA.

Li, W., & Yang, Y. (2002). How many genes are needed for a discriminant microarray data analysis? In S. Lin & K. Johnson (Eds.), *Methods of Microarray Data Analysis* (pp. 137-150).

Lisa, M., DeAngelis, J., & Posner, P. H. (2002). *Intracranial Tumors: Diagnosis and Treatment:* Informa Health Care.

Little, P. (2003). *Genetic Destinies:* Oxford University Press.

Madeira, S. C., & Oliveira, A. L. (2004). Biclustering Algorithms for Biological Data Analysis: A Survey *IEEE/ACM Transactions on Computational Biology and Bioinformatics,, 1*(1), 24 -45.

Marnellos, G., & Mjolsness, E. (2003). *Gene network models and neural development.*

Maton, A., Jean Hopkins, Charles William McLaughlin, Susan Johnson, Maryanna Quon Warner, David LaHart, et al. (1993). *Human Biology and Health.*

Mukherjee, S., Tamayo, P., Rogers, S., Rifkin, R., Engle, A., Campbell, C., et al. (2003). Estimating Dataset Size Requirements for Classifying DNA Microarray Data. *Computational Biology, 10*(2), 119-142.

Niijima, S., & Kuhara, S. (2005). Multiclass Molecular Cancer Classification by Kernel Subspace Methods with Effective Kernel Parameter Selection. *Journal of Bioinformatics and Computational Biology, 3*(5), 1071-1088.

Nutt Catherine L, Mani D. R, A, B. R., & Tamayo, P. (2003). Gene Expression-based Classification of Malignant Gliomas Correlates Better with Survival than Histological Classification. *Molecular Biology and Genetics, 63,* 1602-1607.

Øhrstrøm, P., Andersen, J., & Schärfe, H. (2005). What Has Happened to Ontology. In *Conceptual Structures: Common Semantics for Sharing Knowledge* (Vol. 3596/2005, pp. 425-438): Springer Berlin / Heidelberg.

Oncology (2008). *Central Nervous System - Childhood* from http://www.cancer.net/patient/Cancer+Types/Central+Nervous+System+-+Childhood

Pang, S., Havukkala, I., Hu, Y., & Kasabov, N. (2006). *Classification consistency analysis for bootstrapping gene selection.* Paper presented at the ICONIP 2006.

Petricoin, E. F., Ardekani, A. M., peter, w., & ere, t. (2002). Use of proteomic patterns in serum to identify ovarian cancer. *Lancet,* 359, 572-577.

Pisanelli, D. (2004). *Ontologies in Medicine. Amsterdam: IOS Press. The Nervous System. In: Genes and Disease.,* from http://www.ncbi.nlm .nih.gov=books=bv.fcgi?rid

Pollack, A. (2001, May 14). New Class of Cancer Drugs Shows Promise. *The New York Times.*

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., et al. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature, 415, 436-442.

Protégé. (2007). *StanfordCenter for Biomedical Informatics Research.* from http://protege.stanford.edu/overview/

Quackenbush, J. (2002). Microarray data normalization and transformation. *Nature Genetics Supplement, 32.*

Quinlan, J. (1986). Induction of decision trees. *Machine Learning 1*(1), 81-106.

Ramaswamy, S., Ross, K., Lander, E. S., & Golub, T. R. (2003). A molecular signature of metastasis in primary solid tumors. *Nature Genetics Supplement, 33,* 49-54.

Rhodes, D., Yu, J., Shanker, K., & Deshpande, N. (2005). ONCOMINE: a cancer microarray database and integrated data-mining platform.

Richard , T. (2000). *The Brain: An Introduction to Neuroscience:* Worth Publishers.

Rifkin, R., & Aldebaro, K. (2004). In Defense of One-Vs-All Classification. *Journal of Machine Learning Research,* 101-141.

Rojas, R. (1996). *Neural Networks -A Systematic Introduction* (first ed.). New York: Springer-Verlag.

Rorke, L., B, Packer, R., & Biegel, J. (1995). Central nervous system atypical teratoid/rhabdoid tumors of infancy and childhood *Journal of Neuro-Oncology,* 24(1), 21-28.

Rosenblastt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review,* 65(6), 386-408.

Schena, M. (2002). *Microarray analysis.* New York: John Wiley & Sons.

Schichl, H. (2000). *Models and history of modeling.* from http://www.mat.univie.ac.at/ herman/papers/modtheoc.pdf

Scholarpedia. (2008). *K-nearest neighbor.* from http://www.scholarpedia.org/article/K-nearest_neighbor

Serruya, M. D., Hatsopoulos, N. G., Paninski, L., Fellows, M. R., & Donoghue, J. P. (2002). Instant neural control of a movement signal. *Nature*, 416, 141-142.

Slonim, D., Tamayo, P., Mesirov, J., Golub, T., & Lander, E. (2000). *Class prediction and discovery using gene expression data.* Paper presented at the Proceedings of

the 4th Annual International Conference on Computational Molecular Biology, Tokyo, Japan.

Tan, A. C., & Gilbert, D. (2003). Ensemble machine learning on gene expression data for cancer classification. *Applied Bioinformatics* 75-83.

Tang, E., Suganthan, P., & Yao, X. (2006). Gene selection algorithns for microarray data based ib least squares support vector machine. *BMC Bioinformatics.*

Taylor, C. (2003). An Introduction to Metadata (Publication no. 00025B). from Manager, Information Access Service University of Queensland Library: http://www.library.uq.edu.au/papers/ctmeta4.html

Ultsch, A. (2007). *Emergence in Self-Organizing Feature Maps.* Paper presented at the In Proceedings Workshop on Self-Organizing Maps (Bielefeld, Germany.

Vapnik, V. (1992). *The Nature of Statistical Learning Theory.* New Jersey: Springer.

Veer, L. J., & Dai, H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature, 415*(530-536).

Versteege, I. (1998). Truncating mutation of hSNF5/INIl in aggressive paediatric cancer. *Nature,* 394, 203-206.

W3C. (2008). *W3C Semantic Web Activity.* from http://www.w3.org/2001/sw/#pub

Warwick, C. (1997). *Metadata: an overview.* from http://www.nla.gov.au/nla/staff paper/cathro3.html

Wild, C., & Seber, G. (2000). *Chance Encounters: A First Course in Data Analysis and Inference.*

William, S. (2001). Modelling as a Discipline. *General System 30(3),* 261-282.

Winsberg, E. (1999). *Sanctioning Models: The Epistemology of Simulation.* Paper presented at the Modeling and Simulation, Sismondo, Sergio and Snait Gissis.

Zhu, W., & Wang, X. (2003). Detection of cancer-specific markers amid massive mass spectral data. *PNAS, 100*, 14666-14671.