GENETIC ALGORITHM-NEURAL NETWORK: FEATURE EXTRACTION FOR BIOINFORMATICS DATA

DONG LING TONG

A thesis submitted in partial fulfilment of the requirements of Bournemouth University for the degree of Doctor of Philosophy

July 2010

Copyright

Abstract

With the advance of gene expression data in the bioinformatics field, the questions which frequently arise, for both computer and medical scientists, are which genes are significantly involved in discriminating cancer classes and which genes are significant with respect to a specific cancer pathology.

Numerous computational analysis models have been developed to identify informative genes from the microarray data, however, the integrity of the reported genes is still uncertain. This is mainly due to the misconception of the objectives of microarray study. Furthermore, the application of various preprocessing techniques in the microarray data has jeopardised the quality of the microarray data. As a result, the integrity of the findings has been compromised by the improper use of techniques and the ill-conceived objectives of the study.

This research proposes an innovative hybridised model based on genetic algorithms (GAs) and artificial neural networks (ANNs), to extract the highly differentially expressed genes for a specific cancer pathology. The proposed method can efficiently extract the informative genes from the original data set and this has reduced the gene variability errors incurred by the preprocessing techniques.

The novelty of the research comes from two perspectives. Firstly, the research emphasises on extracting informative features from a high dimensional and highly complex data set, rather than to improve classification results. Secondly, the use of ANN to compute the fitness function of GA which is rare in the context of feature extraction.

Two benchmark microarray data have been taken to research the prominent genes expressed in the tumour development and the results show that the genes respond to different stages of tumourigenesis (i.e. different fitness precision levels) which may be useful for early malignancy detection. The extraction ability of the proposed model is validated based on the expected results in the synthetic data sets. In addition, two bioassay data have been used to examine the efficiency of the proposed model to extract significant features from the large, imbalanced and multiple data representation bioassay data.

Publications Resulting From Thesis

1. Microarray Gene Recognition Using Multiobjetive Evolutionary Techniques *(Poster)*.
   By D.L. Tong and R. Mintram.
   In *RECOMB'08: 12th Annual International Conference on Research in Computational Molecular Biology*, Poster Id: 74, 2008.

2. Hybridising genetic algorithm-neural network (GANN) in marker genes detection *(Proceedings)*.
   By D.L. Tong.
   In *ICMLC'09: 8th International Conference on Machine Learning and Cybernetics, proceedings*, volume 2, pages 1082-1087, 2009.

3. Innovative Hybridisation of Genetic Algorithms and Neural Networks in Detecting Marker Genes for Leukaemia Cancer *(Supplementary Proceedings)*.
   By D.L. Tong, K. Phalp, A. Schierz and R. Mintram.
   In *PRIB'09: 4th IAPR International Conference on Pattern Recognition in Bioinformatics, suppl. proceedings*, 2009.

4. Innovative Hybridisation of Genetic Algorithms and Neural Networks in Detecting Marker Genes for Leukaemia Cancer *(Poster)*.
   By D.L. Tong, K. Phalp, A. Schierz and R. Mintram.
   In *PRIB'09: 4th IAPR International Conference on Pattern Recognition in Bioinformatics*, Poster Id: 2, 2009.

5. Extracting informative genes from unprocessed microarray data *(Proceedings)*.
   By D.L. Tong.
   In *ICMLC'10: 9th International Conference on Machine Learning and Cybernetics, proceedings*, 2010.

6. Genetic Algorithm Neural Network (GANN): A study of neural network activation functions and depth of genetic algorithm search applied to feature selection *(Journal)*.

   By D.L. Tong and R. Mintram.

   In the *International Journal of Machine Learning and Cybernetics (IJMLC)*, *in press* 2010.

7. Genetic Algorithm Neural Network (GANN): Feature Extraction for Unprocessed Microarray Data.

   By D.L. Tong and A. Schierz.

   Submitted to the Journal of Artificial Intelligence in Medicine on April 2010.

Abbreviations

Biology

ALL - Acute Lymphoblastic Leukaemia

AML - Acute Myelogeneous Leukaemia

APL - Acute Promyelocytic Leukaemia

BL - Burkitt Lymphoma

DDBJ - DNA Data Bank of Japan

DNA - Deoxyribonucleic Acid

cDNA - complementary DNA

CML - Chronic Myelogeneous Leukaemia

EBI - European Bioinformatics Institute

EMBL - European Molecular Biology Laboratory

EST - Expressed Sequence Tag

EWS - Ewing's Sarcoma

FAB - French-American-British

FISH - Fluorescent In-Situ Hybridisation

FPR - Formylpeptide Receptor

GO - Gene Ontology

ISCN - International System for Human Cytogenetic Nomenclature

LIMS - Laboratory Information Management System

MGED - Microarray Gene Expression Data

MIAME - Minimum Information About a Microarray Experiment

MLL - Mixed Lineage Leukaemia

MPSS - Massive Parallel Signature Sequencing

NB - Neuroblastoma

NCBI - National Center for Biotechnology Information

NHGRI - National Human Genome Research Institute

NIG - National Institute of Genetics

PCR - Polymerase Chain Reaction

RT-PCR - Reverse Transcription-PCR

PNET - Primitive Neuroectodermal Tumour

PMT - Photo-Multiplier Tube

RMS - Rhabdomyosarcoma

ARMS - Alveolar RMS

ERMS - Embryonal RMS

PRMS - Pleomorphic RMS

RNA - Ribonucleic Acid

mRNA - messenger RNA

tRNA - transfer RNA

SAGE - Serial Analysis of Gene Expression

SNP - Single Nucleotide Polymorphism

SRBCTs - Small Round Blue Cell Tumours

Computing

ANN - Artificial Neural Network

AP - All-Pairs

BGA - Between-Group Analysis

BIC - Bayesian Information Criterion

BSS/WSS - Between-Group and Within-Group

CART - Classification And Regression Tree

COA - Correspondence Analysis

DAC - Divide-And-Conquer

DT - Decision Tree

EA - Evolutionary Algorithm

FDA - Fisher's Discriminant Analysis

FS - Feature Selection

GA - Genetic Algorithm

GANN - Genetic Algorithm-Neural Network

GP - Genetic Programming

HC - Hierarchical Clustering

IG - Information Gain

KNN - k-Nearest Neighbour

LDA - Linear Discriminant Analysis

LOOCV - Leave-One-Out Cross-Validation

LRM - Logistic Regression Model

MDS - Multidimensional Scaling

NB - Naive Bayes

NSC - Nearest Shrunken Centroid

OBD - Optimal Brain Damage

OVA - One-Versus-All

PAM - Prediction Analysis of Microarrays

PART - Projective Adaptive Resonance Theory

PCA - Principal Component Analysis

PLS - Partial Least Squares

RA - Relief Algorithm

RFE - Recursive Feature Elimination

S2N - Signal-to-Noise

SA - Simulated Annealing

SAC - Separate-And-Conquer

SOM - Self-Organising Map

SVD - Singular Value Decomposition

SVM - Support Vector Machine

WEKA - Waikato Environment for Knowledge Analysis

WV - Weighted Voting

Table of Contents

List of Figures

List of Tables

Acknowledgements

This thesis would not have been possible without the assistance and support of many great people. My greatest gratitude goes to Dr. Robert Mintram, my former research supervisor. His insightful guidance, persisting support and encouragement have given me the greatest experience. Thanks also go to Dr. Amanda Schierz, my thesis supervisor and Prof. Mark Hadfield. Dr. Schierz's constructive advices and inspiring discussions have made a lot of improvement to this work.

I am also grateful to all the staff in the School of Design, Engineering and Computing and Graduate School for helping and supporting in any form. I also thank to all my friends and colleagues who have supported me and helped me. Special thanks to Mrs. Dongxia Wang who look after me during my injury period and Mrs. Margaret Lofthouse who proof-read my thesis.

I would like to acknowledge the financial support of the ORSAS. I give my deepest love and appreciation to my mother, brother and sister for their unconditional love and support. This thesis is dedicated to the memory of my dearest father, whose constant inspiration showed me how to be strong during the hardest of times.

Declarations

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning.

Chapter 1

Introduction

Cancer is a disease caused by abnormal cell growth. It is the second leading cause of death in developed countries and is in the top three causes of death in developing countries (Hayter, 2003). Based on the survey carried out by the World Health Organisation (WHO), deaths from cancer worldwide is projected to continue rising, from 7.4 million deaths in year 2004 to an estimate 12 million deaths in year 2030 (WHO, 2010). The use of microarray gene expression data to diagnose cancer patients has increased dramatically over the past decade, indicating an urgent need for the development of treatment measures for the potential genetic causes of disease.

Microarray experiment is a biological procedure to measure the activities of genes at a specific time frame applied to a subject, i.e. pre-cancer screening, general health check and cancer remission check. It is designed for bioinformatics field to provide an insight for information on the gene interactions and cancer pathways with a potential for cancer diagnosis and prognosis, prediction of therapeutic responsiveness, discovery of new cancer groups and molecular marker identification (Golub et al., 1999; Dupuy and Simon, 2007; Yu et al., 2007; Wang et al., 2008). Microarray experiment contains measurements for thousands of microscopic spot of DNA probes (i.e. DNA spots that have been complimentarily binded in the microarray experiment), however, only a small set of these probes are relevant to the subject of interest, for example, amongst 7129 probes in the leukaemia microarray data available from the Broad Institute, only about 1000 probes are relevant to the leukaemogenesis pathway (Golub et al., 1999). Therefore, techniques for extracting the informative genes that underlies the pathogenesis of tumour cell proliferation, from high dimensional microarrays is necessary (Yu et al., 2007; Osareh and Shadgar, 2008; Wang et al., 2008; Zhang et al., 2008) and the need for computing algorithms to undertake such a complex task emerge naturally. This brings the theme of computational analysis in microarray studies to the forefront of research.

Microarray gene expression data is characterised by high feature dimensionality, sample scarcity and complex

gene behaviour (i.e. the interaction between genes within the data), which pose unique challenges in the development of computing algorithms in class prediction, cluster discovery and marker identification, with the aim of deriving a biological interpretation of the set of genes which underlies the cause of the disease. In addition, microarray gene expression data may contain subgroup of cancer classes within a known class, for example, the leukaemia microarray data in Figure 1.1 on page 4 contains two subgroup of cancer classes, i.e. B-cell ALL and T-cell ALL, within a known cancer group called ALL. This makes the analysis of microarray difficult. Thus, the first and foremost consideration for analysing microarray data, is feature extraction. For class prediction, the extracted gene subset is used to avoid the over-fitting problem on supervised classifiers and to achieve better predictive accuracy that generalises well to unknown data (Wang et al., 2008). For unsupervised cluster discovery, the extracted gene subset is essential for discerning the underlying cluster grouping tendency in a lower dimension and to prevent false cluster formation (Wang et al., 2008). For molecular marker identification, the extracted gene subset provides a smaller feature search space with high potential true cancer markers and thus, reduces computational cost on performing an exhaustive search over the full feature space.

The goal of this research is to devise a more effective way to extract features with highly important information to a specific disease, i.e. informative features, using genetic algorithms (GAs) and artificial neural networks (ANNs) due to their learning abilities to construct hypotheses that can explain complex relationships in the data (Nanni and Lumini, 2007). This research explores the effectiveness of a genetic algorithm-neural network (GANN) hybrid, in analysing gene expression activities, based on a specific tumour disease and identifying the informative genes that underlie different precision levels in the extraction process. The identified gene subset may give an enhanced insight on the gene-gene interaction in response to different stages of abnormal cell growth which could be vital in designing treatment strategies to prevent any progression of abnormal cells.

This chapter provides the motivation of this research and an overview of our work, including the existing problems in the field, our approach to the problem and our contributions to the field.

1.1 Motivation

The advances of microarray technologies to measure gene expression levels in a global fashion have significantly improve the accuracy of morphological and clinical-based diagnosis results (Lu and Han, 2003). It also produces high dimensional noisy data (i.e. features which are not associated or least important to the subject of interest) during the microarray production and, in most cases, it contains multiple cancer subclasses within a known cancer class (see Figure 1.1). Numerous biology analysis methods have been

introduced to study the gene-gene interaction and the functionality of genes. These methods include serial analysis of gene expression (SAGE) (Velculescu et al., 1995, 1997, 2000), massive parallel signature sequencing (MPSS) (Brenner et al., 2000) and mass spectrometric analysis (Pandey and Mann, 2000). However, the lack of standardisation on the gene probes (Asyali et al., 2006) and the gene annotations due to the rapid development of microarray technology, plus, the variability on gene expression measurements based on similar arrays from different research laboratories, makes the integration of microarray results impossible. In addition, the relationships between genes have complicated the finding of marker genes. As a result, the need for computational analysis of microarray gene expression is required.

Frequently, data preprocessing is required on microarray data to remove undesirable data characteristics with the idea of ensuring data integrity and improving classification performance. For instance, missing values in microarrays require some mathematical formulas to impute reasonable estimates to salvage the data. Feature reduction is the approach most commonly used to remove data redundancies. Data normalisation is generally expected to scale down the magnitudes of data values prior to computational analysis, such as prediction; rather than to scale up the magnitudes of data values. Numerous normalisation techniques and feature reduction approaches have been reported in the literature, such as standardisation with mean and variance values of the data (Golub et al., 1999; Dudoit et al., 2002; Yu et al., 2007; Cheng and Li, 2008), scaling with maximum and minimum values (Cho and Won, 2007; Gonia et al., 2008), logarithmic transformation (Dudoit et al., 2000; Zhou et al., 2005; Chen et al., 2007) and filtering (Bø and Jonassen, 2002; Dudoit et al., 2002; Futschik et al., 2003; Liu et al., 2004a; Ross et al., 2004; Chu et al., 2005; Jirapech-Umpai and Aitken, 2005; Lee et al., 2005). Consequently, different sets of identified genes were reported.

Existing research emphasises effective classification predictiveness (Khan et al., 2001; Dudoit et al., 2002; Cho et al., 2003b; Lee and Lee, 2003; Bloom et al., 2004; Liu et al., 2004a,c; Lee et al., 2005; Yu et al., 2007; Osareh and Shadgar, 2008; Zhang et al., 2008) and cluster discovery (Ross et al., 2000; Wang et al., 2003). This research under-estimated the complexity of microarray data and overlooked the 'true' objective of microarray studies, i.e. to extract molecular-based informative genes underlying the pathogenesis of tumour development. Figure 1.1 shows example of some gene expression values extracted from the leukaemia oligonucleotide Affymetrix chips. Generally, the oligonucleotide microarray preserve the exact measurement of the expressed genes under the fluorescence labelling process in the microarray experiment. We will review the microarray experiment in Chapter 2.

From the microarray perspective, the leukaemia data set shown in Figure 1.1 represents two distinct types of leukaemia cancer: ALL and AML, as is indicated by columns in the figure. Most expressed genes, denote in rows in the figure, were inhibitory (negative expression values), i.e. over-suppressed to the leukaemia cancer. This means that there is a high amount of trivial information in the data set which has no significant

contribution to the leukaemogenesis pathology. However, from the computing perspective, the leukaemia data set shown in Figure 1.1 contains four different classes, rather than only 2 classes, as the ALL_B-cell and the ALL_T-cell are considered as separate classes where, in fact, they are just a variant of ALL cancer from the medical perspective. Furthermore, the data set is highly skewed as most data values are in negatives, which is not commonly expected in standard classification problems. As a result, most microarray data, especially the oligonucleotide microarray, has to be preprocessed to remove any incompatibility values for effective classification results which, in fact, will yield the variation in the gene subset selection from similar microarray data. This shows the inefficacy of the classification methods to analyse microarray data.

File Edit View Insert Format Tools Data Window Help

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1	Gene Desc	Gene Acce	ALL-1_B-	call	ALL-2_T	call	ALL-3_T	call	ALL-4_B-	call	ALL-5_B-	call	ALL-6_T-	call	ALL-9_T	call	ALL-10_T	call	ALL-11_T	call	ALL-14_T	call	ALL-15_B	call	ALL-47	call	ALL-42	call	AML-28	call	AML-29	call
2	AFFX-Bio	AFFX-Bio	-214	A	-139	A	-76	A	-135	A	-106	A	-138	A	5	A	-88	A	-165	A	-113	A	-107	A	-476	A	-81	A	-318	A	-32	A
3	AFFX-Bio	AFFX-Bio	-153	A	-73	A	-49	A	-114	A	-125	A	-85	A	-127	A	-105	A	-155	A	-147	A	-72	A	-213	A	-150	A	-192	A	-49	A
4	AFFX-Bio	AFFX-Bio	-58	A	-1	A	-307	A	265	A	-76	A	215	A	106	A	42	A	-71	A	-118	A	-126	A	-18	A	-119	A	-95	A	49	A
5	AFFX-Bio	AFFX-Bio	88	A	283	A	309	A	12	A	168	A	71	A	268	A	219	M	82	A	243	M	149	A	301	A	78	A	312	A	230	P
6	AFFX-Bio	AFFX-Bio	-295	A	-264	A	-376	A	-419	A	-230	A	-272	A	-210	A	-178	A	-163	A	-127	A	-205	A	-403	A	-152	A	-139	A	-367	A
7	AFFX-Bio	AFFX-Bio	-558	A	-400	A	-650	A	-585	A	-284	A	-558	A	-535	A	-246	A	-430	A	-398	A	-284	A	-394	A	-340	A	-344	A	-508	A
8	AFFX-Bio	AFFX-Bio	199	A	-330	A	33	A	158	A	4	A	67	A	0	A	328	A	100	A	-249	A	-166	A	-42	A	-36	A	324	A	-349	A
9	AFFX-Cre	AFFX-Cre	-176	A	-168	A	-367	A	-253	A	-122	A	-186	A	-174	A	-148	A	-109	A	-228	A	-185	A	-144	A	-141	A	-237	A	-194	A
10	AFFX-Cre	AFFX-Cre	252	A	101	A	206	A	49	A	70	A	87	A	24	A	177	A	56	A	-37	A	1	A	98	A	96	A	105	A	34	A
11	AFFX-Bio	AFFX-Bio	206	A	74	A	-215	A	31	A	252	A	193	A	506	A	183	A	350	A	113	A	-23	A	173	A	-55	A	167	A	-56	A
12	AFFX-Bio	AFFX-Bio	-41	A	19	A	19	A	363	A	155	A	325	A	284	A	-143	A	204	A	188	A	205	A	-133	A	-209	A	-50	A	147	A
13	AFFX-Bio	AFFX-Bio	-831	A	-743	A	-1135	A	-934	A	-471	A	-631	A	-829	A	-684	A	-524	A	-625	A	-437	A	-959	A	-362	A	-820	A	-841	A
14	AFFX-Bio	AFFX-Bio	-653	A	-239	A	-962	A	-577	A	-490	A	-625	A	-844	A	-468	A	-599	A	-413	A	-351	A	-271	A	-427	A	-231	A	-657	A
15	AFFX-Bio	AFFX-Bio	-462	A	-83	A	-232	A	-214	A	-184	A	-177	A	-230	A	-51	A	-188	A	-229	A	-210	A	-228	A	-4	A	-273	A	-240	A
16	AFFX-Bio	AFFX-Bio	75	A	182	A	208	A	142	A	32	A	-94	A	292	A	233	A	34	A	0	A	39	A	-42	A	199	A	208	A	-15	A
17	Dynamin-l	AF000430	7	A	-5	A	5	A	3	A	112	P	-51	A	-52	A	-71	A	-13	A	15	A	82	A	4	A	-4	A	84	A	-47	A
18	GB DEF =	AF000545	-273	A	-71	P	-191	A	-269	A	-48	A	-243	A	-129	A	-152	A	-29	P	-237	A	79	M	-149	P	24	P	84	A	-651	A
19	TTF-I inte	AF000560	457	A	-45	A	376	A	325	A	221	A	280	A	296	A	-150	A	127	A	489	A	75	A	-13	A	342	A	289	P	405	A
20	Uroplakin	AF000562	1002	A	1152	A	953	A	1012	A	885	A	901	A	976	A	967	A	1008	A	811	A	716	A	2046	A	817	A	1397	A	1018	A
21	Homogenti	AF000573	-62	A	-83	A	-91	A	-172	A	-40	A	-49	A	-45	A	8	A	-69	A	-61	A	-70	A	-184	A	-43	A	-138	A	-17	A
22	Transmemb	AF000959	-468	A	-829	A	-496	A	-739	A	-55	A	-542	A	-451	A	-425	A	-235	A	89	A	-141	A	-678	A	-22	A	-309	A	-695	A
23	IPL (IPL) r	AF001294	-154	A	-50	A	-177	A	193	A	-71	A	104	A	-217	A	62	A	31	A	60	A	-97	A	232	A	30	A	441	A	311	A

Figure 1.1: The gene expression values extracted from the leukaemia oligonucleotide Affymetrix chip. Each column represent a biological sample and each row corresponds to a gene in the sample. From the microarray perspective, this data shows 2 distinct types of leukaemia cancer: ALL and AML, and most expressed genes were over-suppressed, indicating that there is a high amount of trivial information in the data set which has no significant association to the cancer pathology. From the computing perspective, this data contains 4 different classes: ALL_B-cell, ALL_T-cell, ALL and AML, and the data is highly skewed as most data values being negatives.

Classification is merely a mathematical validation mechanism for assessing the significance of identified genes in perfectly discriminated cancer groups, but it does not have the ability to assess the correlation of genes at a genomic level. Most classification methods suffer generalisation problems. Some classifiers are sensitive to data distribution, for instance, a neural network classifier (ANN) generally perform well with normalised data and a naive bayes (NB) classifier inferiors when the number of features is larger than the number of samples (Asyali et al., 2006). Some classifiers are sensitive to the fitness of the model, for instance, a weighted voting (WV) classifier performs well on binary classification but its performance decreases when working on a multiclass problem (Golub et al., 1999). As a result, classifiers are only effective for certain data sets and for data sets which contain no relationship between features in different classes, and consequently, the reliability of the reported feature subset becomes inexplicit, due to the little attention that has been made for improving feature extraction technique.

Typical microarray data contains thousands of genes (i.e. *curse of dimensionality*) extracted from a few

samples (i.e. *curse of data sparsity*) which are possibly obtained from the same arrays (source) because of the high processing cost of microarrays, and, most of the genes in the microarray data are inter-related, as shown in Figure 1.2. This complicates the process of finding informative genes. Numerous feature selection techniques have been developed to extract informative genes, however, the core of the study is focused on effective classification. For instance, Golub et al. (1999) introduced a signal-to-noise (S2N) ratio to improve the classification performance of WV classifier in discriminating two dominant groups of acute leukaemia; Khan et al. (2001) used gene signatures, identified by principal component analysis (PCA), to classify four types of SRBCTs tumours using ANN classifiers and Tibshirani et al. (2002) developed a nearest shrunken centroid (NSC) algorithm with respect to the prediction analysis of microarray (PAM) classifier. Albeit, encouraging results had been achieved in most hybrid selection/classification models, the functionality of the reported genes is inconclusive due to an ill-conceived hypothesis based on classification performance. For instance, the S2N ratio calculates the correlation between individual genes based on the mean and the standard deviation of the gene for the samples. This is not feasible for gene expression data as it omits the correlation between a combination of genes. For instance, the ALL class in the leukaemia microarray data in Figure 1.1 showed that there is more than one variant of ALL leukaemias, i.e. B-cell ALL and T-cell ALL. Although, they are formed by different leukaemia cells, however, they shared some commonality in certain genetic behaviour, i.e. lymphoblastic-based, which could be used as the signature markers in diffentiating ALL patients from non-ALL cancer patients. Figure 1.2 shows the correlation between some expressed genes in the leukaemia data which is presented in Figure 1.1.

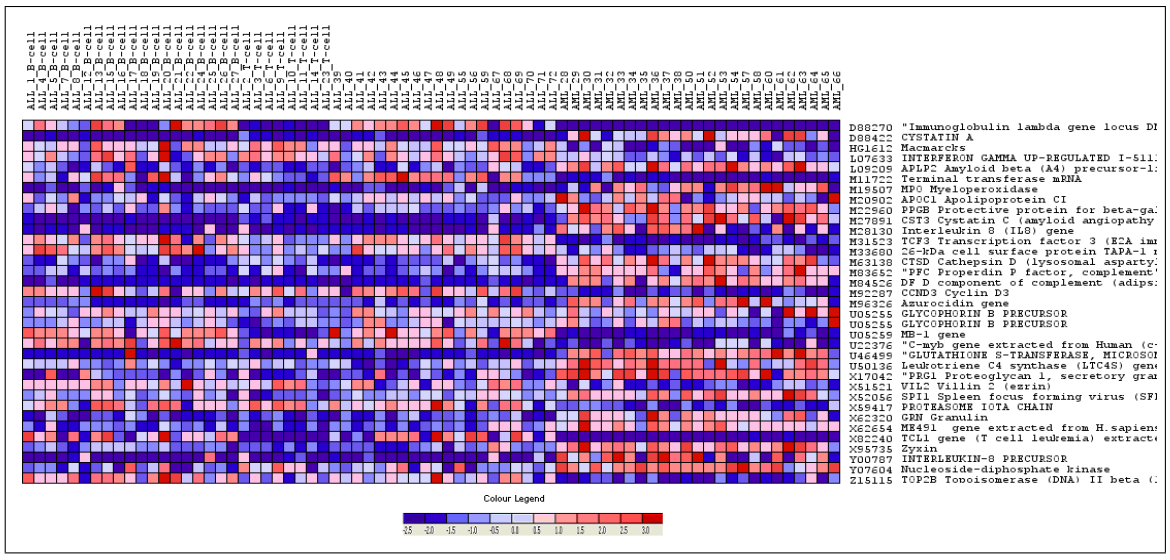


Figure 1.2: The heatmap of the leukaemia microarray data. Each column represent a biological sample and each row corresponds to a gene in the sample. The density of the significant genes to the sample is presented with the shade of two colours: red and blue. Shades of red indicate elevated expression (i.e. highly significant to the sample) while shades of blue indicate decreased expression (i.e. zero significant to the sample).

Furthermore, the implementation of more than one feature selection for the classification method may lead

to different gene selection results due to over-complication in the model structure which could resulted in model over-fit.

Over-fitting/under-fitting is a potentially serious problem in most computing algorithms, especially in the classification methods. It occurs when the algorithm is learned for too long or too little. Normally, this problem can be alleviated by continually monitoring the quality of training using a separate set of data. However, there is no standard validation mechanism for assessing the algorithm's performance. Some studies validate the algorithm using a separate set of test data (Golub et al., 1999; Deutsch, 2001; Li et al., 2001a; Hwang et al., 2002; Cho and Won, 2003; Liu et al., 2005b; Cho and Won, 2007; Zhang et al., 2008). Some employed k-fold cross-validation procedures (Tibshirani et al., 2002; Cho et al., 2003b; Guan and Zhao, 2005; Osareh and Shadgar, 2008) to assess the performance of the algorithm. Some utilised leave-one-out cross-validation procedures (Bø and Jonassen, 2002; Peng et al., 2003; Zhou and Mao, 2005; Zhou et al., 2005; Chen et al., 2007) in the performance assessment. The over-fitting/under-fitting problem can also arise when the algorithm has too many or too few parameters to learn, and consequently, its generalisation ability may be inferior (Asyali et al., 2006).

The performance of a computing algorithm is usually defined based on a standard hypothesis that works effectively in most real-world problems. This hypothesis is based on classification performance, i.e. '*the higher the classification accuracy obtained by the classifier, the better the solution to the problem*'. However, this hypothesis is not always correct in interpreting the gene correlation on microarray studies. Unlike ordinary real-world data which has small levels of interaction between features, such as financial data, bioassay data, intrusion data; microarrays has high complexity, as is depicted in Figures 1.1 and 1.2. The genes in microarray data sets are all correlated either in a direct manner, for example, a high regulated gene activates another gene with high expression, or in an indirect manner, for example, expression of a gene is triggered by the detection of another genes. The p53 protein will only be activated by the presence of the p53 gene (either highly expressed or has been detected) that contributes to the transformation and malignancy of cells due to the failure to bind the consensus DNA binding site. The p53 protein is used to co-ordinate the repair process of cells or induce cell suicide to stop any further growth of cancer cells. These correlated genes may not be detected in high classification accuracy if the high individual correlated genes are not presented or are "buried" by other more highly expressed genes.

There is an omission in the work of finding genes expressed in lower precision level, or, does not mention it at all, due to the ill-conceived hypothesis and implicit research objective that was biased effective classification. These genes, to some extent, may be important for early malignancy detection as some genes will only become significant with the presence of its correlated genes which could be detected in lower precision levels.

The possible reasons for the immaturity of the computational analysis of microarrays is due to little un-

derstanding of the complex relationship between genes and the importance of gene integration within a cell or a tissue. As a result, the general hypothesis that worked effectively in ordinary real-world problems is assumed to be effective for microarray data.

To conclude, our research is motivated by such challenges poses to the use of computational analysis on gene expression aspects exposing several weaknesses, such as the implicit research objective and the ill-conceived hypothesis, the normalisation of microarray data, the over-fitting problem and the omission of genes expressed in lower precision levels.

1.2 Statement of the Problem

As mentioned in the previous section, research in microarray gene extraction is still inconclusive. Thus, several problems on identifying informative genes have been exposed. This thesis concentrates on four main aspects which are as follows:

1.2.1 Implicit Research Objective and Ill-conceived Hypothesis

Most research emphasises the effective classification and overlooks feature extraction. The analysis models that were constructed based on the hypothesis emphasising the classification ability, which have been shown to be successful in improving classification performance, but suffers from the problems of model fitness and data distribution, as described in previous section. For instance, a conventional discriminant analysis model requires more sample patterns than features (Culhane et al., 2002) to deliver high classification results. Conversely, microarray data sets contain only a few sample patterns that are associated with thousands of genes. Thus, the use of data preprocessing techniques and appropriate feature selection methods to circumvent the problem are common solutions. However, such gene selections might involve arbitrary selection criterion and overlook highly informative combinations of genes (Culhane et al., 2002). This is due to microarray data containing more than one variant of cancer groups within a known cancer class, as is shown in Figure 1.1 and a high correlation between genes expressed to a specific cancer disease, as is indicated in Figure 1.2. Using the data preprocessing techniques, such as data normalisation, filtering and data imputation, a potential consequence is that the features may end up equalised and what was originally a primary feature may become of equal significance as secondary and less significant features. Furthermore, the primary features may be removed in the filtering process and the features interactions may be altered by improper impute values into some features. Thus, the lack of understanding of microarray data could, possibly, lead to the improper research objectives outlined.

1.2.2 Data Normalisation

There is often a large difference between the maximum and the minimum values within a gene in microarray data, especially in oligonucleotide arrays, as is indicated in Figure 1.1. Some may be due to outliers, i.e. values that are greatly different from the other values in the same gene, or missing values in the data. Thus, data normalisation is usually expected to remove undesirable characteristics in microarray data to ensure data integrity and better classification performance (Dudoit et al., 2002; Asyali et al., 2006; Kotsiantis et al., 2006) rather than discovering correlated features. Normalisation, in the context of this thesis, is a scaling process that reduces the magnitude of data values to a specific range and the degree of scaling is reliant on the mathematical formulas applied. Data normalisation is normally expected in a classification problem, as it is a very effective way of removing unwanted features from the data set, in particular when the features are not correlated. In microarray data, due to the complex biological interaction between expressed genes, normalisation is not always versatile. Conversely, it may deteriorate the finding of the correlated genes, in response to the labelled classes, by compressing the intensity of expressed genes to minimal. As a result, the correlated genes expressed in a lower expression level may be ignored. Furthermore, different types of normalisation technique may also produce different set of data values on the similar data set. Table 1.1 shows some examples of the relevant work on the leukaemia microarray data involving data preprocessing as shown in Figure 1.1.

Table 1.1: Some examples of the work related to the leukaemia microarray data.

Author	Data preprocessing	Selection method	Classification method
Golub et al. (1999)	Mean and deviation normalisation	S2N ratio	WV
Culhane et al. (2002)	for COA: negative values transformation; for PCA: mean and deviation normalisation	COA, PCA	BGA
Dudoit et al. (2002)	thresholding, filtering, log-transformation, mean and variance normalisation	BSS/WSS ratio	various discrimination methods
Li and Yang (2002)	Log transformation	Stepwise selection	LRM
Lee and Lee (2003)	as similar to Dudoit et al. (2002)	BSS/WSS ratio	SVMs
Mao et al. (2005)	as similar to Dudoit et al. (2002)	RFE	SVMs
Cho and Won (2007)	Max-min normalisation	Pearson correlation	ensemble ANNs

Dudoit et al. (2002); Mao et al. (2005) and Lee and Lee (2003) conducted a comprehensive preprocessing step comprising values truncation, genes ratio filtering and log (base-10) transformation on the data set before it has been standardised using zero mean and unit variance. Consequently, the integrity of the gene selection results have been compromised by these over-compressed techniques. Golub et al. (1999), on the other hand, standardised the similar data set using different mathematical formula. Instead of using the variance parameter (i.e. the square of the standard deviation), Golub et al. used standard deviation function. Cho and Won (2007) adopted maximum-minimum function to scale down the magnitude of the data set into the interval [0,1]. Li and Yang (2002), however, used only the log transformation to scale down the data values in the data set. Culhane et al. (2002) converted all the negative expression values in the data set to positive values before performing normalisation technique. This has, in fact, altered the context of the expression values in the genes, i.e. from the original inhibitory became excitatory.

1.2.3 Over-fitting Problem

Over-fitting normally arises when the algorithm has learned too much due to several factors, such as an over-parameterised model structure, too many repetition assessments in the algorithm and the complexion of the algorithm. A typical example is the use of an external feature reduction method to filter the redundant features and then analyse the remaining features with different selection approaches that are embedded in the classification technique. As shown in Table 1.1, Dudoit et al. (2002); Mao et al. (2005) and Lee and Lee (2003) used a normalised matrix of intensity values to filter the least significant genes of the leukaemia data set before the selection method was applied. As a result, the correlated genes may have been discarded in the filtering process and over-optimistic classification results were reported.

1.2.4 Omission on Features expressed in Lower Precision Level

In existing bioinformatics literature, the reported gene selection results are based on the optimum classification accuracy. Thus, the correlated genes expressed in the lower accuracy have not received any attention as these genes do not possess a predictive benefit in a classification result. This may lead to disregarding the 'true' underlying genes responsible for the early stage of a cell abnormality. A possible approach to solving this problem is to monitor differentiation of the genes expressed in different precision levels. The main advantage of this approach is to provide a concrete formulation on the reported genes.

The outline of our approach to solve these problems is presented in the next section.

1.3 GANN: Feature Extraction Approach

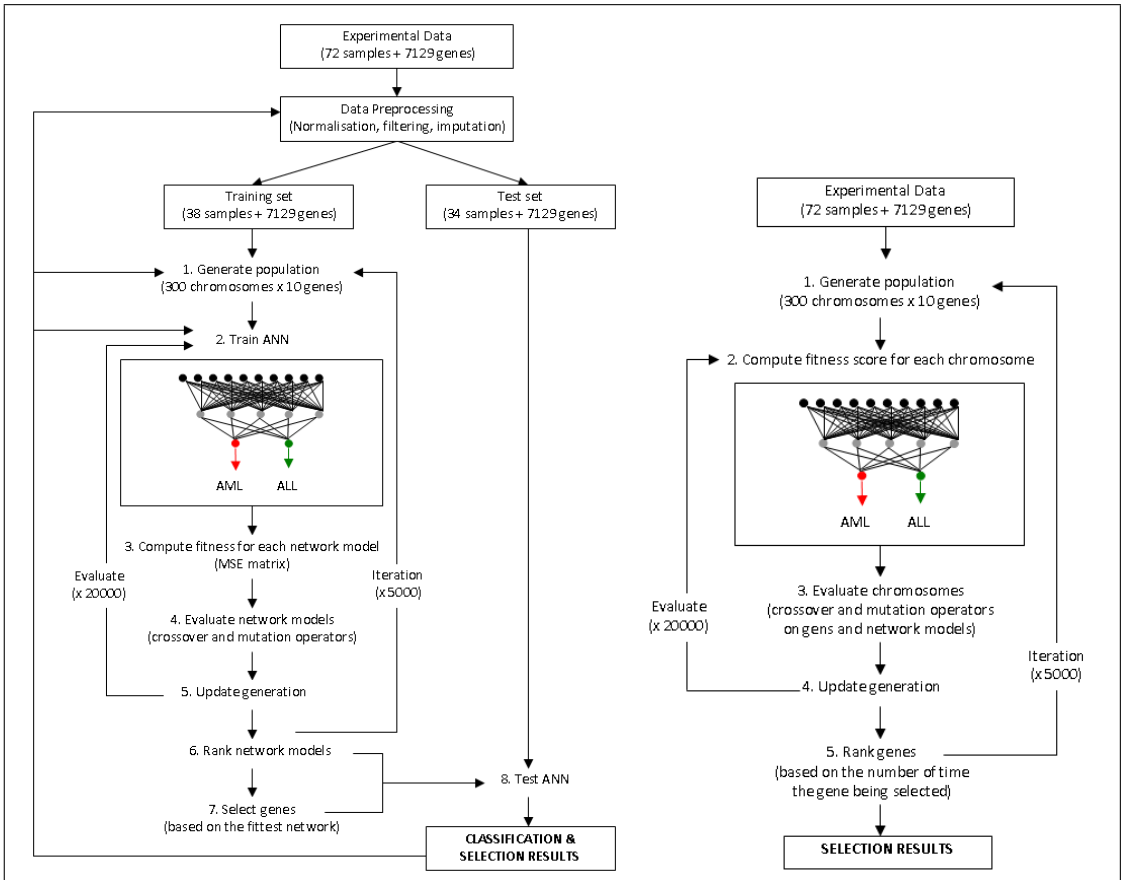
This thesis focuses on extracting informative features from the data set that contains high feature dimension and feature correlation, as well as sample scarcity.

According to existing bioinformatics literature, none of the computational classification models are superior to the other. This is due to the implicit research objective and model abilities in extracting informative genes. Although many variants of hybrid selection/classification methods have been proposed, the performance of the models are still heavily reliant on the characteristics of the data sets and the nature of classification methods. In our solution, we analyse differentially expressed microarray genes using genetic algorithms (GAs) and artificial neural networks (ANNs).

The reasons for choosing GA and ANN in this research are that they are the only two algorithms based on the analogy of nature and have received high recognition for the delivery of promising results from various disciplinary areas, such as medical diagnosis (Dybowski et al., 1996; Khan et al., 2001; Djavan et al., 2002; Zhang et al., 2005; Froese et al., 2006; Heckerling et al., 2007), environmental forecasting (Nunnari, 2004; Fatemi, 2006; Nasseri et al., 2008), hardware utilisation prediction (Barletta et al., 2007; Taheri and Mohebbi, 2008), real-time series prediction (Kim and Han, 2000; Sexton and Gupta, 2000; Arifovic and Gencay, 2001), food lifespan forecasting (Gonia et al., 2008), sonar image reading (Montana and Davis, 1989) and computational problem (Sexton and Dorsey, 2000; Kwon and Moon, 2005; Cheng and Ko, 2006; Hu et al., 2007). The ANN is a universal computation algorithm that has the ability to compose complex hypotheses that can explain a high degree of correlation between features without any prior information from the data set (Cartwright, 2008a). Meanwhile, the GA is an effective population-based search algorithm designed for a large, complex and poorly understood data space due to its ability to exploit accumulating information about this unknown data space and to bias subsequent a search into useful subspaces (DeJong, 1988). In addition, GA is robust from trapping into local minima, i.e. the over-fitting problem (Montana and Davis, 1989).

GA/ANN hybrid systems are not new in microarray classification, but, are innovative for gene extraction. Several examples of GA/ANN hybrid systems on classification include breast metastasis recurrence (Bevilacqua et al., 2006a,b), multiclass tumour classification (Cho et al., 2003a; Karzynski et al., 2003; Lin et al., 2006) and DNA sequence motif discovery (Beiko and Charlebois, 2005). In these studies, the data sets were normally normalised and partitioned into several smaller sets to ensure better classification performance of the system. The GA acts as a supporting tool to optimise the classification performance of ANN. This could contribute to the gene variability in the selection results. Rather than emphasize classification performance, our research focuses on the extraction ability of the hybrid GA/ANN. Our approach optimises the connection

weights of ANN and, at the same time, evaluates the fitness function of the GA using 3-layered ANNs. The distinct difference between the existing GA/ANN hybrid systems and our GA/ANN hybrid approach is that rather than using ANN as a classifier to predict cancer classes, the ANN in our approach is act as a fitness score generator to compute the GA fitness function. Figure 1.3 presents the graphical hybridisation of the GA/ANN approach used for classification and our selection approach.



(a) A typical GA/ANN hybrid system for microarray classification.

(b) The proposed GANN hybrid system for microarray gene extraction.

Figure 1.3: A typical GA/ANN hybrid classification model and the proposed GANN feature extraction model. The diagram (a) shows a typical GA/ANN hybrid system used in microarray classification. In this hybrid system, the ANN is used as a classifier to discriminate between cancer classes. The diagram (b) presents the proposed hybrid system focusing on the extraction of informative genes from microarray data. In our system, the ANN is act as a fitness score generator to compute fitness score for GA.

Fitness function is the most crucial aspect in GA as it determines the effectiveness performance of GA. Most research concentrates on optimising other aspects of GA and only a few studies on improving GA fitness function, e.g. the use of a penalty function to identify invalid chromosomes and approximating fitness evaluation within a given amount of computation time (Beasley et al., 1993). However, these approaches require an additional task level in a GA algorithm, for instance, a set of rules for determining the invalidity of chromosomes, i.e. how poor the chromosome is, and a set of mathematical formulas to compute penalty values when GA selecting invalid chromosomes, and consequently, the optimisability performance of GA

relies heavily on how 'good' this additional function is in finding the 'optimal' fitness function. Some studies proposed the use of effective classifiers on fitness computation, for instance, Li et al. (2001b) used the classification result returned by k-nearest neighbour (KNN) as the fitness function of GA on acute leukaemia classification, Cho et al. (2003a) computed fitness function based on neural network prediction results on SRBCTs tumours, Lin et al. (2006) and Bevilacqua et al. (2006b) employed error rate returned by neural network classification as GA fitness function on multiclass microarray data and breast cancer metastasis recurrence, respectively. In our approach, instead of letting the user determine the level of invalid chromosomes, we use simple feedforward ANN to compute fitness values for GA chromosomes. A novel feature of our approach is based on the explicit design of the algorithm which explores the potentialities of the GA and ANN methods of extracting informative features with minimal structural requirements on GAs and ANNs, as followed the Ockham's Razor principle.

Figure 1.3b shows our hybrid approach. To formulate an effective feature extraction method and to circumvent the over-fitting problem, a GA is used to initialise a population of chromosomes in which its fitness value is computed using a 3-layered feedforward ANN with centroid vector principle and Euclidean distance. Once all chromosomes are assigned with fitness values, a set of genetic mechanism is used to assess the fitness of the chromosome and the least fit chromosome is replaced by a new chromosome. Through evolution over many generations, ANN connection weights and GA fitness function are optimised, the least fit chromosomes are gradually replaced by new chromosomes produced in each generation and the optimal set of genes are obtained.

1.4 Research Question and Hypotheses

The research questions are derived from the problems identified in existing literature. Thus, two of our research questions are as follow:

1. Can we use the simplest parameters in both GA and ANN to solve the problems stated in Section 1.2? The simplicity in this context referring to the use of the minimal necessity parameters in both GA and ANN to extract optimal gene subset from the raw (i.e. unprocessed) microarray data.

2. Can we identify informative genes which underly different precision levels in the microarray data? The precision level in this context referring to the minimum fitness accuracy required by the model in selecting informative genes from the raw, unprocessed microarray data.

These questions yield the aim of this research which is to devise a more effective way for extracting informative features using machine learning methods. Thus, the hypotheses focuses on the outcomes of the research and

on the conceptual design of hybridising GA and ANN methods. A feature extraction system has been built based on these hybridised techniques. The hypotheses are as follow:

1. Without the use of acceleration techniques in ANN, the feedforward learning is able to compute the GA fitness function *(Model simplicity, generalisability and normalisation-free)*.
2. The proposed technique is able to detect genes that are differentially expressed in different tumour development stages, i.e. different fitness precision levels *(Biological plausible results)*.

These hypotheses are tested by the design developed in Chapter 3, the prototype and the experimental study in Chapter 4, and the experimental results and discussion in Chapter 5.

1.5 Contributions

The aim of this research is to formulate, from the identified computing-related problems stated in Section 1.2, an innovative feature extraction model using machine learning methods for extracting informative and relevant features using GAs and ANNs. This aim leads to major contributions, which are as follow:

- The realisation of problems pertaining to microarray experiment and the data structure of microarrays. Unlike ordinary real-world data which has small level of interaction between features and high sample size, microarray data contains thousands of genes associated with less than a hundred samples that have been collected from various sources, which could, possibly, yield heterogeneity solutions in gene combinations to the data. Additionally, genes in the microarray data are all correlated to some extent. The aim of microarrays is to provide biological insights into gene interactions for the design of a treatment strategy at the molecular level, instead of finding genes that can perfectly discriminating between cancer classes. This realisation leads to the inducement to design a novel feature extraction approach with as minimal an involvement of statistics as possible.
- A practical approach to identifying informative genes using an innovative hybridising GA and ANN. This solution will assist in answering the problems addressed in Section 1.2.
- A prototype to realise the proposed techniques. This prototype will assist in validating the hypotheses and in providing the fundamental basis for conducting an experimental study.
- The analysis of experimental results to indicate the cognitive performance of the prototype in different precision states and the effect of an innovative hybridisation solution, as well as to demonstrate the significance of the identified genes from a biological perspective.
- The publications of experimental result to various conferences.

In additional to major contributions, the minor contributions stemming from this thesis are as follow:

- The review of existing bioinformatics literature pertaining to cancer classification and gene selection techniques. The strengths and limitations for classification techniques, feature selection and/or reduction techniques and model evaluation approaches were discussed to determine the unresolved problems.
- The identification of informative genes in different stages of tumour development may enhance the insight of the gene-gene interaction in the growth of abnormal cells and may assist practitioners in designing treatment strategies to prevent further progression of a cell abnormality.
- The research in optimising GA fitness function. In our approach, the fundamental ANN paradigm is exploited to increase the capability of the existing fitness optimisation techniques on the aspect of improving the performance of GA.

1.6 Structure of Thesis

This thesis contains six chapters and three appendices. Chapter 2 explores the literature of the human microarray cancer analysis framework. The literature covers the current works in the field and details the intrinsic evaluation between existing works. The current works such as techniques for creating microarrays, selection approaches for the selection of highly expressed genes in the cancer classes, classification methods for discriminating sample patterns and validation mechanisms for validating the performance of classification methods are reviewed.

Chapter 3 describes the planning and design phases that have been carried out to deliver the theme of this research. A conceptual design of our feature extraction prototype namely, Genetic Algorithm-Neural Network (GANN), is designed to study the interaction between informative genes that trigger the proliferation of a specific tumour disease. In this chapter, the experimental data sets, used in supporting our research theme will be discussed. These data sets including two synthetic data sets, two benchmark microarray data sets, i.e. ALL/AML and SRBCTs, and two bioassay data sets, i.e. AID362 and AID688. The accuracy (i.e. correctness) of the selection results will be evaluated with synthetic data sets and the robustness of the our model in dealing with large data sets will be examined with bioassay data sets.

Chapters 4 and 5 describe the prototype and experiments of our approach to identify informative genes that underlies the data. Chapter 4 describes software tools and the experimental study used to support this thesis, including the prototype of our model, and Chapter 5 evaluates the reliability of our approach by comparative studies of our results with four commonly used ANN activation functions, i.e. sigmoid, linear, hyperbolic tangent (tanh) and threshold, as well as the results from the original studies. The insight of the

identified genes will be verified against NCBI genbank via the Entrez Gene search system and the Stanford SOURCE search and retrieval system.

Chapter 6 concludes our work and direction for future work. Our method will be evaluated based on the goal achievement and further improvement in the method will be discussed. Finally, the thesis is concluded.

Chapter 2

Background and Literature Review

Chapter 1 gave the overview of our solution to the problems concerning informative genes detection in Section 1.2. This chapter describes the literature related to the problems and solution presented in this thesis, from both the biological and the computing perspectives.

This thesis describes an intelligent gene extraction method using hybrid GAs and ANNs in microarray studies. Hence the related literature from a biological perspective includes microarray production and its challenges, and the computing perspective includes data preprocessing, classification and prediction modelling, as well as computational challenges.

This chapter contains three sections. Section 2.1 provides the background on microarrays. Aspects concerning the array design, fabrication techniques, fluorescent labelling systems and microarray-related problems will be presented. Section 2.2 reviews computing approaches that have been used in cancer microarray analysis, including data preprocessing approaches, classification and prediction techniques, model validation mechanisms and aspects pertaining to the problems set out in Section 1.2. Section 2.3 provides a summary of the chapter and to what follows next in the thesis.

2.1 A Biological Perspective

The understanding of genetics has advanced remarkably in the last three decades since the first recombinant DNA molecule in the early 1970s. DNA plays a vital role in our daily activity as it makes cells more specialised to perform certain functions, for example pancreatic cells to produce enzymes and insulin for digesting food, or red blood cells to produce haemoglobin to transport oxygen to other cells (oxygenated) and to carry carbon dioxide/monoxide away from cells. To do so, DNA makes *ribonucleic acid (RNA)* by unbinding DNA strands to synthesise message RNA (mRNA). The mRNA is then sythesises with amino

acid units to produce proteins to make cells function. Pragmatic studies have been performed to find a way to measure DNA gene expressions in order to study the cancer biology of some genetic diseases over the last decade and this has led to the vast development of publicly available repositories for microarray experiments and microarray data. Table 2.1 presents examples on well-recognised resources for microarray experiments and repositories. A *microarray* is the DNA chip that is used to store and to analyse the information contained within a genome (the entire DNA sequence of a particular organism) or proteome (the entire complement of proteins expressed by a genome). A microarray contains microscopic spots, i.e. the identical single-stranded *deoxyribonucleic acid (DNA)* that attach to a solid surface, which is then used to detect the presence and abundance of labelled nucleic acids in a biological sample. In the process of making microarrays, *messenger RNA (mRNA)*, *transfer RNA (tRNA)* or *complementary DNA (cDNA)* are extracted from the sample RNA and labelled with a fluorescent dye system, these DNA probes are then hybridised and scanned to produce an image of the surface of the array (Ebert and Golub, 2004). Figure 2.1 on page 19 presents the making of microarrays based on two widely used DNA probes, i.e. cDNA and oligonucleotide arrays.

The microarrays quality and interpretation are influenced by the type of probe used, the way in which the probes are aligned onto a solid support and the technique of target preparation (Ebert and Golub, 2004), thus, great care is taken in conducting microarray experiments. In all cases, the first step is to extract the RNA from the tissue/cell of interest by diluting the biological sample with certain chemical substances and the RNA is then amplified using *polymerase chain reaction (PCR)* assays. The subsequent sections describe the steps in the microarray experiment, including array design, fabrication technologies, labelling systems, hybridisation and image analysis. This section concludes with problems concerning the cDNA arrays.

2.1.1 Array Design

At present, two prevalent approaches for DNA arrays are *cDNA* and *oligonucleotide* arrays, depicted in Figure 2.1, adopt different experimental platforms. The clone-based platform is used to produce cDNA arrays, while the oligonucleotide-based platform is used to create a high density of oligonucleotide arrays. Both arrays exploit hybridisation, however, they differ in terms of probe lengths and its composition, layout of sequences in the array, cross-hybridisation and hybridisation effects from an immobilised substrate (Mah et al., 2004), as well as objectives of the studies. For the studies where the focus is on a specific subject area and the abundance ratio of differentially expressed genes is needed, such as genes relevant to particular metabolic pathways, the low-density of cDNA array is required. Whereas, for the studies where little prior information on relevant genes is available, or where an unbiased overview of global changes in gene expression patterns is required, the high-density of oligonucleotide array is the best option (Tomiuk and Hofmann, 2001). Table 2.2 on page 21 presents a comparison study based on cDNA and oligonucleotide arrays, along with their

Table 2.1: Resources for microarray experiments and microarray repositories.

Website	URL	Resources
National Center for Biotechnology Information (NCBI)	http://www.ncbi.nlm.nih.gov/	GEO database, genbank, analysis software, search browsers
European Bioinformatics Institute (EBI)	http://www.ebi.ac.uk/arrayexpress	genbank, biological ontology, ArrayExpress database
National Human Genome Research Institute (NHGRI)	http://research.nhgri.nih.gov/	cDNA microarray protocols, cDNA microarray repository, analysis software
Broad Institute, cancer genomic group	http://www.broad.mit.edu/	analysis software, microarray repository & associated articles
Standford University, genomic department	http://smd.stanford.edu/	experiment protocols, SOURCE & AmiGO browsers, analysis software, cDNA microarray repository
Microarray Gene Expression Data (MGED) Society	http://www.mged.org/	MIAME standard, gene ontologies, MAGE
GO Consortium	http://www.geneontology.org/	gene ontology and participating laboratories

distinct advantages and disadvantages.

cDNA arrays containing cDNA fragments that are generated by PCR amplification of the cDNA clone, which is the *reverse-transcriptase* of two different biological samples mRNA that are labelled with different dye colours and hybridised to DNA sequences, that are robotically spotted on the surface of the glass slide (Ebert and Golub, 2004). After hybridisation, a special scanner is used to measure the intensity of fluorescence of each differentially expressed gene on a fine grid of pixels and to produce the digital image of hybridised arrays. Normally, higher fluorescence indicates a higher expression value of the gene in the sample. The cDNA array is relatively simple to produce and is inexpensive for laboratories with access to robotic equipment, however, it needs careful attention to the chemistry which adheres the DNA to the glass (Ebert and Golub, 2004). A lack of standard procedure due to manufacturing errors and improvised techniques used in producing high-quality cDNA arrays by individual research laboratories has caused more unnecessary problems than one might expect. For instance, the primary technical difficulty in microarray experiments is the amount of each DNA probe that is robotically spotted on different arrays. To control inconsistency, sample RNA is often hybridised with a defined amount of reference RNA that is labelled with a different fluorescent

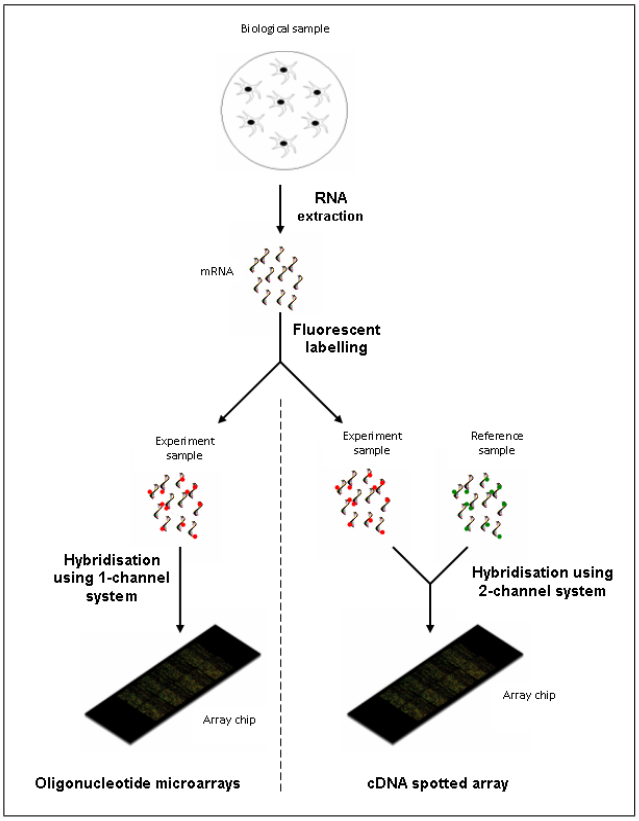


Figure 2.1: The microarray experiments: Oligonucleotide versus cDNA arrays. In all microarray experiments, the RNA is extracted from the biological sample and the RNA is then amplified using PCR assays. For oligonucleotide microarrays, the probes are directly synthesised onto solid surface and the single-dye colour is used to read the gene expression in the sample. For cDNA microarrays, the PCR products from cDNA libraries are deposited onto a solid surface and the two-dye colour is used to read the gene expression in the samples.

dye (Ebert and Golub, 2004). However, this yielded another technical concern, that is, the amount of reference RNA in the hybridisation process is dependent on the amount of probes that are robotically spotted and also the manufacturer's guideline. Additionally, cDNA probes often contain repetitive sequences, as a result, the process becomes intensive, especially when the experiment is conducted on a genome-wide scale, consequently, the cross-hybridisation has become more problematic (Ebert and Golub, 2004). A significant advantage of cDNA array is that it does not require prior sequence information due to it being initially designed for sequence modelling. Thus, it is an attractive alternative for model organisms whose genomes are not yet sequenced (Ebert and Golub, 2004). The disadvantages of cDNA array are, inconsistency in the procedure adopted by the individual research laboratories and manufacturers resulting in a variation in gene measurements, intensive computational cost on a genome-wide scale experiment and is problematic on cross-hybridisation due to repetitive sequences in cDNA microarrays.

Comparing to cDNA arrays, high-density *oligonucleotide array* is an active area of technological development (Parmigiani et al., 2003) and the most widely used oligonucleotide arrays are manufactured by Affymetrix (Santa Clara, CA) which uses the in-situ photolithographic synthesis technique to produce oligonucleotides

onto the array chips. Therefore, oligonucleotide arrays, also referred to as *Affymetrix arrays*, are less problematic than cDNA arrays. The oligonucleotide arrays contain short oligonucleotide probes with a length between 25 and 60 mers (base-pairs) that are either synthesised in-situ or robotically spotted on the surface of the glass side. In oligonucleotide array, sample RNA is prepared, labelled with dye colour and hybridised to an array which is then scanned into digital image to obtain a fluorescence intensity value for each probe. Unlike cDNA arrays, oligonucleotide arrays use single-channel labelling system, i.e. single dye colour, in hybridisation and each oligo probe contains a unique oligonucleotide sequence (Ebert and Golub, 2004) that ease the hybridisation process. Due to the short probe length in oligonucleotide arrays, the hybridisation specificity is more easily controlled than cDNA arrays.

Literature (Mah et al., 2004; Asyali et al., 2006) shows that although cDNA and oligonucleotide arrays have a poor correlation on the DNA probes, both arrays are able to display similar characteristics in the data, even though, the combined results of both arrays have been not possible. Hence, studies on the combination of results on multiple similar arrays have gained close attention from the bioinformatics field. With the robust and reproducible gene expression data that can be generated on multiple similar arrays, the technicality aspects of array design have become less critical (Ebert and Golub, 2004). However, the degree of similarity of the DNA probe sets and the expression level of the corresponding transcript in the experiments still play important roles in the reproducibility aspect and yet, this issue remains a topic of intensive research (Asyali et al., 2006).

2.1.2 Fabrication Technology

Two prevalent microarray fabrication technologies in microarray experiments are robotic spotting and in-situ synthesis. The *robotic spotting* synthesises DNA probes prior to the array deposition which is then detected onto glass, while the *in-situ synthesis* synthesises DNA probes directly onto the array surface without depositing intact DNA probes.

For *robotic spotting*, robotic-controlled pins are dipped into wells that contain DNA probes and then deposited each probe at the designated locations on the array surface. The amount of probes collected depend on the number of arrays being made and the amount of liquid the pins can hold. The pins are then washed to remove any residual solution before the next sample is collected to prevent contamination on the subsequent sample. Once all locations on the array are occupied with probes, the known reference template, i.e. complementary cDNA or cRNA "targets" derived from experimental samples which represent the nucleic acid profiles is prepared to hybridise with cDNA probes. The probes spotted by the spotting technique can be cDNA, oligonucleotides or even small fragments of PCR products that correspond to mRNA.

For *in-situ synthesis*, the probes are short oligonucleotide sequences, in the range of 25 to 60 mer probes, that

Table 2.2: A comparison between cDNA and oligonucleotide arrays.

Description	cDNA arrays	Oligonucleotide arrays
Platform design	clone-based	Oligonucleotide-based / Affymetrix-based
Probe length	Long sequence	Short sequence, approx. 25 - 60 mer (base pair)
Fabrication technique	Robotic spotting	Synthesising in-situ
Labelling system	Two-dye fluorescent colours	Single-dye fluorescent colour
PCR correlation	Low	High
Production cost	Low	High
Normalisation	Yes (pre- & post-normalisation)	Yes (pre-normalisation)
Advantages	Emphasize on genes related to a specific subject areas Easily customise No prior information on cDNA sequences required	Unbiased overview on the global fluctuations in gene expression patterns Less sensitive to cross-hybrdisation Uniformity of probe length and the ability to discern splice variants Able to recover samples after hybridisation to a chip
Disadvantages	Difficulties on the production of high-quality cDNA arrays Variability in gene expression measurement Vulnerable to cross-hybridisation Problems on reproducibility of genes May contain latent non-specific DNA sequences	Sensitive to base pair changes due to short length of DNA probes involved Requires prior information on relevant genes

are built up base-by-base on the array surface that are designed to match parts of the known sequence (Stekel, 2003a). With each added nucleotide to the array, there is a protective group to prevent more than one base being added during each round of synthesis. This protective group is then converted to a *hydroxyl group*, either with photolithographic synthesis or chemical reagents via inkjet technology. The *photolithographic synthesis* approach uses light to convert the protection group, while the *reagent approach* employs a similar chemistry substance as a standard DNA synthesiser and droplets of the appropriate base are fired at each step of synthesis onto the desired spot via inkjet printers which fires adenine (A), cytosine (C), guanine (G) and thymine (T) nucleotides. The main advantage of the inkjet approach over photolithographic synthesis arrays and spotted arrays is that the gene synthesis process is highly flexible as it is fully controlled by the computer, based on the user requirement. However, it is less efficient for making a large quantity of identical arrays (Stekel, 2003a). Chemical reagents via inkjet technology have been used by some reputable laboratories such as Rosetta, Agilent and Oxford Gene Technology (Tomiuk and Hofmann, 2001; Stekel,

2003a).

At present, there are two types of photolithographic synthesis: masks and maskless. The *masks photolithographic synthesis* is the basis of the Affymetrix technology that uses masks to allow light to pass to a certain area of the array and at each step of synthesis, different types of masks are required (Stekel, 2003a). The *maskless photolithographic synthesis* used by Nimblegen and Febit, employs a computer-controlled mirror to direct light to the desired parts of the glass slide at each step of synthesis (Stekel, 2003a).

2.1.3 Labelling Systems

After RNA is extracted and fabricated, the sample can then be labelled. The labelling process can be either fluorescence or radioactive, depending on the fabrication technique used in the microarray experiment. This thesis focuses on the fluorescence labelling system as it is the most commonly used system rather than the radioactive system. Two types of fluorescence labelling systems for oligonucleotide and cDNA arrays are single-channel (i.e. single-dye) and two-channel (i.e. two-dye) systems, respectively.

On Affymetrix arrays, i.e. oligonucleotide arrays, a biotin-labelled cRNA is constructed for hybridising to the Affymetrix GeneChip, thus, a *single-channel* is commonly used to show the expression values of given genes. A significant advantage of single-channel labelling is that the collected data can represent absolute expression values of genes, as a result, an aberrant sample cannot affect the data collected from other samples as each array chip can only hold one sample as opposed to a two-channel labelling system where an array chip is exposed to many samples, i.e. cross-hybridisation. Consequently, the overall data precision on 2-channel labelling system could be affected if there is an outlier between samples in the chip. Additional strengths of the single-channel system are the comparison studies between arrays which can be conducted in noncontinuous time frames, such as months or years and the combination results of multiple arrays on different array chips are possible as a similar dye colour system is applied. A significant drawback of this labelling system is that it requires more array chips to store the data.

Meanwhile, on cDNA arrays, cDNA probes and reference template are differentially labelled with *two fluorophores* to allow for the quantification of differential gene expression. Expression values are reported as ratios between two fluorescent values (Ramaswamy and Golub, 2002), as shown in Figure 2.2. Two samples, i.e. disease cell versus normal cell, are compared and labelled using two different fluorescent dyes, normally with Cy3 corresponding to the green fluorescent probes and Cy5 corresponding to the red fluorescent probes. These two Cy-labelled cDNA samples are then hybridised to a single microarray using control probes provided by oligonucleotide microarrays which is then scanned to visualise the fluorescence of the probes after they have been excited with a laser beam. The fluorescence intensities of each probe are then normalised, based on the preselected control probes and analysed using a specially designed computer algorithm to detect

the regulation of genes. A two-channel system is prone to dye biases due to different dye properties and dye ability which result in a variation in gene expression measurements using the same DNA sample (Parmigiani et al., 2003). To remove dye biases, additional normalisation is performed after digitising hybridised arrays, which is elaborated in Section 2.1.5.

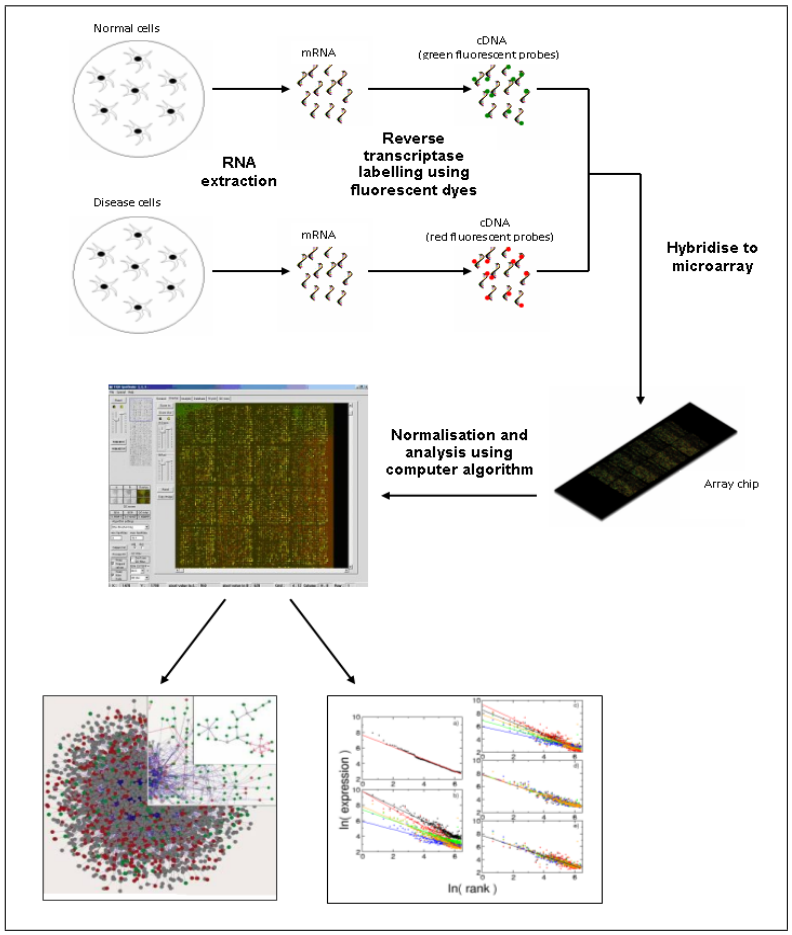


Figure 2.2: A typical 2-channel microarrays. The RNA are extracted from the normal and disease cells and amplified using the PCR assays. The RNA are then reverse-transcriptase into the cDNA, which is then labelled with two different fluorophores using two-channel labelling system. Two fluorescent dyes for cDNA labelling are Cy3 (green) and Cy5 (red). The two Cy-labelled cDNA samples are mixed and hybridised to a single microarray that is subsequently scanned in a microarray scanner to visualise the fluorescence of the fluorophores after the excitation with a laser beam of a defined wavelength. The intensities of each flurophore are then be used in the ratio-based analysis to identify the regulation of genes.

2.1.4 Hybridisation

After DNA probes have been labelled with a proper dye system, hybridisation is performed by complementarily combining DNA probes and the labelled RNA reference template. Hybridisation can be performed either manually or robotically. In the past, the sample has been manually hybridised and, to date, due to the advance of biology technology, most research laboratories use robotic hybridisation which can provide a much better control of the temperature of the target and slide (Stekel, 2003a), thus reducing the variability error

in gene expression measurement. In addition to the ambient condition factor, variability errors can also arise from other factors, such as salt concentrations, formamide concentration, target solution volume and operator, edge effects (effects seen only at the gene spotted near the edges of the array) and cross-hybridisation (Stekel, 2003a; Parmigiani et al., 2003).

2.1.5 Image Analysis

The final step of microarray experiment is to produce an image of the hybridised array. A special scanner is used to read the fluorescent dyes on the surface of the glass slide that are excited by the scanner laser beam. In order to make a digital image of the array, the laser must focus at the desired point on the array so that the dye at that point is excited by the laser and then detected by a photo-multiplier tube (PMT) in the scanner (Stekel, 2003a). The quality of the digital image is generally associated with fabrication techniques, hybridisation and dye labelling. For instance, Affymetrix arrays have light refraction problem on the masks photolithographic synthesis, thus, it is compensated by a high quality of image processing software (Stekel, 2003a), meanwhile spotted array images can be of variable quality, depending on the variation in hybridisation and differential dye labelling (Auburn et al., 2005).

After producing a digital image of the array, some form of normalisation is generally applied to microarrays to remove any form of biases yielded by fluorescent dyes used to label cDNA-based samples (Leung and Cavalieri, 2003; Wilson et al., 2003). There are two major groups of normalisation, i.e. within-array and between-array normalisation. The within-array normalisation compares samples of a single array chip, meanwhile the between-array normalisation makes comparisons between samples hybridised to multiple array chips.

The *within-array normalisation* generally involves either fitting a regression line to the log intensities of dyes versus average intensity of each probe (Stekel, 2003b) or to form a smooth curve (lowess curve) based on the joining of a large number of local regressions in every subset of data using log intensities of dyes (Leung and Cavalieri, 2003; Stekel, 2003b; Wilson et al., 2003). However, a significant drawback of within-array normalisation is that it is based on the assumption that a majority of the genes are not differentially expressed (Stekel, 2003b).

The *between-array normalisation*, on the other hand, involves the global normalising gene expression based on the equalisation of expression using the mean of all genes within an array across the mean of all expressed genes from different arrays (Asyali et al., 2006; Leung and Cavalieri, 2003), with a core assumption that genes can be differentially expressed as a result of the experimental condition and do not represent biological variability (Stekel, 2003b; Asyali et al., 2006). However, depending on the biological conditions of different arrays, this assumption could be violated. A solution to this violation is to use a set of common *housekeeping genes* on the arrays to replace the mean expression normalisation of all the genes in the array or the known

amount of *exogenous control genes* to be added to the finished microarrays. Housekeeping genes are genes that have stable differential expression across different biological conditions (Asyali et al., 2006) with a low differential expression and a small variability after normalisation (Wilson et al., 2003). Recent reviews show that housekeeping genes are not as constantly expressed as was previously assumed (Leung and Cavalieri, 2003) and improper use of housekeeping genes may yield another potential source of error. Hence, the use of the *dye-swapping experiment* is seen as a plausible solution. However, this experiment is impractical in use, due to the limited supply of samples that fulfil certain criteria (Leung and Cavalieri, 2003).

To conclude, fluorescence biases are primarily dependent on the average expression level and spatial position of the probe in the cDNA microarrays in which such conditions are determined by several factors, including the experimental protocols, the lengths of probes, the number of tissues to be measured per array and the array size (see Section 2.1.1). Therefore, the post-normalisation in the image analysis phase (see Section 2.1.5) is used to remove biases yielded from such conditions, but not to remove biases caused by other factors such as printing resolution, scanning efficiency, labelling efficiency and PCR assays.

2.1.6 Microarray Challenge

cDNA microarray has been widely studied over the last decade, however, it is still imperfect in some ways due to some technical problems incurred during the production process. The key problem is the variability in gene expression measurements using the same DNA sample. This variability arises in almost every phase of the cDNA microarray experiment, such as sample preparation (amplification, purification, concentration, spotting volume), sample labelling (dye properties), hybridisation (ambient conditions, spotting effects, cross-hybridisation) and image processing (scanner property, image algorithm and settings) (Parmigiani et al., 2003). Although these errors are relatively small, however, the compounding of their effects can be severe, particularly, in verifying the underlying functionality of correlated genes to specific subject studies. A solution for gene validation is to apply alternative biological techniques, such as reverse transcriptase-PCR (RT-PCR), or fluorescent in-situ hybridisation (FISH) , to validate the gene functions (Ebert and Golub, 2004). RT-PCR, however, it is only practical in a smaller number of genes and it is not to be used in detecting variability errors. Thus, variability errors are usually expected in finished microarrays (Parmigiani et al., 2003).

The low production cost of cDNA microarrays has motivated individual research laboratories to design 'indoor' microarrays with no proper guideline in experimental design, and so far, these arrays have only been used internally. However, with the advances of microarray technologies, these 'indoor' microarrays have caused additional problems in the "standard" microarrays. These problems are:

- The lack of a systematic procedure to design microarray due to individual research laboratories customising the experimental process to suit their requirements. For instance, all DNA probes are usually replicated at least twice in every stage of the experiment to obtain a solid conclusion on both the statistical and biological significance. This replication process may be conducted on two independent RNA extractions or on two aliquots (i.e. a portion of a DNA sequence) of the same extraction. Each spot on the array may also be replicated to provide a measure for each hybridisation process. Thus, an inconsistency of experimental design occurs. To overcome this inconsistency, several institutions have participated in the Laboratory Information Management System (LIMS) scheme to understand the methodology adopted by different laboratories in designing microarray experiments (Stekel, 2003c). A LIMS repository records all information regarding laboratory experiments, including procedures, protocols and methods in microarray manufacture, sample preparation, labelling and hybridisation (Stekel, 2003c). However, LIMS receives little attention from research laboratories.

- The lack of annotation control due to there being no standard practice applied in the experimental stages, such as fabrication techniques, assay protocols, statistical analysis methods and annotation protocols adopted by individual research laboratories (Leung and Cavalieri, 2003). For example, Bruchova et al. (2004) quantitated the cDNA microarrays based on the manufacturer protocol of Atlas Img 2.01 software in which the global sum normalisation method has been applied the gene expression values and they used the ratios of signal intensities ≥ 2 and ≤ 5 to determine the significance of the genes. Rather than using the global normalisation method, Sakhinia et al. (2006) quantitated the cDNA gene expression values based on the defined housekeeping genes and the significance of the genes are determined using the Mann-Whitney test with $p \leq 0.05$. Mah et al. (2004), however, performed 2-steps normalisation in the cDNA gene expression quantitation to remove fluorescent bias that was introduced in the labelling phase. They first used the log-transformed global mean method on the data values and the scaling factor is applied to prevent negative log values to remove fluorescent bias. They then normalised the expression values according to the median expression intensity over all arrays (i.e. the Zipf's law), followed by a scaling factor to return the expression values to their original magnitude. As a result, it is almost impossible to either judge the validity of the cDNA results or to compare the cDNA results from different laboratories. Furthermore, the cDNA gene expressions are annotated in the numerical format. Without a standard, similar numeric values to annotate different cDNA probes become possible. For example, cDNA probe with the Image Id 782427 in the SRBCTs cDNA microarray data (Khan et al., 2001) is annotated as EST (expressed sequence tag, which referring as the unclassified gene or gene in which it function is unknown) can referring to two distinct genes in the NCBI genbank, i.e. GRN gene in chromosome 17 and INHBB gene in chromosome 2. To circumvent the problem, the Microarray Gene Expression Data (MGED) society enforces the Minimum Information

About a Microarray Experiment (MIAME) protocol, that outlines the minimum information required for creating the microarray data (Leung and Cavalieri, 2003). This is to ensure that the data can be easily interpreted and the results can be verified independently by different laboratories. However, there will always be some laboratories which have different experimental procedures by chance.

2.2 A Computing Perspective

The advancement of computing technology in the early 1960s has laid the fundamental methodology for genetic analysis in the bioinformatics field, with an emphasis on cancer classification. In the past, cancer classification has always been morphological and clinical-based (Lu and Han, 2003), as a result, the medical conclusion has always been statistical-based. With the bloom of the Internet in 1990s, a fully developed bioinformatics field was born and extensive research in computing methods for microarray classification has been studied. Unlike clinical-based classification, microarray-based classification focuses on the search for feature genes that are differentially expressed between cancer classes. Figure 2.3 presents a cancer classification process based on supervised classification methods. The data preprocessing step is generally expected in microarray data to remove any sort of data incompatibility in supporting classification results. Depending on the circumstances, sometimes more than one preprocessing techniques is needed, such as imputing missing data, normalising data, removing redundant information or preselecting data which meets certain criteria. After the data has been preprocessed, a validation mechanism is normally chosen to evaluate the performance of the classifier and the entropy rate of the trained classifier is computed. Depending on the entropy results returned by the classifier, appropriate adjustments will be made on the classifier, the validation method and the preprocessing approach. Normally, a separate set of test samples will be used to evaluate the generalisability of the classifier in discriminating these unknown samples.

The subsequent sections describe the computing aspects in classifying microarray data, which include data preprocessing, classification methods and validation mechanisms. A review of the existing selection methods and problems pertaining to the computing aspects addressed in Section 1.2 will be discussed in this section.

2.2.1 Data Preprocessing

There are many factors that affect the success of classification methods. The first and foremost aspect is the representation and quality of the data (Asyali et al., 2006; Kotsiantis et al., 2006). This is important for classifiers, such as ANNs and KNN estimation, as these techniques are sensitive to the value magnitude, and consequently, the predictive results will deteriorate when the magnitude is high. Thus, the removal of undesirable characteristics, such as outliers, missing data values, excessive data values and data redundancy,

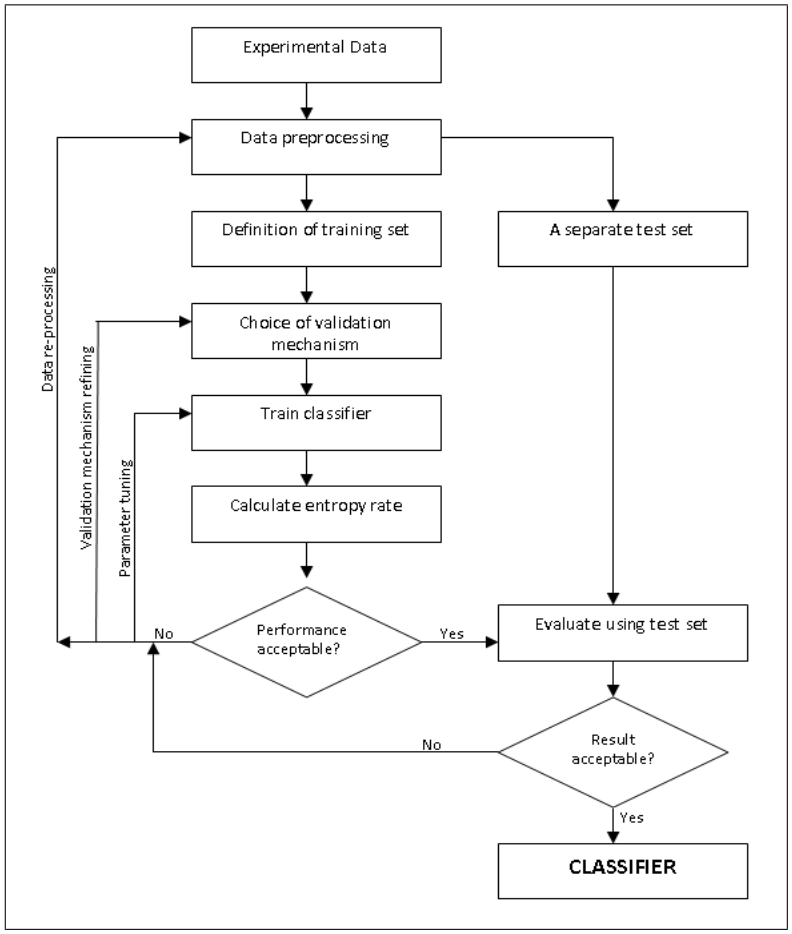


Figure 2.3: The process of supervised classification methods.

can ensure data integrity and better predictive accuracy of the classification results. In general, there are three principle aspects for preprocessing data which are missing value estimation, data normalisation and feature selection, depending on the nature of the data, the appropriate action is performed. For instance, an estimated value, based on standard statistics, is imputed for the missing entity in the data, normalisation is performed to remove outliers and excessive data values and feature selection is used to identify significant features and to remove irrelevant and redundant information from the data.

2.2.1.1 Missing value estimation

The imputation of estimated values for missing entities is not an uncommon issue in data preprocessing as most real-world data contains incomplete information (Kotsiantis et al., 2006) and sometimes, due to technical problems in microarray experiments, expression levels for some genes cannot be accurately measured, resulting missing data (Kim et al., 2005; Asyali et al., 2006). Classifiers such as nearest-neighbour (KNN) algorithm, support vector machines (SVMs), principal component analysis (PCA), singular value decomposition (SVD), hierarchical clustering (HC) and K-means clustering, cannot be directly applied to

data with missing values, therefore, these missing values need to be imputed with some reasonable estimates using imputation techniques. A commonly used imputation approach is to substitute a missing entity with the global mean value of a feature computed from the remaining available samples. A variant of using a general feature mean is to use the mean value of a feature computed from all samples belonging to the same class for a missing value (Kotsiantis et al., 2006). These are straightforward approaches for imputing missing values in the data with a minimal effort in statistics. The other imputation approach includes the use of KNN-based imputation (KNNimpute) (Troyanskaya et al., 2001; Dudoit et al., 2002) based on the average weight value of the selected genes with expression profiles similar to the missing gene, the employment of SVD-based imputation (SVDimpute) (Troyanskaya et al., 2001) to identify a set of mutually orthogonal expression patterns that can be linearly combined to estimate the expression levels of all genes in the data set, the least squares-based imputation (LSimpute) (Kim et al., 2005) which utilises Pearson correlation in selecting genes and arrays, the Bayesian PCA (BPCA) (Oba et al., 2003) that estimates a probabilistic model and latent variables within the framework of Bayes inference using principal component analysis, and the local least squares imputation (LLSimpute) (Kim et al., 2005) which only uses similar genes based on a similarity measure in the imputation. As a result, the variability of imputation measurements on the similar data set is inevitable.

2.2.1.2 Data normalisation

In addition to incomplete information in microarray data, there is often a large difference between the maximum and the minimum values within a feature, some may be due to outliers or missing values in the data. It is important to note that the normalisation technique in this thesis refers to the process of scaling down the magnitude of a feature to a certain level of range prior to computational analysis, rather than scaling up the magnitude of a feature. Ideally, normalisation enhances the predictive performance of classifiers as it preserves the relationship between features in the samples and simplifies computation process. The straightforward normalisation approach is the scale normalisation (Golub et al., 1999; Dudoit et al., 2002; Cheng and Li, 2008) where samples in the data set are standardised with zero mean and unit variance across features, to prevent dominance of feature values in a sample over the mean values of the features in all samples. The other commonly used approach includes the max-min normalisation (Cho and Won, 2007; Gonia et al., 2008) which divides the feature value with the substraction of the maximum and the minimum values within a feature and logarithmic transformation (Golub et al., 1999; Dudoit et al., 2002).

However, due to the complex biological interaction between genes in microarray data, normalisation is not always versatile. Conversely, normalisation deteriorates the finding of correlated genes in response to labelled classes. Normally, marker genes are triggered by its correlated genes which are either over-expressed or

suppressed in a tumour cell. By scaling down the gene magnitudes, the expression values of the genes in the microarray data may end up equalised. In other words, the intensity of expressed genes are suppressed into a specific range to form a clear discrimination between cancer classes, and consequently, what was originally a primary gene may become of equal statistically significance as secondary and less significant genes. The true marker genes can be 'buried' by higher expression genes and may not be selected in the classification process as they do not provide 'beneficial' information in class discrimination.

At present, most classifiers require data to be normalised to ensure better classification accuracy and model efficiency. No attention has been paid to the implication of normalisation in diminishing genes dependencies and little attention has been made to the influence of normalisation technique in classification performance (Futschik et al., 2003). This is mainly because data normalisation is a standard practice applied to all discipline areas and has been proven effective for computing methods to achieve better classification results. Most real-world data that has been dealt with does not exhibit such complex behaviour or underlying meaning features as compared to microarray data. However, all microarray data has been normalised to some extent during the microarray production (see Section 2.1.5 on page 24 and Table 2.2 on page 21) and therefore, further normalisation could deteriorate the quality of microarrays.

To conclude, normalising data enables a better classification accuracy and model efficiency for classifiers rather than improving the ability of the classifiers in discovering genes interactions. In addition, there are many variants of normalisation which can be applied depending on the subject of studies. In the case of improving classification performance, normalising data will significantly improve processing time and reduce computational effort in classifiers. However, in the case of explaining the context of features dependencies, the normalisation process jeopardises the finding of informative features. To validate our argument, a comparison study based on the original data set and the normalised data set is discussed in Section 5.6 on page 169.

2.2.1.3 Feature selection/reduction

An important aspect for designing a cancer classifier is to reduce feature dimension or to select a set of informative genes to enable the reliable prediction of the model with a limited number of available samples. Thus, the algorithms for removing redundant features and identifying informative data have been developed. Feature selection/reduction is crucial in microarray studies because microarray data contains a greater number of noisy genes rather than informative genes, and these informative genes are less likely to be detected among the large numbers of irrelevant genes due to their biological behaviour. There are three main categories of feature selection/reduction methods, namely filter, wrapper and embedded methods. The filter method, obtaining genes based on its *individuality* correlation to the known cancer classes, the wrapper method, identifies a group of genes that is *correlated* in response to the classification task and the embedded method,

similar to the wrapper method with the only difference that the embedded method is a *built-in* component of the classifier. Table 2.3 on page 32 provides a common taxonomy of feature selection/reduction methods with their prominent advantages and disadvantages.

Filter selection is the earliest selection approach used in cancer classification. It measures the marginal relevance of an individual gene to the known class with standard statistics criteria such as t-statistic methods, and ranked genes according to its entropy rate. A primary advantage of filter selection is that it is classifier-independent, thus, it can be easily applied to any classifiers. Furthermore, it is easily scaled to a very high dimension as the genes are individually evaluated and it is computationally simple and fast. However, a significant drawback of the filter method is that it does not guarantee the delivery of relevant genes due to most filter methods being univariate-based, thus, gene dependencies are not considered. Examples of filter selection are t-statistic methods including ANOVA, t-test, signal-to-noise (S2N) ratio (Golub et al., 1999; Inza et al., 2004; Osareh and Shadgar, 2008; Zhang et al., 2008), principal component analysis (PCA) (Khan et al., 2001; Wei et al., 2004; Wang et al., 2006), between-group/within-group (BSS/BWS) ratio (Dudoit et al., 2000), information gain (IG) (Osareh and Shadgar, 2008; Zhang et al., 2008) and Wilcoxon ranksum (Jeffery et al., 2006; Zhang et al., 2008). With the advance of microarray technology, it is evident that genes combination, rather than genes in isolation, contributes to tumour development. Thus, multivariate-based selection was introduced into cancer classification.

Unlike the filter approach, *wrapper selection* measures the marginal relevance of gene subsets based on the estimation of entropy in trained classifiers and ranks them based on the classification accuracy obtained with a separate set of test data. In wrapper selection, the classifier is 'wrapped' by a selection method. A hypothesis space containing an optimal gene subset extracted from the entire gene space is first constructed using an exhaustive search procedure and these genes are then used to train the classifier. The correlation of the genes are measured based on the estimation of the training accuracy percentage (i.e. entropy rate) of the classifier. Finally, the optimality of genes is validated using a separate set of data. A primary advantage of wrapper selection is that it examines genes dependencies, thus, the genes with low expression levels but strong interactions can easily be detected. The common disadvantages of wrapper selection are that it is computationally intensive and has a higher risk of an over-fitting problem than the filter approach, depending on the choice of classifiers and the parameters of the selection method. Although the wrapper approach has been extensively studied by the machine learning community and widely applied in pattern recognition and classification problems, it is not commonly used in microarray studies (Inza et al., 2004; Asyali et al., 2006), as compared to filter selection and embedded approach, probably due to the intensive computational efforts involved. A commonly used wrapper method is evolutionary algorithm (EA) techniques including genetic algorithm (GA), simulated annealing (SA) and genetic programming (GP) (Li et al., 2001a,b; Ooi and Tan,

2003; Peng et al., 2003; Yu et al., 2007).

Embedded approach is a variant of wrapper selection in which the search of optimal genes is built into the classifiers, thus, the gene space and the hypothesis space can be viewed as one (Saeys et al., 2007). The claimed advantages of the embedded approach are that it explores the interaction between a search algorithm and classifiers and it is less computationally intensive than the wrapper method. However, the embedded approach does not guarantee the genes interaction and it is classifier-dependent. The common examples of the embedded approach include regression search in Fisher discriminant analysis (FDA) (Dudoit et al., 2000), recursive feature elimination (RFE) using weight vector of support vector machines (SVMs) (Guyon et al., 2002), signal-to-noise (S2N) utilised the mean and deviation of genes in weighted voting (WV) algorithm (Golub et al., 1999), random forest (Díaz-Uriarte and de Andrés, 2006) and nearest shrunken centroid (NSC) that normalises the gene deviations within a class in a supervised classifier (Tibshirani et al., 2002). There is an interchangeable term used in the embedded search approach and some supervised classification methods. For instance, much of the literature has generally identified FDA as a supervised classifier rather than a variant of the feature selection approach and NSC is normally used for classification problems, although it is initially used to support a supervised classifier. To avoid any form of confusion, we categorise the embedded approach under the umbrella of the classification method in this thesis because the "classification" term has been used since the establishment of bioinformatics and the "embedded" term has just recently been introduced to bioinformatics field.

Table 2.3: A common taxonomy of feature selection/reduction methods.

Model search	Advantages	Disadvantages	Examples
Filter	Scaleable Fast Simple Classifier-independent	Ignore gene interactions Statistical significant Ignore interaction with the classifier	t-statistics; ANOVA; PCA; BWS; IG; Wilcoxon ranksum
Wrapper	Model biological gene interactions Interacts with classifier Less prone to local optima	Over-fitting risk Computationally intensive Classifier-dependent	GAs; SAs; GPs
Embedded	Interacts with classifier Less computationally intensive than wrapper methods	Classifier-dependent May ignore gene interactions	RFE-SVM; S2N in WV; regression in FDA; NSC; Random forest; CART; Decision Tree

The goal of this research is to devise a more effective way for extracting informative features using machine learning techniques. Therefore, a review of the existing feature selection methods is presented in Section 2.2.4 on page 45.

2.2.2 Validation Mechanism

Assessment of the statistical significance and validation of findings is a critical step in cancer classification (Ebert and Golub, 2004). The principle of microarray-based classification is to use it as an insight for the existing data for prediction of future data. As Simon (2003) said: *"We want to be able to predict class membership for future samples whose class membership we do not know"*. However, the major problem of the existing classification methods is the over- or under-fitting problem, which is due to the nature of microarray data, such as high gene dimension, sample scarcity and complex interaction between genes within the data; the structure of the classification framework, such as data preprocessing prior to classification that may homogenised the primary features and the secondary or the less important features; complex hybrid classification model, such as DT/SVM (Statnikov et al., 2004); and high network size in the ANN model (Chen et al., 2007). Thus, methods for obtaining an unbiased assessment on the classifier's error rate are required. These methods, ideally, assess one or more of the following factors (Simon et al., 2003; Somorjai et al., 2003; Dabney, 2005):

- The unbiased estimation on the prediction accuracy in which high classification corresponds to low misclassification errors *(Classification Accuracy)*.
- The practicality of the prediction model dealing with the data characterised by high feature dimension and sample sparsity *(Scaleability and Generalisability)*.
- The interpretability of the prediction models based on the complexity of the hybridised approach to reduce computational cost and to minimise the risk of over-fitting *(Simplicity)*.
- The interpretability of the results at a biological level rather than at a statistical significant *(Biological versus Statistical Significance)*.

The validation techniques in this section, however, evaluate the accuracy and the generalisability of the classification models in providing statistical-based results rather than biological-based. These techniques include splitting sample patterns and cross-validation (CV) procedures. Table 2.4 on page 34 shows the taxonomy of validation mechanism for evaluating classification methods.

The straightforward approach for properly evaluating a classifier is to base the evaluation on a separate set of test sample data. In a *sample-splitting* procedure, the data set is randomly split into two independent subsets, namely *training set* and *test set*. The splitting ratio is normally $0.7 : 0.3$ in which 70% of the data is used as a training set and the remaining 30% as a test set. The training set is used by a classifier to learn discriminant patterns and the test set is used for evaluating generalisability of the trained classifier. The fundamental principle of this procedure is that the samples in the test set must not have been used in

training the classifier (Ambroise and McLachlan, 2002; Dupuy and Simon, 2007). In fact, some studies use a third set, known as *validation set*, to act as a secondary test set, on the quality of the classifier. This method will prevent the over-fitting problem, providing the data set is large and there are sufficient samples in all three sets (Cartwright, 2008a).

Another commonly used validation approach is *cross-validation (CV)*. Unlike the sample-splitting procedure, CV is an iterative process. In each iteration, part of the data set is used to develop a classifier and another part of the data set is left apart as a validation set to the classifier. Similar to the sample-splitting procedure, the samples in the validation set must not have been used in training the classifier (Dupuy and Simon, 2007).

The widely used CV approach in bioinformatics literature is *leave-one-out cross-validation(LOOCV)* due to the small number of samples available in microarray data (Cartwright, 2008a). For LOOCV, each time the classifier is trained on the entire samples, with the exception of a single sample that acts as a test for the trained classifier. The performance of the classifier is then accessed by this exception sample. This is carried out for all samples in the data set. The main advantage of LOOCV is that it ensures the unbias estimation on the classification result, however, the downside of LOOCV is that it is computationally intensive. Thus, it is not recommended if the data sets are large, but, the *n-fold cross-validation* (CV) procedure is suggested. In n-fold CV, the data set is divided into n subset of data with equal sized of sample patterns in each fold. Each time, the classifier is trained using $n-1$ subsets of data and tested with the omitting subset of data. The process repeats for n iteration and each time, a different subset of data is used to evaluate the classifier. Comparing to LOOCV procedure, n-fold CV is efficient when the sample size is large.

Table 2.4: A common taxonomy of validation mechanism on classification model.

Validation mechanism	Description	Advantages	Disadvantages
Sample-splitting	Randomly divides data with the ratio of 0.7 : 0.3 in training and test sets, respectively. The training set contains adequate information to develop the classifier and the test set is used to evaluate generalisability of the trained classifier	Simple and fast Less prone to over-fitting risk	Biased classification estimation when the sample size is small
Cross-validation	Iterative process where the data is randomly divided into n folds with equal seized of sample and classifier is developed on the $n-1$ folds with tested with the omitting fold	Unbiased classification estimation Less prone to over-fitting risk	Computationally intensive when the sample size is large

Most studies in the bioinformatics literature favoured to apply more than one variant of validation mechanism in their studies. The commonly used combined mechanisms are the sample-splitting and LOOCV procedures,

where the prediction performance of the trained classifier is evaluated using the n-fold CV procedure in the training set (see Appendix C).

Simon (2003) and Ambroise and McLachlan (2002) discussed several important issues on model evaluation that have been overlooked by some studies and optimistically bias prediction performances were reported.

2.2.3 Classification Design

At present, the two dominant designs in cancer classification are class prediction and class discovery, each is used as a different study's objective. When the study emphasises either the insight exploration of the known genes or the risk prediction of metastasis activity (survival rate of cancer patients), based on the known cancer classes, *class prediction* design is the only option (Khan et al., 2001; Dudoit et al., 2002; Cho et al., 2003b; Lee and Lee, 2003; Bloom et al., 2004; Liu et al., 2004a,c; Lee et al., 2005). While the objective of study is to discover the unknown cancer class from the existing cancer classes, the model construction is based on *class discovery* paradigm (Golub et al., 1999; Ross et al., 2000). Both approaches are concentrating on the improvement of classification accuracy rather than other issues. The class prediction approach develops the predictive strength of the models based on the classification training accomplished with a sequence of training samples associated with the target output for the training samples. Hence, it is also known as *supervised learning*. While the class discovery approach develops the discovery ability of the models based on the observation of the correlation between samples, that is, genes that are expressed in a similar manner are grouped in similar clusters, while samples with a low similarity of genes are a distance away. Therefore, it is known as *unsupervised learning* in classification literature. Table 2.5 presents a common taxonomy of classification design with their prominent advantages and disadvantages.

Table 2.5: A common taxonomy of classification design.

Model objective	Advantages	Disadvantages	Examples
Class prediction	Robust Simple	Over-fitting risk Unable to generate new knowledge Curse of data dimensionality and sparsity	WV; DT; SVM; FDA; KNN; ANN
Class discovery	Generate new knowledge Less prone to over-fitting risk	Computationally intensive	HC; K-means clustering; SOM

2.2.3.1 Supervise learning

One of the earliest supervised learning methods in microarray classification is *weighted voting (WV)* classifier proposed by Golub et al. (1999). The WV algorithm is a linear classifier based on the amount of weight

carried by each vote in participating entities (i.e. genes). Each entity is assigned a weighted vote and the magnitude of each vote is dependent on the entity value in the sample and the correlation of that entity's with the class distinction. The weighted vote of gene g is the production of the subtraction of the normalised log gene value x_g ,with the average mean log gene values of all classes b_g and the ratio of mean and standard deviation of the gene a_g based on the measure of signal-to-noise (S2N) ratios. Thus, the formation of the weighted vote of a gene can be expressed as $WV_g = a_g(x_g - b_g)$. The WV algorithm has been commented on, as computationally simple and ability to select genes, however, the downside of this algorithm is that it is designed for binary classification with a lack of exploration on gene dependency.

Golub et al. (1999) applied WV classifier to categorise the new unknown samples of acute leukaemia cancer based on the top-50 informative genes identified by S2N ratios. They observed that satisfactory classification results were achieved with 29 out of 34 test samples being correctly classified. However, Dudoit et al. (2000) commented that an incorrect variance calculation in S2N ratios decreased the efficacy of WV algorithm. Dudoit et al. developed a univariate selection approach based on the ratio of mean and variance of the genes for cancer classification.

The *classification tree*, also known as *decision tree (DT)*, is another commonly used classifier in microarray studies (Lidén et al., 2002; Li et al., 2004; Lee et al., 2005). A DT contains a set of internal nodes and leaf nodes in which the internal nodes associated with splitting criterion (i.e. splitting features and splitting predicates of the features) and the leaf nodes represent individual classes (Lu and Han, 2003). The construction of the tree involves two phases: growing phase and pruning phase. In *growing phase*, the tree is constructed by splitting each feature and its predicates into individual internal nodes to learn a set of general rules in class separability. Once the rules have been learned, *pruning phase* begins by pruning the tree with an IG function to reduce the entropy caused by over-partitioning samples into leaf nodes and to avoid an over-fitting problem. The claimed advantage of the classification tree is that it is easily understood by the user.

Lidén et al. (2002) proposed three types of rule induction criteria to improve the performance of DT, namely divide-and conquer (DAC), boosting and separate-and conquer (SAC) based on LOOCV procedure. DAC is a variant of recursive partitioning technique that used IG to select branching features in generating hierarchical rule sets on the tree, boosting is an ensemble method that used iteratively readjusts the weight distribution of the training samples and SAC is the covering method that iteratively finds one rule that covers a subset of the data rather than recursively partitioning the entire data set. They observed that a DT trained with the boosting technique outperformed DAC and SAC approaches, however, there is no significant improvement in binary classification comparing it to other classification methods, such as SVMs, clustering approach and Bayesian approach using 128 genes. For multiclass classification, a boosting classification tree showed 100%

classification accuracy based on about 20 rules regardless on the number of genes. They also observed that the boosting approach works significantly better in a classification tree rather than decision stumps. This observation has also been noted by Dudoit et al. (2002) and Lee et al. (2005) when they applied classification and regression tree (CART) in binary and multiclass cancer classification.

Despite combining a DT with aggregation methods (i.e. boosting and boostrap), Statnikov et al. (2004) combined the DT with SVMs for multiclass cancer categorisation. They observed that a sole classification tree performs significantly worse than other supervised methods, such as SVMs, KNNs and ANNs, regardless with or without the selection method. Li et al. (2004) also observed that the selection method downgraded the performance of a DT, due to the tree being built by dynamically selecting the most informative features from the data set and selection method, in fact, it removes features from the data set and some informative features may be lost during the data removal process. They used the J48 tree algorithm that was implemented in WEKA environment to discriminate multiclass tumour data.

Fisher's discriminant analysis (FDA) is also widely used for microarray classification. FDA is a non-parametric method, that finds an optimal projection matrix (gene subsets) which reshapes the data set for maximum class separability using the ratio of between-class to within-class matrices (BSS/WSS) (Lu and Han, 2003). Unlike linear discriminant analysis, FDA has the ability to discriminate multiclass data, but, its predictive efficacy is dependent on the number of features in the data set and it ignores the correlation between features (Dudoit et al., 2000).

Dudoit et al. (2000, 2002) compared discriminative ability of FDA with four other supervised methods, namely KNN, linear discriminant analysis (LDA), DT trees and aggregating classifiers for binary and multiclass scenarios. They observed that generalisability of FDA is dependent on gene dimension and sample size. The performance of FDA was significantly improved with a smaller number of genes used for classification. A similar pattern of discriminant methods and observations are also reported by Lee et al. (2005) when they extended the comparison studies with more variations of classification methods, microarray data and selection techniques.

Meanwhile, Cho et al. (2003b) utilised the kernel function to enhance the capability of FDA for multiclass classification. They observed that the kernel-based FDA has better classification results in a multiclass scenario than in a binary classification. They reported the minimum overall mean errors of 0.96% and 4.06% based on 5-fold cross-validation (CV) procedure on multiclass and binary microarray data, respectively.

Naive Bayes (NB), is another supervised classifier that can, generally, achieve good classification performance in most application areas and some bioinformatics areas, including bioassay classification and clinical diagnosis. However, it is not a popular method to be implemented in microarray classification because it is a univariate-based approach. A NB classifier is a statistical-based approach based on the elementary Bayes'

Theorem with strong independence assumptions, i.e. all members (i.e. samples or genes in the context of microarray data) in the data set are unrelated to each other. Given a C number of classes in the data set and s is an observation (i.e. a sample or a gene) in the data set with a known probability density of the class $p(S_c)$. If the class prior probability $p(c)$ is known, then the posterior probability $P(c|s)$ of the observation can be expressed as:

$$P(c|s) = \frac{p(S_c)}{p(S)}.\frac{L}{p(S_c)}. \tag{2.1}$$

L is the predefined number of labelled observations (priori) used to measure the likelihood of the observation, irrespective of their class labels. The main advantage of the NB classifier is due to its over-simplified assumptions which often work efficiently in classification problems, however, this simplicity is also the drawback of the NB classifier when identifying correlated set of genes in the microarray data.

In microarray literature, the NB classifier is normally implemented for binary classification problems, rather than for multiclass scenarios, or is used as the performance evaluator in logistic regression model (LRM). Hwang et al. (2002) implemented the NB classifier to discriminate between two types of acute leukaemia classes. Chu et al. (2005) used the Bayes' Theorem to evaluate the significance level of the genes identified by the Gaussian methods in three binary microarray data sets and Li and Yang (2002) used the Bayesian information criterion (BIC) in LRM to determine the minimal number of genes to be needed for a discriminant microarray analysis.

k-nearest neighbour (KNN), on the other hand, is a similarity-based approach based on the minimum distance measure between the testing samples and the training samples (Lu and Han, 2003). For KNN, k number of training samples (i.e. nearest neighbours) are used to label unknown samples according to the distance measure. For instance, if $k = 1$, the test sample is simply assigned to the class of its nearest neighbour (training sample), if $k > 2$, then the test sample is assigned to the nearest class with the minimum distance. The distance metric can be any similarity measure statistics such as Pearson correlation, Euclidean distance, Spearman correlation and many more. The common advantages of KNN are that it is computationally simple, less prone to noise and bias, however, it is not scaleable and is similar to FDA, its performance is reliant on the number of k points used in the classification process (Lu and Han, 2003).

Dudoit et al. (2000, 2002) performed a comparison study between KNN with four other supervised methods, i.e. FDA, LDA, DT and aggregating classifiers based on binary and multiclass scenarios. They observed that the classification performance of KNN is improved when the number of classes is not large and KNN performs remarkably well compared to DT and aggregating methods, albeit, with a lack of transparency. Li et al. (2004) also found that the KNN classifier outperforms DT in ensemble classification. They indicated that the choice of feature selection methods will, in fact, affect the performance of KNNs in the multiclass scenarios.

However, Golub et al. (1999) argued that the performance of classification methods could indeed be based on the types of expression data when they applied KNN to discriminate genes that are uniformly high in one class and uniformly low in the other, for binary cancer classification. Wang et al. (2006) observed that the interaction between genes in microarray data were not explored by KNN based on a leukaemia cancer classification conducted using the LOOCV procedure and a lymphoblastic origin sample misclassified into myelogenous origin.

Meanwhile, Li et al. (2001a,b) applied GA to select informative genes for KNN for leukaemia cancer classification. They claimed that hybrid GA-KNN is superior to KNN, simply because the hybrid approach implicitly assumes that genes are similarly expressed within each type of sample group, however, this could be problematic when subtypes of cancer exist, resulting in the relevant genes not being uniformly expressed in the group. Ooi and Tan (2003) also commented that the GA-KNN strategy might not be optimal for multiclass scenarios due to the identified genes being based on the n top-ranked genes that were picked from a finite number of gene predictor sets, resulting in little exploration on gene interactions which are believed to be more complex for multiclass scenarios. Furthermore, the distance metrics used to determine the k neighbours became less sensitive, as data dimensional increases and sample data might also be unclassifiable, if no satisfactory majority vote was obtained from the k neighbours.

Artificial neural network (ANN) is the commonly used machine learning approach for clinical diagnosis and it has recently received attention in the bioinformatics field. ANN is a non-parametric method that interconnects numerous artificial neurons (i.e. genes) and processes information using connectionist computation. For ANN, all incoming signals are processed by neurons in the layer and the output signal is forwarded to neurons in the next layer. The predictive strength of ANN can be accelerated by the parameter choices including activation function, learning algorithm, momentum and bias on the gene selection. The common example of supervised-based ANN is a multilayer perceptrons network that contains three layers of neurons, using the backpropagation learning algorithm.

Bloom et al. (2004) applied a 3-layered ANN with backpropagation learning to classify multiclass sample data based on the genes selected using H-test statistic, for two different array platforms, i.e. cDNA and oligonucleotide. They reported a mean test accuracy of 83% and 88% based on a 95% confidence interval for cDNA and oligonucleotide platforms, respectively and, a mean test accuracy of 85% in combining platforms. Meanwhile, Ko et al. (2005) compared ANN with two other supervised methods, i.e. DT and KNN for a multiclass scenario. They also reported that ANN outperforms DT and KNNs based on purity estimation values for network structure of 61-10-21.

Despite using multiple layer networks, Khan et al. (2001) proposed a single-layered ANN for predicting 4-class of SRBCTs tumours based on 96 gene expression signatures selected by the PCA. They reported

100% classification accuracy based on test data with no evidence of over-fitting. Due to the encouraging results reported by Khan et al., their work has been extensively studied by other researchers (Wei et al., 2004; Chen et al., 2007). Wei et al. used a 3-layered ANN with 10-3-1 structure to predict the survival rate of neuroblastoma (NB) patients based on DNA clones selected by PCA, while Chen et al. adopted 40-3-4 structure to diagnose and classify SRBCT tumours using the genes selected based on multiplex RT-PCR assays. Markowetz and Spang (2005), however, commented that perfect separation can be an artifact of high dimension rather than an indication of a biological relationship of genes, in fact, it is a sign of over-fitting.

Keedwell and Narayanan (2003) also used a single-layered ANN for acute leukaemia and myeloma microarray classification based on gene combinations selected by GA. Their work was further extended using only a single-layered ANN to identify gene combinations (Narayanan et al., 2005). In the latter work, myeloma data was pruned from the original 7129 genes to only 21 genes based on 3 network models. The first two models were used to reduce the gene dimension, while the third model was used for classification. They reported 98.1% overall accuracy based on test sample data using See5 tool. When data partitioning approach was applied on the models, 29 genes were reported.

Liu et al. (2004a) examined generalisability of ensemble ANNs, based on three selection methods, which are Wilcoxon ranksum, PCA and t-test, to improve predictive accuracy and robustness of ANNs. Three ANN models, each accommodating different selection methods, were used to train the same set of training sample data and the overall mean accuracy from these models was calculated. The bootstrap technique was used to re-sample the data 100 times, resulting 300 ANN models for each data set. They claimed that the ensemble ANN method performed better, or, is at least comparable, to bagging tree classification.

Schwarzer et al. (2000) discussed several important issues on the improper implementation of ANN which might led to serious consequences in the classification results and they have reported some important common mistakes that have been omitted in the reported results.

Support vector machine (SVM) is another machine learning technique that has recently received attention in the bioinformatics field. It was originally introduced by Vapnik and his co-workers for data mining problems (Lu and Han, 2003). SVMs adopt the structure risk minimisation principle to identify a hypothesis that can guarantee the lowest probability of error (Lee et al., 2005). The underlying principle of SVM is to map the features to a higher dimensional space using a linear function and to identify the maximum-margin hyperplane, i.e. a linear line that can separate samples into two distinct classes. The common advantages of SVM are that it explores gene interactions, scaleable to high dimensional and robust performance, however, a significant drawback is that it is designed for binary classification. The suggested solutions to the problem are either to breakdown the multiclass problem into several binary scenarios, or to iteratively performing binary classification until all classes have been separated (Lu and Han, 2003). However, these suggestions are

computationally intensive. To reduce the computational cost of SVMs, two weighting strategies are proposed for multiclass problems, i.e. one-versus-all (OVA) and all-pairs (AP). The OVA approach builds k number of linear SVMs corresponds to k number of classes in multiclass data and distinguishing one class at a time from all the other classes, while the AP approach distinguishes two linear SVMs from the rest of the classes at one time. Additionally, different kernels applied to different data sets can also improve the classification performance of SVM (Cruz and Wishart, 2006).

Guyon et al. (2002) introduced RFE selection in SVM algorithm to find the optimal gene subset for binary classification. They revealed that the classification results are significantly affected by types of selection methods, rather than by classification methods, based on cross test results obtained with a WV classifier and FDA classifier. However, Li et al. (2004) argued that the accuracy of classification is dependent on the choice of the classification methods rather than on selection methods. They made their comment based on the evaluation of eight different selection methods constructed using the RankGene algorithm and four supervised classification methods on multiclass microarray data.

Peng et al. (2003) applied AP strategy in linear SVMs for each binary classification ramified from a multiclass data set and these results are then combined to form the final result for a multiclass data set based on the gene subsets derived from GA's selection. They observed that the combination of GA and SVM bestows benefit to microarray analysis, including avoiding over-fitting, no priori information on the data set is required and robust to noise. Better performance on GA/AP-SVM is reported compared to KNN, HC and SVM trained with a OVA strategy. Similar hybridisation of GA and AP-SVM also reported by Liu et al. (2005a) and unlike Peng et al. who used RFE to further eliminate the non-predictive features in the GA-derived gene set, Liu et al. used NSC to refine the GA-derived gene set.

Yeang et al. (2001) compared AP and OVA weighting strategies based on the multiclass classification. They noticed that the performance of strategies is dependent on the number of genes defined in the classification process. The AP approach outperforms the OVA approach when a fixed number of genes is applied, however, the OVA approach achieves lower error rates than the AP approach in a LOOCV procedure when a random number of genes is applied. In contrary, Li et al. (2004) commented that the performance of weighting strategies appeared to be problem-dependent and they made their comment based on the experiments conducted on eight multiclass microarray data.

Meanwhile, Shen and Tan (2006), attempted to improve the performance of SVMs using various selection methods, such as BSS/WSS ratios, partial least squares (PLS) and PCA, however, the improvement is not significant.

Instead of refining weighting strategies, some research focuses on the implementation of nonlinear kernel functions on SVMs. For instance, Lee and Lee (2003) developed Gaussian-based kernel function on multiclass

scenarios and presented promising results when the data sets were appropriately preprocessed with adequate selection methods; Statnikov et al. (2004, 2005) used polynomial function to evaluate the performance of AP and OVA strategies; and Mao et al. (2005) applied fuzzy-based kernel function in microarray classification.

Class prediction has been extensively studied in literature. For instance, Boulesteix et al. (2008) reviewed various statistical aspects of supervised classifier evaluation and validation, such as accuracy measures, error rate estimation procedures, selection methods, choice of classifiers and validation strategy. Kotsiantis (2007) reviewed various supervised learning classification techniques that cover the major theoretical issues and possible bias combinations of the techniques. Markowetz and Spang (2005) discussed four supervised learning methods, i.e. likelihood-based methods, DTs, SVMs and regularised binary regression, based on model selection and over-fitting perspectives. Meanwhile, Simon (2003) discussed the key factors to be observed in developing diagnostic and prognostic prediction models based on microarray data and the pitfalls to be aware of in reading reports of microarray-based studies. The statistical issues that arise from the use of microarrays are addressed by Simon et al. (2003). Cruz and Wishart (2006) conducted a detailed review of published studies employing machine learning methods for cancer prediction and prognosis, focusing on the key aspects of the types of methods being used, the types of integrated training data, the kinds of endpoint predictions being made, the types of cancers studied and the overall performance of the methods in predicting cancer susceptibility. Table 2.6 presents a unified view of supervised learning methods reviewed in this section.

2.2.3.2 Unsupervised learning

One of the earliest unsupervised learning methods in microarray studies is *hierarchical clustering (HC)*. HC organises data into a hierarchical tree structure according to the proximity matrix such as Manhattan distance or Euclidean distance, and the result is depicted in a binary tree, known as a dendrogram (Xu and Wunsch, 2005). The root node of the dendrogram represents the entire data set and each leaf node is regarded as a data object (i.e sample) containing information of cluster formation and the correlation between clusters. Two common strategies for constructing dendrograms are the agglomerative (bottom-up) and the divisive (top-down) approaches (Jiang et al., 2004). For the *agglomerative approach*, the dendrogram is overgrown in a massive structure and then pruned layer-by-layer by merging the two closest clusters at each step, until a predefined number of clusters are obtained. The *divisive approach*, conversely, starts with one cluster containing all data objects and at each step, a new cluster is created. The main advantage of the HC algorithm is due to its ability to graphically represent the data set which provides a global view of the distribution of data. However, it is not robust and is computationally intensive (Xu and Wunsch, 2005; Jiang et al., 2004). Zhang et al. (2006) also commented that the HC algorithm is not as robust as the SVM,

Table 2.6: A unified view of supervised learning methods. The * indicates the performance of classifier in dealing with specific task and is range from * (represents poor performance) to *** (represent best performance)

Task	WV	DT	SVM	FDA	KNN	ANN
Strategy	Vote weighting	Entropy function	Maximum-margin	Maximum-likelihood	Similar proximity	Perceptron
Generalisability	**	*	***	**	**	***
Missing value handling	*	**	**	*	*	*
Noise handling	*	**	**	*	**	**
Model transparency	**	***	*	**	*	*
Multiclass handling	*	**	**	**	**	***
Feature selection	distance matrix	backward/ forward search	weight vector	forward search	distance matrix	entropy function
Genes interaction	*	**	***	*	*	***

a supervised learning method, in producing an unbiased estimation of predictive power for new unknown sample data based on the splice recognition problem.

Ross et al. (2000, 2004) and Alizadeh et al. (2000) used the HC algorithm in analysing the variation of gene expression patterns for both binary and multiclass scenarios. For Ross et al. (2000), the HC algorithm is used to group 60 cell lines of an anticancer drug screen (NCI60) with similar repertoires of expressed genes and to group genes whose expression level varied among the cell lines in a similar manner. They conducted the experiments twice using different gene subsets to ensure the robustness of the analysis results and the proximity similarity of genes was computed using the Pearson correlation coefficient. Meanwhile, Alizadeh et al. applied the HC algorithm for lymphoid malignancies clustering based on the basis of the similarity in the pattern with which the expression varied over all samples. Ross et al. (2004), however, developed a 2-dimensional HC algorithm to discover a new sub-cluster of an acute myeloid leukaemia cancer.

K-means clustering is another unsupervised approach based on gene partition and centroid adjustment principles. For K-means clustering, the cluster centroid value is iteratively adjusted each time when a gene is introduced so that the distance between genes within a cluster can be minimised using a proximity matrix. The K-means clustering algorithm commented on, as a fast algorithm converges with a small number of iterations, however, this feature tends to lose out with high gene dimension and noisy data (Jiang et al., 2004). Xu and Wunsch (2005) commented that the iteratively optimal cluster centroids on K-means cannot

guarantee convergence to a global optimum, moreover, K-means clustering is sensitive to outliers and noisy data. Even if a gene is located far away from the cluster centroid, it is still forced into a cluster and thus, distorts the cluster shape (Xu and Wunsch, 2005).

The Kohonen's *self-organising map (SOM)* is an ANN model based on unsupervised learning paradigm. It is a single-layer mapping algorithm in which input and output neurons (i.e. genes) are organised in a two-dimensional map (matrix). Each neuron is associated with a reference vector and each gene point in the matrix is mapped to the neuron with the nearest reference vector (Jiang et al., 2004). The training process of SOM commented on, is more robust than K-means in dealing with high noisy data, however, it requires a predefined number of clusters and the grid structure of the neuron map (Jiang et al., 2004). In addition, SOM may suffer from input space density misrepresentation in which low pattern density genes may be 'buried' by high pattern density genes (Xu and Wunsch, 2005). In this case, SOM is not effective because most of the interesting patterns that merge into only one or two clusters and cannot be identified (Jiang et al., 2004).

Golub et al. (1999) applied binary SOM to automatically cluster acute leukaemia microarray data on the basis of the 6817 gene expression patterns. They observed that SOM was effective, although not perfect, for class discovery with 34 of 38 test samples correctly classified. Wang et al. (2003), however, used SOM to find the optimal map neurons, that can represent the configuration of the input tumour data, and this optimal map was used to classify tumour samples and identify informative genes using fuzzy C-means clustering, a variant of K-means, and pair-wise based FDA.

Jiang et al. (2004) and Xu and Wunsch (2005) presented comprehensive literature on the clustering algorithms. Jiang et al. compared three clustering categories in gene expression, i.e. gene-based, sample-based and subspace clustering methods, as well as the problems pertinent to each clustering category. Xu and Wunsch reviewed clustering approaches for data sets appearing in statistics, computer science and machine learning.

Microarray classification has been extensively reviewed over the last decade. For instance, Mocellin and Rossi (2007) presented the principles underlying the analysis of microarray data, such as cancer classification, microarray data collection and normalisation, gene expression comparison and clustering algorithms. Valafar (2002) conducted a survey on the techniques that have been used in mining microarray data, including missing values imputation, selection approach and clustering method. Lu and Han (2003) presented a comprehensive overview on cancer classification and selection methods, along with an assessment based on aspects such as computation time, classification performance and the ability to reveal biologically meaningful gene information. Kuo et al. (2004) also outlined the main challenges and critical considerations regarding the construction of classification models in gene expression studies, such as modelling techniques and biological

validation of results. Asyali et al. (2006) reviewed class prediction and class discovery that are applied to gene expression data, along with the implications of the findings. Tarca et al. (2007) conducted a survey on the supervised and unsupervised learning methods in the application to biology, along with the methods and examples implemented in the open source data analysis and visualisation R language. Dupuy and Simon (2007) discussed the dispute between microarray-based clinical research and published microarray studies in which gene expression data is analysed in relation to cancer outcomes. They proposed guidelines for statistical analysis and reporting based on the common discrepancies identified.

2.2.4 Feature Selection (FS)

A known challenge in microarray studies is to identify smaller sets of informative genes that are highly associated with the pathogenesis condition of malignancy proliferation from high feature dimension of data. Thus, FS techniques are normally expected in the preprocessing stage. FS is the process of finding a subset of the original features of a data set, either individually correlated or combinatory correlated, and run on data containing only these features, generates a classifier with the highest possible accuracy. Some FS methods have the ability to remove redundant information from microarray data, thus, it is also known as feature reduction in literature. Several claimed advantages of FS include (Saeys et al., 2007; Rocha et al., 2007):

- To provide a focus group of genes that are useful in either biological or classification way.
- To avoid the over-fitting risk and to improve the prediction accuracy of classifiers.
- To provide faster and cost-effective models.
- To reduce the complexity of the classification model.

Table 2.3 on page 32 presents the taxonomy of FS along with its advantages and disadvantages.

2.2.4.1 Filter selection

Filter selection measures the marginal relevance of the individual feature to the known class with standard statistics criteria and is independent of the classification model used. Although the filter selection is simple and efficient, however, it overlooks the relationships between features. Pruning these correlated features in the preprocessing stage seem unimportant, when individually evaluated, but, are crucial for explaining the problems when more than one of these features are taken into consideration.

The simple filter scheme is to compute a mean and a variance, or a standard deviation in the data. For instance, Dudoit et al. (2000) proposed between-group/within-group (BSS/WSS) ratio based on mean and

variance computation between selected features with the known class which is expressed as below:

$$\frac{BSS(j)}{WSS(j)} = \frac{\sum_i \sum_k I(y_i = k)(\bar{x}_{kj} - \bar{x}_j)^2}{\sum_i \sum_k I(y_i = k)(x_{ij} - \bar{x}_{kj})^2}, \tag{2.2}$$

where x_{ij} denotes value of feature j for sample i, $\bar{x}_j$ indicates overall mean value of feature j across all samples in the data set, $\bar{x}_{kj}$ indicates average value of feature j for its class k and I is indicator function in the ratio. Dudoit et al. (2002) commented that the use of a feature variance value in filtering features is more accurate and efficient than using standard deviation of the feature. They made such comments based on the comparison study with S2N ratios proposed by Golub et al. (1999). However, BSS/WSS is not scaleable to high dimension and is sensitive to data magnitude and as a result, feature reduction and normalisation are generally expected to preprocess the data. For instance, Dudoit et al. (2002); Lee and Lee (2003) preprocessed acute leukaemia oligonucleotide data with steps involving thresholding, filtering, log-transformation and normalisation based on a variance, to remove data redundancy and suppress value magnitude before the implementation of BSS/WSS ratios. Meanwhile, Shen and Tan (2006) preprocessed GCM oligonucleotide data and lymphoblastic leukaemia oligonucleotide data by computing a standard deviation in the data and statistic approaches, such as PCA and PLS for feature reduction. Lin et al. (2006), however, attempted to use BSS/WSS ratios to reduce the dimensionality space of genes for NCI60 oligonucleotide data and GCM data which have been preprocessed by zero mean normalisation.

More complex filter selection methods include the use of statistics functions, including PCA and PLS for redundancy reduction and feature extraction (Khan et al., 2001; Cho et al., 2003a; Liu et al., 2004a; Wei et al., 2004; Tan et al., 2005; Shen and Tan, 2006), the use of information gain (IG) matrices to rank features (Cho and Won, 2003), the use of proximity matrices, such as Euclidean distance, Pearson correlation, Spearman correlation and Cosine correlation for feature extraction (Cho and Won, 2003, 2007), the use of t-test and F-test in selecting important features (Inza et al., 2004; Liu et al., 2004a; Mao et al., 2005; Tan et al., 2007) and many more. Amongst these methods, PCA is commonly applied to microarray data.

The principal component analysis (PCA) is a profoundly statistical model implemented in the chemoinformatics field (Gasteiger, 2006; Brown, 2009) that finds a set of orthogonal principal components, i.e. features, based on the eigenvector concept to describe the correlation of sample data in different independent variables and is an effective approach for redundancy reduction without the loss of data characteristics. However, this may be its drawback from a classification point of view because there is no guarantee that the principal component representing the large variance in an independent variable would necessarily be the component most strongly related to the dependent variable (Tan et al., 2005). Unlike PCA, the partial least squares (PLS) maximises the sample covariance between the linear combination of dependent features and the orthogonal

component of independent features (Tan et al., 2005), thus, the relationships between features are taken into consideration in the pruning process.

2.2.4.2 Wrapper selection

A wrapper selection method conducts the selection process with an optimisation algorithm that searches the space of possible feature subsets to find the best subsets of features based on the predictive error estimation returned by a supervised classifier. A typical wrapper approach based on the hybrid GA/ANN is presented in Figure 1.3a on page 11. The idea of wrapper selection is to "wrap" the classifier in an optimisation algorithm that would make a feature subset from the current set of features. This feature subset will continue to grow until the accuracy of the model was no longer more accurate (Kohavi and John, 1997). An objective function is generally defined for wrapper selection that takes into consideration different criteria, such as the accuracy of a classifier that is trained in the feature subset. The number of features to reward the model or the predictive performance of the trained classifier is evaluated by using a separate set of unknown data samples. Wrapper selection does not have the possible shortcomings of filter selection, however, wrapper selection easily over-fits the data and is computationally intensive. In addition, wrapper selection is dependent on the 'wrapped' classifier, thus, there is no guarantee that an optimal feature subset for one classifier will be the optimal for another classifier.

The evolutionary algorithm (EA) is commonly used selection method to "wrap" the classifier to find correlated gene subsets in microarray data. An EA algorithm normally involves two search processes, i.e. *exploration* and *exploitation*. The former process conducts a global search (i.e. heuristic search) on data space for possible regions of feature subsets and the latter process exploits the favourable feature subsets from the region space in which the possible subsets of features are retained using an exhaustive search. The examples of EA include genetic algorithm (GA), simulated annealing (SA) and genetic programming (GP).

EA has been studied in bioinformatics literature as an optimisation algorithm for efficient classification and gene subset selection. Li et al. (2001a,b) used GAs to select an arbitrarily fixed set of 50 and 40 genes for colon and lymphoma cancer classification, respectively, using a consensus KNN classification method. Meanwhile, Deutsch (2003) developed a replication algorithm based on EA with a KNN classifier and better classification accuracy and smaller gene subsets used in the classification were observed. A similar approach is also used by Jirapech-Umpai and Aitken (2005) on acute leukaemia and NCI60 oligonucleotide data. They evaluated the generalisability of a classifier based on separate test sets and a variation of the number of selected genes for classification. Ooi and Tan (2003), on the other hand, used an EA with a maximum likelihood classifier for multiclass scenarios and they defined the objective function based on the independent test error rate returned by the classifier. Liu et al. (2005a) and Peng et al. (2003) also applied similar approaches as used

by Ooi and Tan (2003), however, instead of using a maximum likelihood classifier, they used SVM as the classifier. They used LOOCV procedure as an objective function. The validation of their work is based on a single test sample for each data set. Tan et al. (2007) also used SVM as a classifier in their work, however, instead of using LOOCV as objective function, they applied multiobjective functions in EA, i.e. to maximise the classification accuracy of the feature subsets and to minimise the number of features selected, based on multiple feature subsets produced by multiple feature selection methods. However, there may be a risk of an over-fitting of their results due to an over-pruning of the data and the high complexity of the proposed model.

In addition to conventional classification methods, perceptron-based classification is also proposed with consensus of GA. Keedwell and Narayanan (2003) used a single-layered ANN as a classifier to discover small combinations of genes which lead to the correct classification for acute leukaemia and myeloma data. Similar ANN architecture is also used by Cho et al. (2003a) and Karzynski et al. (2003). Cho et al. defined the objective function based on the prediction results returned by ANN with a 3-fold cross-validation procedure and they evaluated the generalisability of the classifier with a separate blind test set and a variation of the number of selected genes for classification. Meanwhile, Karzynski et al. used GA to find the population of optimal solutions for ANN architecture in where the balance between samples and parameters to train is optimal. Bevilacqua et al. (2006a) and Lin et al. (2006) proposed the use of error estimation returned by the ANN classifier as the objective function in the classification of breast cancer metastasis recurrence and multiple microarray data, respectively. Bevilacqua et al. (2006a) observed that GA/ANN hybrid model is a robust algorithm and would only be affected by a very low variability of results due to the efficacy of GA to focus its search on feature subsets and to avoid local minima entrapment.

2.2.4.3 Embedded selection

Embedded selection is the latter variant of wrapper selection in which the search procedure is built in and is specific to a classification model. Unlike wrapper selection, embedded selection does not guarantee the findings of correlated feature subsets, however, it is being far less computationally intensive than wrapper selection.

The simple scheme of embedded selection is a univariate-based approach that selects features using a mean and a standard deviation computation in the data, such as signal-to-noise (S2N) ratios developed by Golub et al. (1999) for a WV classifier. Given gene g in class c, in order to classify gene g, the mean μ and the standard deviation σ of gene g in class c are computed. $[\mu_1(g), \sigma_1(g)]$ and $[\mu_2(g), \sigma_2(g)]$ denotes the means and the standard deviations of the log of expression levels of gene g for the samples in class 1 and class 2,

respectively. The S2N can thus be expressed as:

$$S2N = P(g, c) = |\frac{\mu_1(g) - \mu_2(g)}{\sigma_1(g) + \sigma_2(g)}|. \tag{2.3}$$

For S2N ratios, large value indicates a strong correlation between gene expression and the distinction class, while the sign of S2N being positive or negative corresponds to gene g being more highly expressed in class 1 or class 2, respectively. Golub et al. (1999) used S2N ratios to select a fixed set of 50 genes, 25 genes for each known cancer class, in acute leukaemia data and they observed a satisfactory validation of the results was achieved in a blind test.

S2N ratios are normally used as a comparison study with other embedded selection methods and supervised classification methods. Takahashi et al. (2005) compared the number of identified marker genes by S2N ratios with projective adaptive resonance theory (PART) for acute leukaemia data and they observed that a total of 10 informative genes is sufficient for classification rather than using 50 genes for class discrimination. Dudoit et al. (2002) compared the statistics criteria of S2N with BSS/WSS ratios and they concluded that the use of the deviation computation is unstable compared to the use of variance computation.

Recursive feature elimination (RFE) is another embedded selection method that assists in improving the classification accuracy of SVMs as proposed by Guyon et al. (2002). RFE was inspired by backward feature elimination in which the removal of multiple irrelevant features by iterating three steps: train SVM, rank features and removes the feature with the smallest rank, each time only one least-fitted feature is removed. RFE is a univariate-based approach that finds an individual feature using the ranking criterion of optimal brain damage (OBD) algorithm. The OBD algorithm approximates the changes in cost function $\delta J(i)$ by expanding cost function J in the hyperplane. The $\delta J(i)$ can thus be expressed as follow:

$$\delta J(i) = \frac{\partial J}{\partial w_i}\delta w_i + \frac{\partial^2 J}{\partial w_i^2}(\delta w_i)^2, \tag{2.4}$$

where w_i is the weight vector for feature i, $J = |w|^2/2$ for linear SVM computation and the weight vector can be computed with the following expression:

$$w = \sum_{i=1}^{N_s} \alpha_i y_i x_i, \tag{2.5}$$

where N_s is the number of support vectors which are defined with $0 < \alpha_i \leq C$, C is the penalty parameter for the error x_i and y_i are data instances in a d-dimensional Euclidean space. For linear SVM, the margin width can be set to $2/||w||$. At the optimum of J, the first order can be neglected and the second order

becomes $\partial J(i) = (\partial w_i)^2$. As removing the feature i, means $\partial w_i = w_i$, the second order w_i^2 is taken as the ranking criterion. By iterating the processes of training SVM, adjusting cost function and removing the smallest weight vector, a smaller set of features with higher weight vectors is yielded.

Guyon et al. (2002) used RFE to find a small subset of genes for acute leukaemia and colon data. They reported only 2 genes are required for acute leukaemia classification without a validation mechanism, however, a total of 64 genes are necessary for the baseline method to get the best result with a LOOCV procedure. For colon data, they identified 4 marker genes. Shen and Tan (2006) applied RFE to optimise multiclass SVMs. The RFE is used to execute each SVM classifier independently to obtain the best performance and the smallest gene subsets. A 3-fold cross-validation is then applied to validate the fitness of the gene subset for each classifier, so that a set of base classifiers with different gene subsets can be constructed for multiclass classification tasks.

Forward selection, also known as regression selection, is commonly embedded in linear discriminant methods and DT. Contrary to RFE selection, forward selection selects one feature at each step, this gives the best prediction accuracy for FDA and DT classifiers in combination with the previously selected features. Forward selection for FDA is proceeded as follows (Park et al., 2007):

1. After k features are selected, perform the steps (a) - (c) for the remaining candidate features:
   (a) Add one feature to the set of k features already selected.
   (b) Perform FDA for the $k+1$ features and project data samples.
   (c) Measure prediction accuracy in the projected space.
2. Find the $k+1$th best feature by choosing the feature which gives the highest prediction accuracy.
3. Repeat the processes to produce gene ranking forwards.

Although forward search is embedded in an FDA classification model, an independent filter selection method is always applied to an FDA model for better classification. Dudoit et al. (2000, 2002) applied BSS/WSS ratios to find a smaller set of gene subsets from the data that had been pruned for microarray cancer classification. Similar approaches were also reported by Lee et al. (2005) for classification in multiclass scenarios. Meanwhile, Park et al. (2007) used forward selection to identify a number of optimal genes to be selected for acute leukaemia microarray classification based on different splitting to the training and test sets, and LOOCV procedure. They observed that it is difficult to determine the number of optimal genes with LOOCV validation due to the potentiality of over-fitting risk in the training process and different splitting of the data also gave a different number of optimal genes. They also compared consistency in gene ranking of forward selection with RFE selection in SVMs and S2N ratios in WV classifier. Based on the top 100

selected genes between different selection methods, they found that forward selection has less than 10% of common genes among the genes selected by the other two methods.

Nearest shrunken centroid (NSC) was proposed by Tibshirani et al. (2002) for prediction analysis of microarrays (PAM) classifier to identify minimal subsets of genes on SRBCT tumour data. NSC shrinks the class centroids for each gene toward overall centroids for each gene with an absolute value Δ after standardising by the within-class standard deviation for each gene. The NSC value is defined as the following equation.

$$\text{NSC} = sign(d_{ik})(|d_{ik}| - \Delta)_+, \tag{2.6}$$

where d_{ik} is a t-statistic for gene i, comparing class k to the overall centroid and each d_{ik} is reduced by an amount Δ in threshold. $+$ means positive part of the t-statistic (if $t > 0$, then $t_+ = t$, else $t_+ = 0$). To compute d_{ik},

$$d_{ik} = \frac{\bar{x}_{ik} - \bar{x}_i}{m_k.(s_i + s_0)}, \tag{2.7}$$

where $\bar{x}_{ik}$ is centroid for gene i in class k, $\bar{x}_i$ is overall centroids for gene i, s_i is the pooled within-class standard deviation for gene i, s_0 is a positive constant value and m_k is defined as follows:

$$m_k = \sqrt{\frac{1}{n_k} + \frac{1}{n}}, \tag{2.8}$$

where n_k is sample number for class k and n is all sample number.

Tibshirani et al. (2002) tested different shrinkage parameters in NSC with 10-fold cross-validation and identified 43 and 21 informative genes based on shrinkage parameters $\Delta = 4.34$ and $\Delta = 4.06$ for SRBCTs and acute leukaemia microarray data, respectively. A high consistency on the selected genes compared to the original study (Khan et al., 2001) was also reported for SRBCTs data. Dabney (2005) commented that NSC computation is wrapped with layers of complexity which can be simplified. He criticised the insertion of 'fudge factor' to each t-statistic's denominator instead of simple t-statistics. A further level of complexity is added to shrink the class centroids toward their overall mean which is not necessary for gene ranking. He presented an alternative gene ranking scheme, namely ClaNC, based on the principle of NSC without a shrinkage or fudge factor component and observed that ClaNC is much simpler and has substantially lower error rates than NSC.

Feature selection/reduction is a vast topic by itself and has been extensively studied in literature. Saeys et al. (2007) reviewed the usage of feature selection approaches in existing bioinformatics domains, including sequence analysis, microarray analysis and mass spectra analysis, as well as upcoming domains such as single

nucleotide polymorphism (SNP) analysis. Meanwhile, Guyon and Elisseeff (2003) presented a wide range of aspects discerning variable and feature selection in microarray analysis. Park et al. (2007) evaluated the consistency of three embedded supervised approaches, namely S2N in WV algorithm, RFE in SVM and forward selection in FDA, in gene ranking under the changes of training samples or different selection criteria. They concluded that the performance of selection methods are dependent on the number of available samples and selection criteria, and this data dependency has posed a dilemma between the necessity and reliability of feature selection in microarray studies. This findings also been noted by Jeffery et al. (2006) when different sizes of samples and features were tested in multiple binary microarray classification. Meanwhile, Osareh and Shadgar (2008) commented that the selection methods, especially the filter selection, is strongly dependent on the size of the selected features rather than the choice of the selection method itself. To alleviate the data sparsity problem, Saeys et al. (2007) suggested two alternatives which could enhance the robustness of the finally selected gene subsets: (1) the external evaluation on the selected genes at each stage of the training process and (2) the use of ensemble selection approaches. Asyali et al. (2006), however, proposed the use of feature extraction to reduce the gene dimension first, before the application of selection approach. However, these alternatives may yield an intensive computational cost and the identification of correlated genes is purely dependent on the choice of selection approaches in ensemble.

Inza et al. (2004), on the other hand, compared six different filter criteria, namely Shannon-entropy, Euclidean distance, Kolmogorov-dependence, Kullback-Leibler, P-metric and t-score for discrete and continuous data with a sequential forward selection procedure which is a variant of the hill-climbing wrapper search in colon and leukaemia studies. They observed that the application of a gene selection approach leads to considerably better classification accuracy results in comparison to the non-gene selection approach, moreover, several common genes have been found in both filter and wrapper approaches which may lead to biological interpretation, albeit, that further investigation is required. Osareh and Shadgar (2008) also observed a significant improved classification performance in a supervised classification method with the application of a gene selection approach. They made such comments based on five supervised classifiers, i.e. SVMs, KNNs, NB, ANNs and DT; in four filter selection methods, i.e. t-statistics, IG, Relief Algorithm (RA) and PCA.

Asyali et al. (2006) commented that there is no clear distinction between feature selection and feature extraction in bioinformatics literature as in pattern recognition literature. They argued that PCA, FDA, SOM and multidimensional scaling are all examples of feature extraction algorithms which have been miscategorised as feature selection or classification methods and this can lead to a poor diagnostics test and biomarker development due to feature extraction algorithms using metagenes (i.e. new genes from the linear combination of some particular genes) instead of the existing genes in the data. However, their argument has not receive wide attention from the bioinformatics field. This may be due to the following reasons:

- The output provided by feature extraction is not so biologically different from feature selection. Thus, the differentiation between these approaches is not important.
- The bioinformatics researchers might not be aware of the difference between feature extraction and feature selection as both methods are commonly used to solve the dimensionality problem in the field. For instance, PCA has proven its usefulness in identifying gene expression signatures for diagnosing SRBCTs tumour (Khan et al., 2001), predicting the survival of neuroblastoma patients (Wei et al., 2004) and differentiating different groups of primary lung tumours (Wang et al., 2006).

To conclude, majority of the bioinformatics literature is favourable on the filter selection approach. This is mainly because such approach has been profoundly used in biology experiments and has been scientifically proven by mathematics scientists on its efficiency in spotting the significance difference between individual features in the data. With the increased understanding in molecular biology, many unknown areas in the past have been discovered.

These areas include the gene-gene interactions (i.e. gene regulatory network) in the microarray data, the gene-gene interactions in different microarray platforms (i.e. cDNA and oligonucleotide), the gene-peptide interactions (i.e. the association between genes and its protein peptides) and the peptide interactions within a protein. As a result, the univariate filter selection became obsolete, as it was unable to explore the interaction between features within the data. The multivariate selection, such as wrapper selection, has become the primary research in the bioinformatics field. Even so, the univariate filter selection method is generally expected as a comparison method to the multivariate selection.

2.2.5 Computing Challenges

The advance of computing technology in cancer classification has created a new era for genetic analysis in the bioinformatics field. However, it also poses some challenges including:

- The lack of understanding of microarray data resulting in an ambiguity of the objectives of the study. Most existing studies have been misconceived by the usage of a classification method. Microarray investigation has been biased towards discovery-based research rather than hypothesis-driven research (Dupuy and Simon, 2007) due to microarray data containing complex interactions between the correlated genes underlying certain biological pathways. Genes with high expression levels in microarrays do not always underly the biological causes, however, its correlated genes with moderate expression levels trigger the problems (Markowetz and Spang, 2005). Thus, hypothesis-driven research based optimal feature sets for classification does not imply that this feature set is highly relevant to the problem (Kohavi and John, 1997), thus flaws in the results occur.

- The risk of over-fitting and biased under-estimates of the error rate due to the misuse of a validation mechanism and resubstitution estimation for classification (Simon, 2003; Simon et al., 2003; Markowetz and Spang, 2005; Dupuy and Simon, 2007). Microarray data has a high gene dimension and small sample size, thus, n-fold cross-validation and sample-splitting procedures are not appropriate for model construction and evaluation. Additionally, the construction of a classification model based on genes selected using all samples in the data set and with no separate set of sample data for validation purpose could lead to an overly optimistic and inflated prediction accuracy. This phenomena is known as *resubstitution estimation*, a common error occurred in classification process. Moreover, some studies presented a dual-validation procedure in which the classifier is validated with a cross-validation procedure and "additional independent samples", have brought more confusion than clarity to the results (Dupuy and Simon, 2007).

- The lack of supporting evidence in the declaration of new prediction models. Many prominent studies made claims for gene expression classifiers and for new classification algorithms based on the invalid result findings from improper cross-validation (Ambroise and McLachlan, 2002; Simon, 2003; Simon et al., 2003), an overly complication of the hybridisation of multiple filter selection approaches on two distinct acute leukaemia specimens by Sethi et al. (2009), selection bias on the selected genes based on the entire data sets by Osareh and Shadgar (2008), and ambiguous validation approach by Chetty and Chetty (2009) in classifying multiclass microarray data sets. A feasible solution to the problem is to conduct a comparative study with other classification algorithms using the same data set and the same validation mechanisms (Simon et al., 2003).

- The unrealisation of the influence of the model complexity to the prediction results which result in model over-fitting (Markowetz and Spang, 2005). As Simon (2003) stated: *"Complex methods with large number of parameters often fit the data used to develop the model well, but provide inaccurate predictions for independent data"*. The large parameters in complex models allow them to fit any kind of problem, but simultaneously, redundant parameters yield confusion in training the classifiers due to the fact that the classifiers are statistically programmed to learn all parameters in the algorithm without knowing the usage of the parameters in the study.

- The unrealisation of the effects of data preprocessing in the findings of the relevant information of the problems. Oligonucleotide microarray data are generally high expression magnitude and most genes are suppressed (i.e. negative expression levels), thus, data preprocessing is generally expected to provide a higher prediction accuracy of classifier. As a result, the 'true' cancer markers are 'buried' by other higher gene expressions and are omitted from classification.

2.3 Summary

Over the past decades, vast developments in the medical field, particularly, in molecular diagnosis. The creation of microarrays to examine genome-wide expression data has provided a global view on the fluctuations of gene expression levels in response to either the physiological alterations or the manipulations of transcriptional regulators of living systems. Such development provides insight into information of the interaction of biological behaviour for both normal and diseased tissues which are ultimately important for the design of an effective treatment therapy for patients.

This chapter presents the relevant literature covering both the biology and the computing perspectives on microarray studies. The related literature has been reviewed and shows that the research area of microarray production and marker identification is still immature, as a result, technical aspects pertaining to these two domains have been exposed. Comparisons on capabilities and limitations for microarray design, selection techniques, classification methods and validation mechanism are shown in Tables 2.2 - 2.4. The challenges pertaining to biology and computing communities are also discussed in Sections 2.1.6 and 2.2.5. Four major points from the literature are summarised as follows:

Gene expression measurements. The variation in gene expression measurements is inevitable due to a lack of control in microarray production (see Section 2.1.6). Care in every phase of microarray production, from sample preparation to image analysis, can minimise the impact of these errors but does not completely prevent them. Thus, errors are generally expected in the finished microarrays.

cDNA microarray handling. The low production cost on cDNA microarrays has led to the inconsistency of microarray handling in individual research laboratories due to different laboratories adopting different approaches in conducting microarray experiments (see Section 2.1.6).

Computational Analysis on microarray data. Most existing studies have an ill-conceived hypothesis in microarray classification by treating it as an ordinary data set (see bullet points 1-4 in Section 2.2.5). Microarray data is distinct from any clinical correlative studies and statistically-based classification studies in which a large number of samples are analysed with respect to a limited number of features. Therefore, attention to the choice of computing methods and the hybridisation of selection techniques, classification methods and validation mechanism are advisable to minimise the over-fitting problem.

Data Preprocessing. Data preprocessing on oligonucleotide microarray data may result in the suppression and exclusion of the true cancer markers from the marker identification and the classification process (see Section 2.2.1 and bullet point 5 in Section 2.2.5).

This thesis describes an intelligent gene extraction method using hybrid GAs and ANNs to efficiently extract

differentially expressed genes for a specific cancer pathology. Chapter 3 presents the methodology used for constructing a feature extraction model for this thesis.

Chapter 3

Experimental Methodology

Chapter 1 described the problems pertaining to microarray analysis using a computational algorithm and our approach to solving them. Related literature on the production of microarrays and the computing methods is reviewed in Chapter 2, along with the challenges posed to both the biology and the computing communities.

In this chapter, a feature extraction model is designed using machine learning methods to overcome the computational problems addressed in Section 1.2. The existing GA/ANN hybrid models emphasise effective classification and overlook the level of complexity of feature extraction and the influence of data preprocessing of the classification results. The microarray data, specifically the oligonucleotide data, will generally require data preprocessing techniques, such as to scale down the magnitude of the feature values and to convert the negative values into positive values, before the computing analysis takes place. Thus, inevitable errors in gene variability results from such techniques occur. Our approach tackles the problem yielded by the normalisation process in the data preprocessing stage. The novelty of our approach is its simplicity, as it follows the Ockham's Razor principle, which can avoid the risk of gene variability errors that yielded by data preprocessing techniques that may alter the quality of the data to be analysed by a computing analysis model. Our approach is referred to the overview as illustrated in Figure 1.3b on page 11.

This chapter contains five sections. Section 3.1 describes the acquisition of empirical data to be used in supporting the theme of this thesis. Section 3.2 discusses the issues concerning the design and building of the feature extraction model, taking into consideration the hybridisation of GAs and ANNs techniques, the model simplicity and the adequacy of the parameters in deriving satisfactory results. Section 3.3 presents the genomic analysis tool, namely GenePattern, which is used to visualise the gene findings of the model. Section 3.4 presents the validation mechanism via the NCBI Genbank and the SOURCE search system, which is used to validate the gene findings of the model. Lastly, Section 3.5 provides a summary of the chapter and to what follows next in the thesis.

3.1 Empirical Data Acquisition

The bloom of the Internet in the 1990s has produced a fully developed bioinformatics field with a massive cyber-space to store microarray data and allow microarray integration from various sources to improve its quality. The aim of this thesis is to formulate, from the identified computing-related problems stated in Section 1.2 on page 7, an innovative feature extraction model, namely Genetic Algorithm-Neural Network (GANN), for extracting informative and relevant features using GAs and ANNs. Two benchmark microarray data sets, namely acute leukaemia (ALL/AML) and small round blue cell tumours (SRBCTs), were obtained from its original public repositories and have been used to examine our model. Due to the different data format in microarray data set than the ordinary data set as showed in Figure 1.1 on page 4, a specially written C++ program is used to transpose the microarray data sets. Two designed synthetic data sets characterised with high feature dimension and small sample sizes in response to the research goal of this thesis, C++ programming was used. Furthermore, two bioassay data sets were obtained from its original source to evaluate the generalisation ability and the robustness of our model to handle large, imbalanced and different data representation data sets.

The summary of the microarray data sets, the synthetic data sets and the bioassay data sets to be used in supporting this thesis are presented in Table 3.1. Figures 3.1-3.5 present data distributions for each data set based on the multidimensional scaling (MDS) in two-dimensional R plots. The screen shot to construct a two-dimensional plot on the R environment is presented in Figure 4.3 on page 97. To avoid any form of confusion on both the biology and computing fields, the original sample names or descriptions in the data sets were used in this thesis.

3.1.1 Microarray Data Sets

3.1.1.1 Acute leukaemia (ALL/AML)

The acute leukaemia (ALL/AML) microarray data is a 2-class high-density oligonucleotide data set that was originally presented by Golub et al. (1999) to evaluate the predictive accuracy of a weighted voting (WV) classifier based on the marker genes selected using the S2N ratio. This data set contains 72 samples, collected from patients, both adults and children, and were sourced from peripheral blood specimens and bone marrow samples, distributed into *acute lymphoblastic leukaemia (ALL)* and *acute myelogeneous leukaemia (AML)* classes. Amongst 72 samples, 47 samples were associated with ALL tumour and the remaining 25 samples in AML. All samples were prepared using Affymetrix technology and contained 7129 probes for 6817 human genes.

The ALL/AML data set is available online from the Broad Institute that are in partnership with Harvard

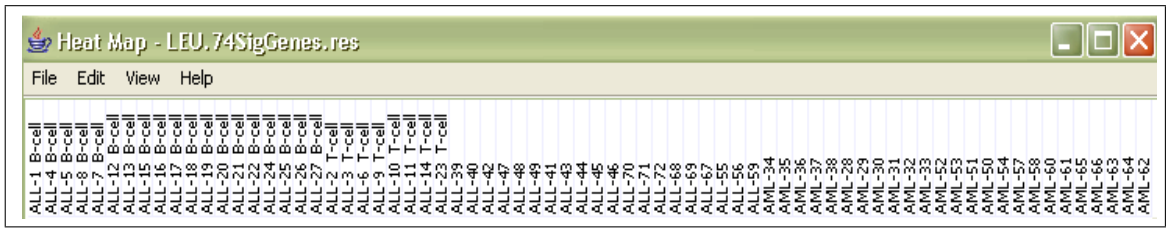
Table 3.1: The summary of the experimental data sets.

Data set	Description	Objective
	Microarray Data	
Acute leukaemia (ALL/AML)	2 classes, i.e. ALL and AML. A total of 72 samples in which 47 samples are expressed in ALL class and the remaining 25 in AML. Each sample is associated with 7129 genes.	To identify relevant informative genes underlying the pathogenesis of acute leukaemia tumours.
Small round blue cell tumours (SRBCTs)	4 classes, i.e. EWS, BL, NB and RMS. A total of 83 samples in which 29 samples are expressed in EWS class, 11 in BL, 18 in NB and the remaining 25 in RMS. Each sample is associated with 2308 genes.	To identify relevant informative genes underlying the pathogenesis of 4 types of small round blue cell tumours.
	Synthesized Data	
Synthetic data set 1	2 classes, i.e. Class 1 and Class 2. A total of 100 samples equally distributed in each class. Each sample is associated with 10000 features.	To determine the minimum parameter setting of the model.
Synthetic data set 2	3 classes, i.e. Class 1, Class 2 and Class 3. A total of 67 samples is distributed with 20 samples in Class 1, 30 in Class 2 and the remaining 17 in Class 3. Each sample is associated with 5000 features.	To simulate a real-world data set that contains a complex level of feature interactions, high dimension of noisy data and inequality distribution of samples size for each classes in multiclass scenario.
	Bioassay Data	
AID362	2 classes, i.e. active and inactive. A total of 4279 compounds, i.e. 60 active compounds and 4219 inactive compounds, and each compound is associated with 144 attributes (1.4% minority class).	To identify active compounds for Formylpeptide Receptor Ligand Binding assay
AID688	2 classes, i.e. active and inactive. A total of 27189 compounds, i.e. 248 active compounds and 26941 inactive compounds, and each compound is associated with 153 attributes (0.91% minority class).	To identify active compounds for Yeast eIF2B assay.

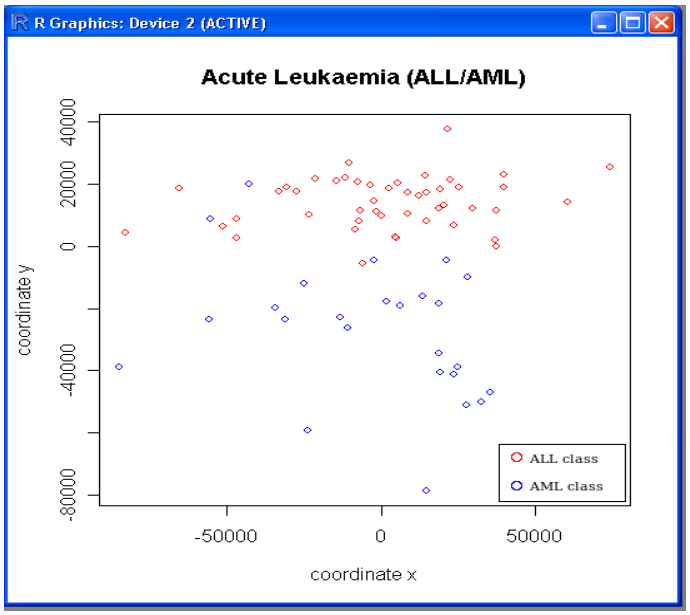
University and its affiliated hospitals, the Massachusetts Institute of Technology and the Whitehead Institute for Biomedical Research. The data set is free to download from the Broad Institute website (ALL/AML oligonucleotide microarray data, 2007). The online source contains 7129 genes including 312 expressed sequence tags (ESTs) in which 59 are control tags.

Figure 3.1 presents the distribution of samples on 2 distinct leukaemia classes. From the plot, 25 AML samples are scattered distantly with a few samples overlapping in the ALL class. Forty-seven ALL samples

have a rather distinct cluster except that a few samples are distance away from the cluster. The reason for the spread of ALL samples may be due to the fact that there are 2 different lineages of ALL tumours in the collected samples, i.e. T-cell and B-cell, which may have similar symptoms but have a distinct genetic alteration, as is showed in Figure 3.1. Both T- and B-cells are two types of lymphocytes (i.e. white blood cells) which cause lymphoblastic leukaemia. B-cells make antibodies and T-cells make other infection-fighting substances. Due to no distinguishing of T- and B-cell ALL samples in the downloaded data set, and to avoid sample mislabelling, all ALL-related samples are grouped under a generic name, i.e. ALL, despite the genetic differentiation. Even so, the data set is fairly linearly separated.



(a) The description of 72 samples in the ALL/AML data set. There are 25 AML samples and 47 ALL samples (19 were B-cell ALLs, 8 were T-cell ALLs and 20 unspecified).



(b) A two-dimensional plot on the sample distribution in the ALL/AML data set. The coordinates x and y reflect the distance (dissimilarities/similarities) of the samples using the MDS function in R Project. The ALL class has a rather distinct cluster compared to the AML class and only a few ALL samples are distance away from the cluster. There are only few overlapped samples from both classes.

Figure 3.1: The acute leukaemia (ALL/AML) microarray data.

The ALL/AML data set has been commonly used in examining the prediction accuracy of classification methods and it is, generally, altered in some way to facilitate the classification process, for instance, data normalisation is generally expected to remove any incompetent information for better classification results.

Table C.1 in Appendix C shows some of the related works in the ALL/AML data set. Meanwhile, some studies reduced the number of genes to be analysed in the classification process (Dudoit et al., 2000; Lee and Lee, 2003; Mao et al., 2005). As a result, different conclusions can be drawn from these studies as the relevant marker genes may be removed at the pruning stage. Culhane et al. (2002) further differentiated ALL samples according to T-cell and B-cell linkages, and outlined the list of the distinct set of genes for each linkage class. It is a good intention to provide an insight into the information for these linkages, provided that there are sufficient samples to draw a firm conclusion. However, it is not feasible in this data set as only 8 out of 47 ALL samples were labelled as T-cell ALL, while the remaining are labelled as generic ALL.

Due to the rapid development of microarray technology, some gene annotations are no longer supported by the NCBI Genbank. Therefore, to avoid any sort of confusion, the original annotation will be displayed in our findings and will be cross-referenced with the SOURCE search system and the NCBI Genbank.

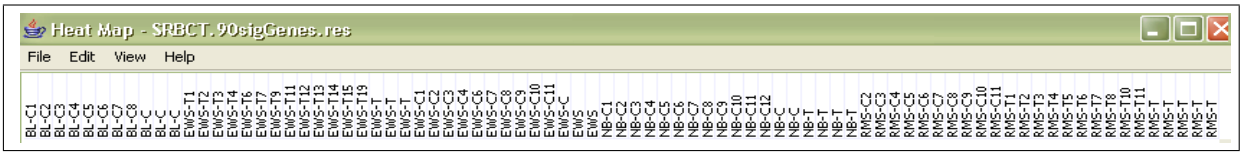
3.1.1.2 Small round blue cell tumours (SRBCTs)

The SRBCTs cDNA microarray data was originally presented by Khan et al. (2001), with the intention of identifying marker genes that distinguish 4 types of round cell tumours of childhood which often masquerade as each other using ANN classification models in conjunction with the PCA. This data set contains 88 samples, each associated with 6567 probes that were filtered to 2308 genes. The 4 types of tumours are *Ewing's sarcoma (EWS)*, *rhabdomyosarcoma (RMS)*, *neuroblastoma (NB)* and *Burkitt lymphoma (BL)* which are collected from two different biological sources, i.e. cell lines and tissue samples, that were prepared according to standard NHGRI protocol. Amongst 88 samples, 29 samples were expressed in the EWS class, 25 in RMS, 18 in NB, 11 in BL and the remaining 5 samples were blind tests.

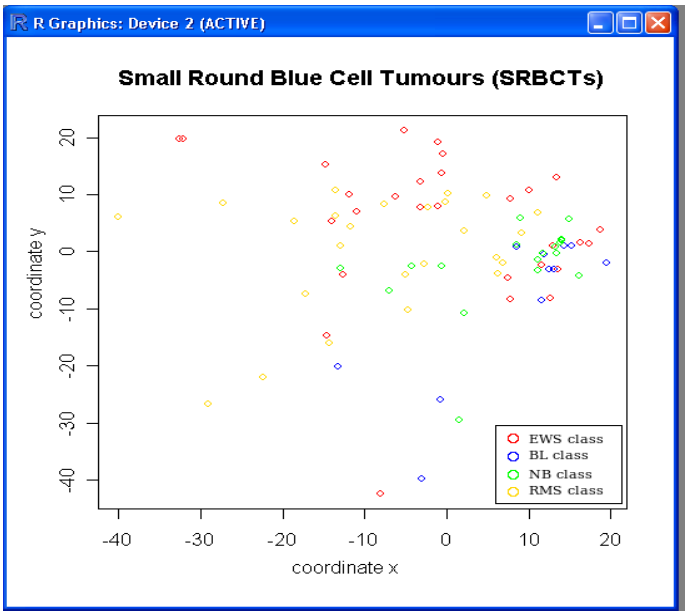
The SRBCTs data set is available online and is free to download from the NHGRI Institute website (SRBCTs cDNA microarray data, 2007). Due to the high dimension of redundant genes expressed during *transcription* process, Khan et al. had reduced the number of genes from the initial 6567 genes to 2308 genes using filtering technique. Therefore, the online version contains only 2308 genes with 83 SRBCTs-related samples and the 5 blind test samples were not published online.

Figure 3.2 shows the distribution of 83 SRBCTs samples on four known classes in the data set. The SRBCTs plot is much more complex than the ALL/AML data set as it involves multiple tumours which often react to similar therapies. From the plot, EWS and RMS classes have a lot of shared features, most probably because both EWS and RMS sarcomas are connective tissue-related cancers, that cause the proliferation of mesoderm and they can be treated with similar prescribed drugs. The BL class, meanwhile, has a distinct cluster rather than other classes, probably as it was collected from lymphocyte cells. RMS class has the widest spread of the samples due to there being 2 subgroups of RMS tumours in the collected samples, i.e.

embryonal and alveolar, which have similar phenotypical symptoms but with a distinctive genetic alteration. The plot also showed that both the sarcoma-related samples and NB samples were scattered distantly. This might be due to these samples were collected from different biological sources. Unlike the other, BL samples were all collected from cell lines. Some related works on the data set is presented in Table C.2 in Appendix C.



(a) The description of 83 samples in the SRBCTs data set. There are 11 BL cell lines samples, 29 EWS samples (16 were tissue samples, 11 were cell lines and 2 unspecified), 18 NB samples (14 were cell lines and 4 were tissue samples) and 25 RMS samples (10 were cell lines and 15 were tissue samples).



(b) A two-dimensional plot on the sample distribution in the SRBCTs data set. The coordinates x and y reflect the distance (dissimilarities/similarities) between samples using the MDS function in R Project. The BL class has a distinct cluster compared to the other classes. The RMS class has the widest spread of the samples, as well as the NB class. The EWS and RMS classes have a lot of common features as both EWS and RMS are connective tissue-related cancers.

Figure 3.2: The small round blue cell tumours (SRBCTs) microarray data.

3.1.2 Synthetic Data Sets

To evaluate the performance of GANN and to ensure the desired set of features to be identified, two synthetic data sets were created using the C++ programming to assess the integrity of GANN in performing a task. Thirty features were predefined on each data set. All feature values excepting the predefined features, were standardised with zero mean ($\mu = 0$) and unit standard deviation ($\delta = 1$). The 30 predefined features on

each data set were assigned with different μ values, ranging from 0.5 - 2.0.

3.1.2.1 Synthetic data set 1

The synthetic data set 1 is a 2-class data set that contains 100 samples equally distributed in each class. For the synthetic data set 1, each sample is associated with 10000 features. Amongst the 30 predefined features, indexed from 1-15 and 5001-5015 were standardised with the mean value of 2 ($\mu = 2.0$). Similar to the ALL/AML data set, the synthetic data set 1 can be linearly separated and does not contain complex feature interactions. Figure 3.3 presents the sample distribution of the synthetic data set 1. This data set is designed to simulate the gene interactivity pattern of ALL/AML data set and is used to determinate the minimal parameter setting of our model.

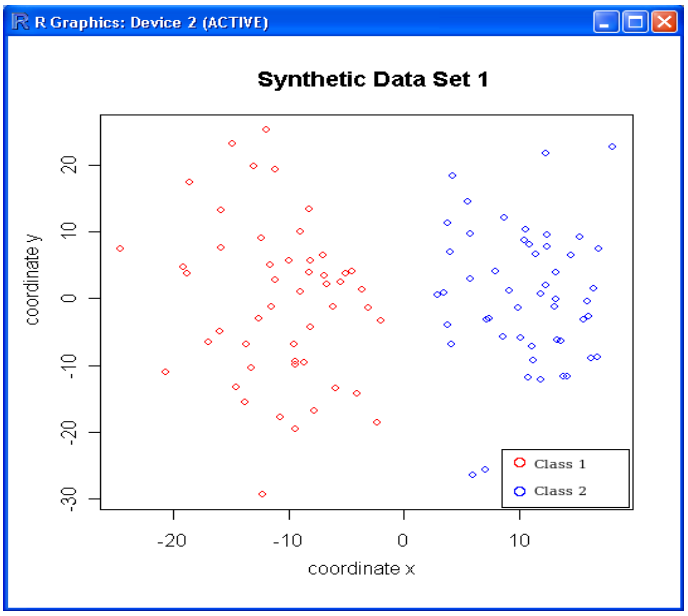


Figure 3.3: A two-dimensional plot on the sample distribution in the synthetic data set 1. The coordinates x and y indicate the distance (dissimilarities/similarities) between samples using the MDS function in R Project.

3.1.2.2 Synthetic data set 2

The synthetic data set 2, as plotted in Figure 3.4, contains 67 samples distributed into 3 distinct classes, with 20 samples for class 1, 30 samples for class 2 and the remaining 17 samples for class 3. This data set is designed to simulate the complex feature interactions in the multiclass scenario that contains a high dimension of irrelevant information and inequality distribution of sample patterns available for each class. For the synthetic data set 2, each sample contained 5000 features. Amongst the 30 predefined features, indexed from 1-10 were standardised with the mean value of 0.5 ($\mu = 0.5$) and the remaining 20 features, indexed from 11-30 were standardised with $\mu = 2$. Similar to the SRBCTs data set, this data set has a

complex level of feature relationships and there are no clearly distinct class clusters among these classes. Class 1 shared features from the remaining 2 classes, while Class 2 has a widest spread of sample cluster and, some of the samples overlap with Class 3.

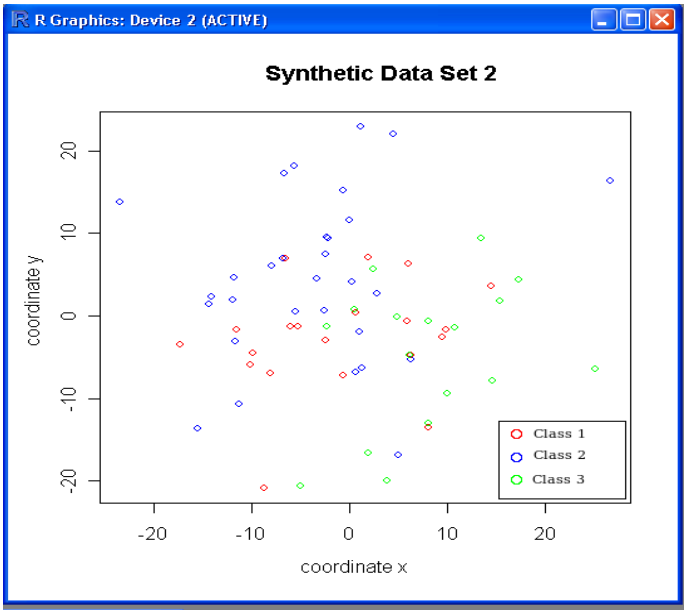


Figure 3.4: A two-dimensional plot on the sample distribution in the synthetic data set 2. The coordinates x and y indicate the distance (dissimilarities/similarities) between samples using the MDS function in R Project.

3.1.3 Bioassay Data Sets

Bioassay, i.e. biological assay, is a scientific procedure for the determination of the concentration of a particular substance of a mixture. It is designed for chemoinformatics field and is essential in the development of new pharmaceutical drugs as it estimates the effect of a substance on living organisms. Unlike the bioinformatics field that focuses on the sequence of DNA data (i.e. genes expression), the chemoinformatics field studies the chemical structure of small molecules (i.e. chemical compounds) (Gasteiger, 2006; Brown, 2009). A microarray data normally involves multiple 'targets' (i.e. growth factor receptors, kinase, oncogenes) applied to a 'subject' (i.e. colon cancer, diabetes), a typical bioassay involves only one target (i.e. an ingredient in a pharmaceutical drug, a substance of a vitamin) in each subject (i.e. active screening drug compounds for anti-epileptic activity, effects of sodium phosphates substance in the betnovate cream). In this thesis, we used two primary screening bioassay data to evaluate the generalisation performance of our model.

A screening process is a technical analysis of a biological specimen, such as urine, blood, saliva; to determine the presence (i.e. active) or absence (i.e. inactive) of the compounds in a target. The term *primary screening* in the context of bioassay represents the first-hand information on a target of interest and is usually involves thousands of unfiltered compounds (i.e. samples) bounded to a set of conditions, known as *attributes* (i.e.

features). Due to the majority class of inactive compounds in the primary screening bioassay data, the number of *false positives (FPs)*, i.e. misclassification of an inactive compound as an active compound, are enormous. This led to the implementation of selection methods to identify the most significance of attributes that can efficiently discriminate between the active and the inactive compounds in bioassay data sets. A typical bioassay data set contains two generic classes, i.e. *active* (i.e. positive) and *inactive* (i.e. negative), and tends to be large and highly imbalanced, i.e. the ratio of 1 active compound to 1000 inactive compounds (Schierz, 2009).

Many selection methods were applied to reduce the number of attributes in the data set. However, not many selection methods are capable to select the most significant attributes from a bioassay data set, mainly due to the multiple data representation in the data set and the data are highly imbalanced. Many of the methods have considered only qualitative information (Brown, 2009) of the data set, i.e. active or inactive of an attribute against a particular compound (binary value), but not the quantitative information of the data set, i.e. the concentration level of a particular compound (real number).

In this thesis, two virtual bioassay data sets, namely AID362 and AID688, were used to assess the generalisability of our model in handling highly imbalanced data which containing multiple data representation, as well as the extraction capability of our model in finding quantitative attributes from the bioassay data. These data sets were originally introduced by Schierz (2009) to study the efficiency of the cost-sensitive classifiers, i.e. Naive Bayes (NB), Support Vector Machine (SMO), C4.5 Tree (J48) and Random Forest, that were developed on the WEKA environment, to identify active compounds in the data sets. The summary of the data sets is presented in Table 3.1 on page 59.

3.1.3.1 AID362

The AID362 data set is a relatively small data set in the context of bioassay that details the results of a primary screening bioassay for Formylpeptide Receptor (FPR) Ligand Binding assay. It contains 4279 compounds, i.e. 60 active compounds and 4219 inactive compounds, with a ratio of 1 active compounds to 70 inactive compounds (1.4% minority class) and 144 attributes. Amongst 144 attributes, 3 were presented in integer numbers, 27 were real numbers and the remaining 114 were binary numbers. Figure 3.5 presents the compounds distribution of the AID362 data set.

3.1.3.2 AID688

The AID688 data set is a large data set that details the results of a primary screening bioassay for Yeast eIF2B assay. It contains 27189 compounds, i.e. 248 active compounds and 26941 inactive compounds, with

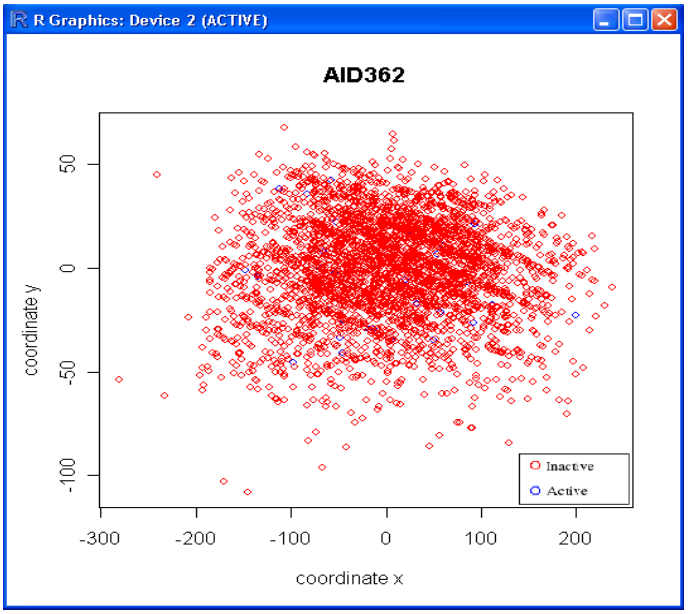


Figure 3.5: A two-dimensional graph on the compound distribution in the AID362 data set 1. The coordinates x and y indicate the distance (dissimilarities/similarities) between compounds using the MDS function in R Project.

a ratio of 1 active compounds to 108 inactive compounds (0.91% minority) and 153 attributes. Four out of 153 attributes were presented with integer values, 27 were real numbers and the remaining 122 attributes were binary numbers. Due to the enormous amount of compounds in the data set, the R Project was unable to simulate the interaction between compounds in the data set.

3.2 Designing Feature Extraction Model using GAs And ANNs

In designing the feature extraction model using GAs and ANNs, the model simplicity, representativeness and adequacy are essential to the reliability of the research to be carried out. Simplicity of the model depends on the way to hybridise two different techniques in which only the simplest theory that fits the fact of a problem is to be considered. In our case the hybridisation of GAs and ANNs with minimal parameters were involved in the algorithm. Meanwhile, representativeness of the model depends on the quality of the data which must represent the type of information that is being investigated, in our case the microarray data sets, synthetic data sets and bioassay data sets.

In this section, we first look at the machine learning techniques to be used in supporting our research. We then discuss the technical issues in designing our model so that a set of meaningful features can be identified.

3.2.1 GA - An optimisation search method

GAs are optimisation methods, in the context of machine learning, that have their inspiration in the analogy of biological evolution. The GA as a search algorithm operates on the basis of a population of potential solutions, rather than a search from general-to-specific solutions or from simple-to-complex. GAs generate, over time, successor solutions by repeatedly mutating and recombining parts of the best currently known solutions (Mitchell, 1997b) with the assistance of the mechanics of selection and natural genetic mechanism. The quality of the solutions in the population will be calculated using a fitness function. For instance, if the learning task is to formulate a treatment strategy for thyroid dysfunction patient, then the GA's fitness could be defined as the number of time a thyroid effective plans that had successfully cured the patient and the risk of side effects of the plans. Over decades, different techniques were implemented in GAs, however, they typically share and iterate the following five processes:

1. Establish the population containing a finite number of potential solutions to the problem.
2. Evaluate individual solutions in the population according to the fitness function.
3. Generate a new population by selecting the most fit solutions from the population.
4. Create new solutions by applying genetic operations.
5. Replace current solutions with new solutions if necessary.

In order to construct a GA search model, the following five technical aspects are considered:

1. A way of representing potential solutions to a problem, as individuals, that can be manipulated by evolutionary mechanism *(population)*.
2. A *fitness function* that can be used to assess the quality of individuals in solving the problem.
3. A method to select better individuals to create a new generation *(selection)*.
4. A mechanism for genetic change so that the individuals in a new generation will not be identical with the individuals in the current generation *(crossover & mutation)*.
5. A way of encoding individuals into the computer environment *(encoding scheme)*.

3.2.1.1 Population of potential solutions

A GA population contains solutions that can be constructed as a *chromosome* which contains all the information needed to describe the problem. The individual unit from which each chromosome is constructed is

referred as *genes* that represent a unique characteristic to the problem. For instance, if the learning task is to formulate the treatment strategy for thyroid dysfunction patients, the chromosome that defines a possible solution to the problem could be formed from the considerations: patient age, phenotypic symptoms, gender, brain neurotransmitters' activities, the hormonal activities and the neuropeptides activities.

The size of a GA population and the chromosomes are normally fixed, as a result, the existing chromosome in the population has to be removed when a new chromosome is introduced by a GA. Cartwright (2008b) commented that the success of a GA does not critically rely on the exact size of the population provided that the population is not unreasonably small. However, it relies on the evolution of solutions which only occurred in GA generations. Too small or too large a population makes proper evolution become impossible and leads to computational cost problems. Cartwright suggested the ideal population size, in most problems applied, should be in the range of $40 - 100$ and this will lead to a fast convergence of the solution and is computationally time effective. However, DeJong and Spears (1991) discovered that the choice of population size had a strong interactive effect on the evolution operator, the crossover operator to be specific, even after augmenting the crossover operators. With smaller population sizes, crossover productivity effects were drastic and caused chromosomes to become homogeneous more quickly. With larger population sizes the crossover productivity effects are far less dramatic. This is due to a lack of information capacity of a smaller population size which provides accurate sampling. As a result, the optimum solution to the problem may be overlooked in the process of initialising the population.

To validate the implication of the population to the efficiency of a GA to perform the task, we examined 3 population sizes and the findings were presented in Chapter 5.

3.2.1.2 Fitness of potential solutions

Most real-world problems are multifaceted and full of trade-offs, several factors contributing to measure the quality of each potential solution in the population can easily be conflicting and it is not always easy to distill them down to a single factor (Cartwright, 2008b). For instance, if the learning task is to learn a strategy for playing chess, the quality of the solution can only be based on a single factor, i.e. the number of games won by the individual when playing against another individual. However, if the task is to formulate a treatment strategy for thyroid dysfunction patient, the quality of the solution could not be based on a single factor, instead, multiple factors should be considered, such as the duration of treatment (time element), the current state of immunological activity of the patient (health condition element), the usage of treatment preparations (preparation element) and the cost of treatment (financial element). These factors, in most cases, are conflicting, such as a trade-off between prescribed drugs for patients who have depression as a side effect with the natural preparations, as the prescribed drugs deteriorate the depression symptoms, although

the prescribed drugs provide effective effects in patients; and the treatment duration for natural preparations are much more longer than prescribed drugs, but its have lower side effects.

In the GA, the quality of a chromosome in solving the problem is computed using a *fitness function* which yields a quantitative measure on how close the individual units in a chromosome to the desired solution that is not defined by GA, instead, it is decided by the use upon a suitable relationship for each problem. For the task to formulate a treatment strategy, the fitness function will include a component that scores the progression estimation of each treatment stage using the historical records of the patient, along with any side-effect problems related to the patient. Normally, the GA fitness function rewards the better chromosomes with a higher fitness value than the poorer chromosomes so that the better chromosome can be further processed by the evolutionary mechanism and has a better chance of survival in the next generation.

Recent research has concentrated on improving GA fitness functions with either the use of a penalty function (Beasley et al., 1993; Busetti, 2001) or the incorporation of machine learning algorithms (Li et al., 2001b; Bevilacqua et al., 2006b; Lin et al., 2006), however, most research is biased towards classification problems and a trade-off of the computational time and/or the algorithm complexity for a better classification accuracy, is generally to be expected.

In our approach (see Section 3.2.3 for detailed explanation), we keep as minimal parameter settings as possible in our model and we defined our model's fitness function as the number of correctly labelled samples in the data sets.

3.2.1.3 Selecting potential solutions

The selection process begins when the fitness has been calculated for all individual solutions in the population. Generally, the selection process is partially stochastic and biased towards better chromosomes in order that the GA can move forward. This is because if the selection was completely deterministic, the population would soon be dominated by the fittest chromosome and would quickly become homogenous before the desired solution is reached (Cartwright, 2008b). This phenomenon is known as *premature convergence*. However, if the solution did not have some guidance for selecting a fitter chromosome, the search would be largely random and the selected chromosome might not be the fittest chromosome, instead, it could be the poorest chromosome in the population. The process will then further deteriorate in subsequent evolutionary operations. Two widely adopted selection mechanism in a GA are the roulette wheel and the tournament selection.

Roulette wheel selection, also known as *proportionate selection*, ranked the GA chromosomes based on their fitness proportions in the current population. For roulette wheel selection, every individual chromosome is

assigned a slot, sized on the proportion of its fitness, on a virtual board. The better chromosome, normally, has a larger slot than the poorer chromosome. The wheel is then spun and the chromosome, into whose slot the virtual ball falls, is copied into the parent pool, i.e. the repository in which the chromosomes have a chance to mate. The selection process is repeated to pick the complement chromosomes until the parent pool is full. Roulette wheel selection leads to the fast convergence of chromosomes with larger fitness proportions being more likely to be picked than those chromosomes with smaller proportions, however, it cannot guarantee that the selected chromosomes are optimal. In addition, roulette wheel selection lacks stochastic power as the population is easily dominated by fitter chromosomes which, consequently, leaves an insufficient resource for the genetic mechanism to further exploit the population, resulting in the loss of better chromosomes being found.

Tournament selection, on the other hand, ranked the GA chromosomes based on the competition basis of at least two or more chromosomes. For a typical tournament selection, two chromosomes are randomly chosen from the population and compared. The chromosome with the greater fitness is selected and copied into the parent pool. The selection process is repeated, to yield a group of competent chromosomes in the parent pool. Since the tournament selection randomly picks chromosomes, the consequent results may vary each time the process is performed and fitter chromosomes may participate more than once in the competition. Even so, tournament selection often yields a more diverse population than roulette wheel selection (Mitchell, 1997b) and it leads to deeper exploitation of the chromosome search. A known benefit of tournament selection is that it provides a certain level of confidence of the selected chromosome being fitter than those not being picked. In addition, it also guarantee that the poorest chromosome will never be selected. The downside of this method is that it takes a longer time to identify fitter chromosomes than roulette wheel selection.

Cartwright (2008b) noticed that both roulette wheel and tournament selections have a lack of stochastic features, as a result, neither can guarantee that the best chromosome, in the current generation, will be chosen again in the next generation. Therefore, a new hybrid-based selection was introduced to overcome the pitfall of roulette wheel and tournament selections, i.e. stochastic remainder selection.

Stochastic remainder selection is a hybrid method that combines a stochastic element with a deterministic step to ensure that the best chromosome in the current generation is never overlooked in the next generation. In stochastic remainder selection, the fitnesses of chromosomes are scaled in accordance with the average chromosome fitness of 1.0 (Cartwright, 2008b). Each chromosome is copied into the parent pool and the number of copies is based on the integer part of the average fitness. The fitness of the chromosome is then subtracted from the average fitness and yields a residual fitness value which must be below 1.0. A modified roulette wheel or tournament selection is then performed using these residual values to fill the remaining space in the parent pool. The deterministic step in the stochastic remainder selection ensures that every

chromosome with a fitness above 1.0 will appear at least once in the parent pool.

Goldberg and Deb (1990) criticised that by using suitable adjustment of selection parameters, except the roulette wheel selection, a similar performance can be achieved with most selection methods, thus, there is no absolute better selection method in the GA. They made such observations based on four different selection schemes: roulette wheel, tournament selection, fitness ranking and steady state selection.

In our design, to avoid the poorest chromosome being selected into the next generation and to prevent the premature convergence in our model, we chose the tournament selection. Additionally, we also applied the elitism strategy to retain fitter chromosomes in the parent pool.

3.2.1.4 Evolving the potential solution

The selection method in GAs can only introduce the existing chromosome to the parent pool or remove the chromosomes from the parent pool, however, it is unable to create new chromosomes for the next generation. Therefore, a mechanism is required to modify the existing chromosome by introducing "new life" into the population. Two GA operators that are able to create new chromosomes are crossover and mutation.

A *crossover operator* is the main operator for generating new chromosomes from existing chromosomes. There are three common ways of performing crossover: single-point, two-point and uniform crossovers, as showed in Figure 3.6. For *single-point crossover*, two chromosomes from the parent pool are each cut at the same corresponding point and exchange the section after the cut point to produce new chromosomes as offspring of the parents. Meanwhile, the *two-point crossover* is very much like single-point crossover, except that two cut points instead of one are performed. For two-point crossover, the offspring are produced by swapping the intervening cut points between two parents. *Uniform crossover* radically differs from single- and two-point crossovers. In uniform crossover, many small sections rather than a single large block are swapped between parent chromosomes. The exact swapping points are determined by a *crossover mask* which contains a list of random binary values generated every time for each pair of parents.

Uniform crossover is simple, but could be severely disruptive and the degree of disruption critically relies on the crossover mask (i.e. a defined string used to determine the cut points on the parent chromosomes) rather than the defining length of the chromosome as single- and two-point crossovers does. Therefore the crossover mark, in general, is on average one half of the chromosome length which avoids unnecessary disruption (Cartwright, 2008b). On the other hand, uniform crossover is also likely to break up any large building blocks in the parent chromosomes and, in some cases, it could be too destructive for a highly correlated group of individual units in the chromosome. Therefore, it is more effective when applied to problems which involve only a limited correlation between genes, probably, one or two places apart in the

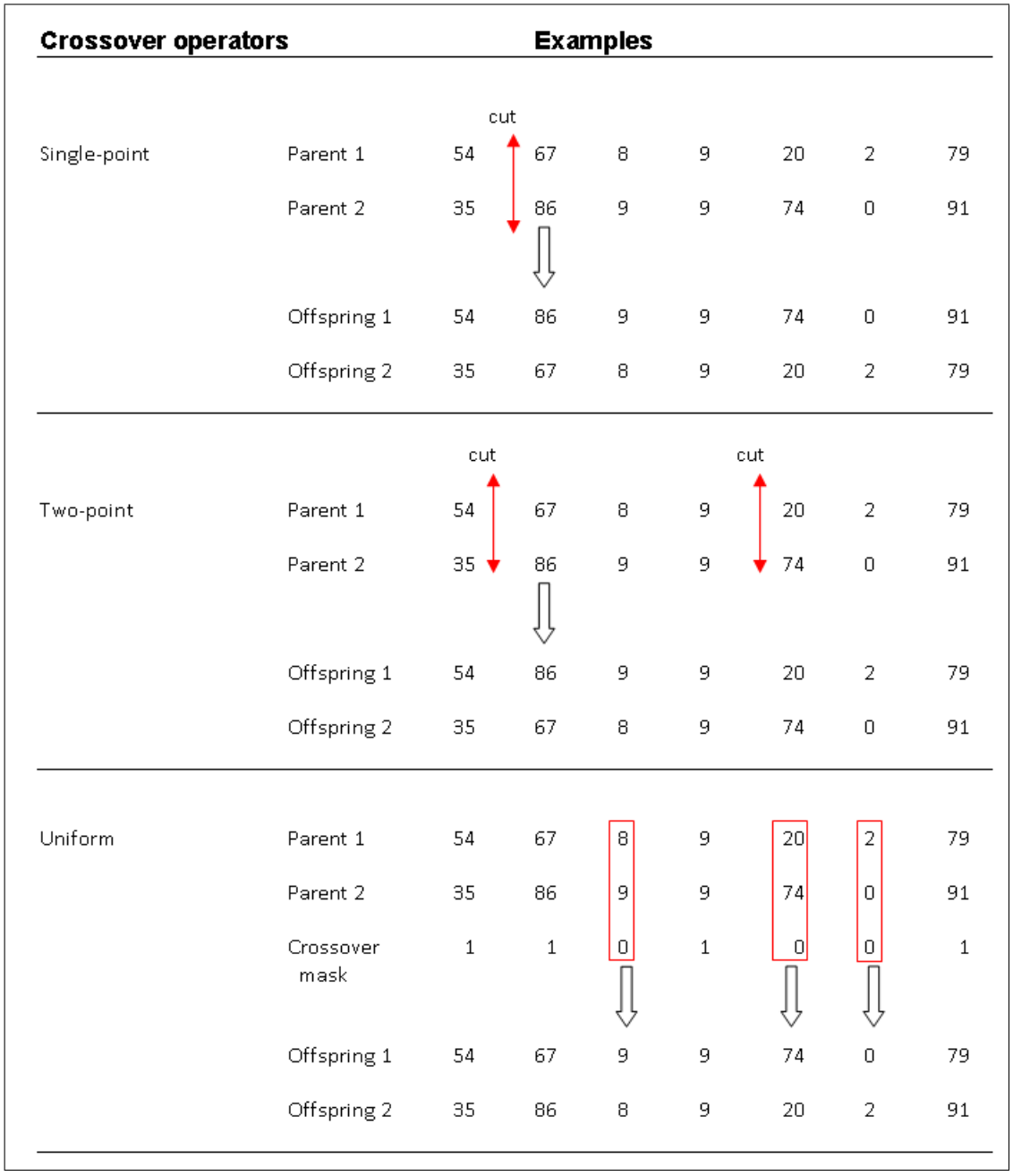


Figure 3.6: Three common crossover operators for GAs. These operators form the offspring of chromosomes represented by real number encoding. The crossover operators create two descendants (offspring) from two parents, using the predefined cut points or the crossover mask to determine which parent contributes which bits.

chromosome. Goldberg (1989) observed that multiple cut points, except the two-point crossover, could lead to the over-interaction between similar chromosomes at the premature stage of convergence, although Syswerda (1989) argued in favour of uniform crossover. Syswerda argued that uniform crossover is beneficial when the ordering of genes in the chromosomes is entirely irrelevant and less disruptive with longer defining length chromosomes. DeJong and Spears (1991) pointed out that the increased disruption of uniform crossover could be an advantage if the population size is small.

In crossover operation, the probability that a chromosome selected from a parent pool is determined by a crossover probability, p_c. The p_c rate is normally large, typically in the range of $0.6 - 1.0$ (Beasley et al., 1993; Busetti, 2001) so that most, and possibly all, chromosomes in the parent pool are selected for mating.

In the process of chromosome evolution, the crossover operator is responsible for introducing new chromosomes to the next generation only by shuffling genes between parent chromosomes. As a result, the population is soon flooded by homogeneous chromosomes and becomes stagnated before the desired solution is found. Therefore a mutation operator which creates new information to the chromosomes is required.

A *mutation operator* acts as a "background" operator in the evolution process and is responsible for providing a small diversity in the population when most chromosomes are identical to prevent *genetic drift* caused by the accumulation of stochastic errors on each generation. Mutation refreshes the population by inserting a new value into a small fraction of chromosomes. However, not all new introduced values are beneficial to the population. This is because mutation does not have a 'blueprint' on what so-called 'good' information and 'bad' information is. It simply changes the values by randomly picking the fraction of the chromosome. As a result, some researchers claimed that mutation does not improve the evolution progress but prolongs the convergence process, because it may destroy important information, by inserting unwanted information into the chromosome. Therefore, it is important to apply mutation cautiously, this is normally at a low rate specified by the mutation rate, p_m, typically 0.001 (Beasley et al., 1993).

The aim of this research is to identify the informative genes that underly the tumourigenesis pathway of a specific cancer, i.e. the genetic pattern of tumour formation and tumour production. Therefore, we used the single-crossover operator in our model to minimise any form of disruption that yielded by the crossover operator.

3.2.1.5 Encoding evolutionary mechanism

In early work, GA chromosomes were encoded using *binary coding* that involves only 1s' and 0s'. With the rapid development of computer technology, most problems, nowadays, are full of complications and contradictions which require better ways to encode the problems. Binary coding became obsolete, as it had a small binary length, which is insufficient to express the present problems. In addition, binary coding has added complexity in manipulating chromosomes in the evolution process, although it can be easily solved by applying an additional programming execution rule. Nevertheless further problems arise when the degree of complexity of the problems are increased, binary coding does not have an adequate length of coding to express problems and it is far more difficult to interpret than other alternative forms of coding such as *real number representation*, which can accommodate far more complicated problems.

Except for binary coding and real number coding, an alternative encoding scheme is *gray code* introduced by Whitley (2001). Gray code is a binary form of encoding scheme and an example of gray coding is presented in Figure 3.7. The advantage of gray code is that successive real numbers only differ with one binary digit, so a small change in the Gray code may translate into a small change in the real value of a parameter. However, gray code has yet to receive attention in science.

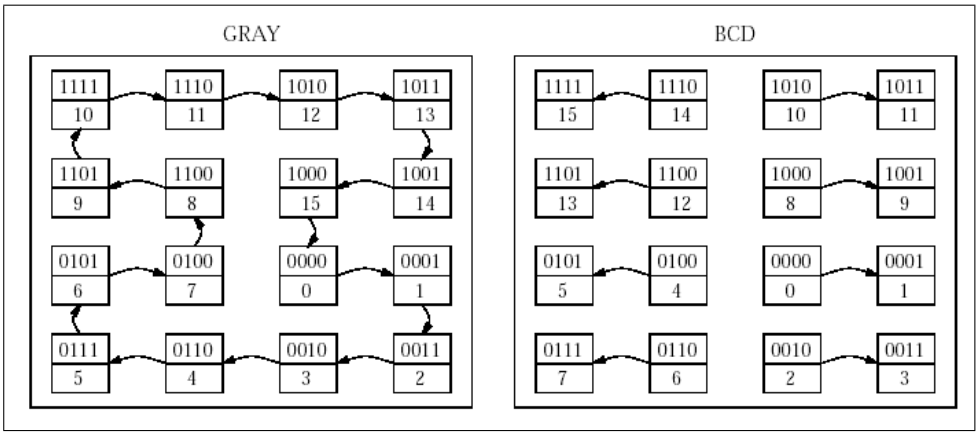


Figure 3.7: Gray code versus Binary coding. This figure is extracted from Whitley (2001). The arrow shows the possible adjacent values can be connected with the change on a single bit in the scheme. The Binary representation has less flexibility than the Gray representation.

3.2.1.6 Elitism

A GA provides some level of certainty of the identified chromosomes, however, during the evolutionary process, the quality of the solution may be distorted. Roulette wheel selection and tournament selection can fail to select the best chromosome from a population as parent, even if no other chromosome is of comparable fitness. Stochastic remainder selection guarantees the best chromosome, but crossover operator and mutation operator might jeopardise the quality of the offspring. In order to avoid persistent problems, an elitist strategy is required.

Elitism is an independent selection scheme that preprocesses the chromosomes in the population before the conventional selection method in a GA is applied. In elitist strategy, the best chromosome in the current generation is copied into the parent pool without having to compete against other chromosomes for selection. This process ensures that the best chromosome, in any generation, will remain in the next generation until it is eventually replaced by a chromosome of superior fitness. Elitism promotes diversity and encourages a wider exploitation of the search space, at a stage when most chromosomes in the population have almost converged. At the early stage of GA evolution, the elitism accelerates the convergence process of the chromosomes by retaining chromosomes with similar characteristics in the population, and at the later stage when most chromosomes in the population are homogenised, the elitism slows down the convergence

process by introducing chromosomes with different characteristics into the population. Therefore, it reduces the disruption which may be caused by mutation during the evolutionary process.

3.2.1.7 Exploration versus Exploitation

For an efficient search in GAs, two contradictory search techniques are generally required to find a global maxima (optimal peak): *exploration (performed by crossover and mutation operators)* to investigate new and unknown areas in the search space and *exploitation (performed by selection operator)* to make use of information (knowledge) found in the previous points will help in finding better points within a region space (Beasley et al., 1993). A balance between both techniques are vital in identifying optimal results, thus, a trade-off between these two techniques must be wisely judged when applied to the problem.

Holland (1992) showed that a well-balanced ratio between exploration and exploitation can be found in a GA using a k-armed bandit analogy (evolving from a traditional slot machine with a tradeoff between *exploitation* of a lever that has the highest expected payoff and *exploration* to get more information about the expected payoffs of the other levers). However, Beasley et al. (1993) questioned the practicality of the assumptions made by Holland. A commonly found problem is *genetic drift* which is caused by the accumulation of stochastic errors, which result in a gene becoming predominant in the population, as once a gene has converged in this way, it is fixed. This produces a domino effect, as generations go by, each gene eventually becomes fixed (Beasley et al., 1993). The impaction of genetic drift to a GA could be beneficial if the predominant gene is what is being look for, however, it can become disastrous if the population is dominated by the wrong gene.

Beasley et al. (1993) suggested the rate of genetic drift can be reduced by increasing the *mutation rate* to explore more unknown peaks in the space. However, excessive mutation could lead to a poor exploration of the region space and high computation processing. This produces skewed results in GAs.

There are no standards or guidelines for balancing the powers of exploration and exploitation in a GA. The efficiency of a GA to find an optimal solution is mainly reliant on the trial-and-error experiments conducted with different GA's parameters and sufficient number of evolution (i.e. fitness evaluation) provided to the chromosomes. Therefore, in our design, we examined various sizes of fitness evaluations to determine the best balance ratio between exploration and exploitation of a GA.

3.2.2 ANN - A universal computational method

An ANN is a computational model that attempts to simulate the structure and/or functional aspects of the human brain in the way that an individual biological neuron reacts in pattern recognition. In the

human brain, it is the combined efforts of numerous neurons acting together to create complex behaviours, while in the ANN, it mirrors the structure of the human brain in which many simple processing elements, known as *nodes*, acting as neurons, work co-operatively. As a consequence, ANNs are often referred to as a *connectionist model*, in the context of machine learning.

An ANN contains numerous nodes that are interconnected using connection links which are associated with different *connection weights* that are arranged in layers (see Figure 3.8). The initial connection weights for constructing ANN are given random values as the network does not have prior knowledge for modelling the problem. The learning process then takes place, during which the network is shown with a large number of examples that are required to learn, and in response adjusts its connection weights in order to give meaningful results. A network in which all the weights have been determined is considered as fully trained and is ready for use. The training process of ANN is shown in Figure 3.9.

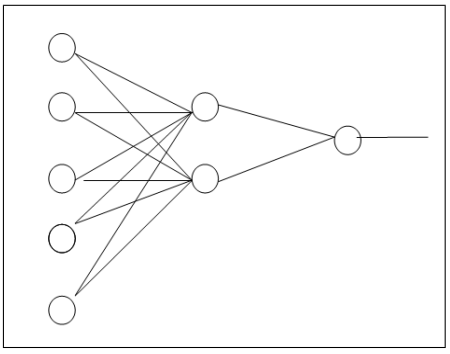


Figure 3.8: A typical 3-layered ANN.

Each ANN node is featured by a set of connection weights, an activity level that determines the polarisation state of a node, an output activation value and a bias value, each represented mathematically by real numbers. The connection weight can be positive (i.e. excitatory) or negative (i.e. inhibitory), and determines the effect of the incoming signal on the activity level of the node. The input signals, generally, are sum linearly as showed in Equation 3.1, yielding an output value for the node. If the output value exceeds the activation level, the node raises the activation (i.e. positive) sign. If the output value is below activation level, the node lowers the activation (i.e. negative) sign. The activation of a node is determined by the activation function of the network.

$$\textsc{Net}_j = \sum_{i=0}^{n} w_{ij} x_{ij} + b_j, \tag{3.1}$$

where w_{ij} is the connection weight between nodes i and j, x_{ij} is the input from node i to node j and b_j is the bias value for node j.

An ANN can execute complex types of tasks, for instance, forming a model from data which does not require a comprehensive theoretical or prior model exist. Although ANN is difficult to interpret rather than other

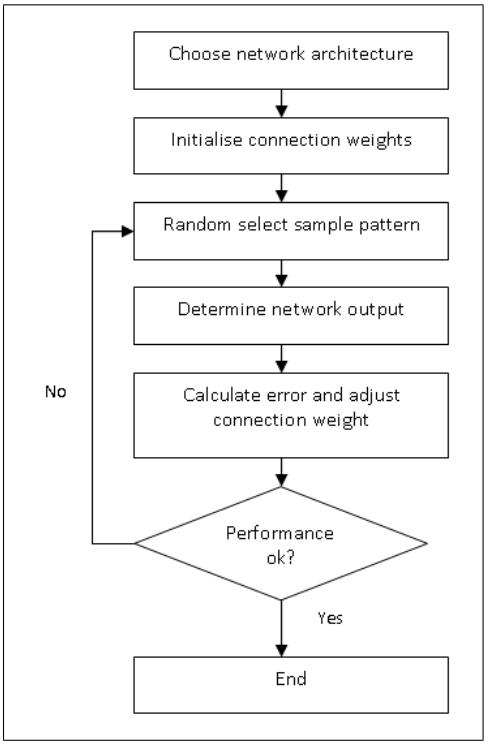


Figure 3.9: The training process of an ANN.

computational models, however, its superior performance in pattern recognition is still impeccable.

A typical ANN model is characterised by:

1. The connection pattern between nodes *(architecture)*.
2. The method of determining weights on the connections *(training/learning)*.
3. The *activation function* that define how the output from a node is determined from its input.

3.2.2.1 The architecture of the network

Neurons, more commonly known as *nodes*, are fundamental elements used to construct an ANN model. Each node receives and integrates input signals, performs simple computations on the sum of the signals, using an activation function and outputs the result to its adjacent node. The ANN is built with numerous nodes interconnected in the form of layers (see Figure 3.8 on page 76). Nodes are joined by *connection links* which allow messages to pass from one node to another. Some nodes take input directly from the external environment; others may have access only to data generated internally. Consequently, some connection links act as inputs providing a pipeline through which data arrives. Nevertheless, every ANN model contains at least one output connection link that provides feedback to the user.

Generally, smaller ANN structures are used rather than larger ones. This is due to the fact that the over-

fitting problem is normally occurred in larger ANN structures and to be specific, the over-fitting problem is generally expected when larger hidden nodes are used in the network (Schwarzer et al., 2000). The number of hidden nodes is generally one half of the number of nodes in the input layer. Therefore, to reduce the risk of the over-fitting problem, the ANN structure is kept small, ideally in the range of 6 - 50 nodes. However, in most microarray studies, an ANN model contains up to hundreds or thousands of nodes (Bloom et al., 2004; Narayanan et al., 2005; Ko et al., 2005), which could lead to a severe over-fitting problem and the results were flawed.

To prevent the over-fitting problem in our model, we used smaller ANN structure which comprised of 10 input nodes, 5 hidden nodes and 2 - 4 output nodes in which each output node corresponded to a cancer class in the data set.

3.2.2.2 The training of the network

In the iterative ANN training process, as shown in Figure 3.9 on page 77, the network is shown a sample pattern and uses the pattern to calculate its output. The output is then compared with the target output, i.e. an ideal output for the sample. The difference between the target output and the network output is the measure of how well the network is performing. Unless the output is perfectly matched with the target output, an adjustment is usually made to the weights to improve network performance.

The adjustment of the connection weights is measured by the error δ, i.e. the discrepancy between target output and network output, which, if the network contains only one output node y and the target output t, thus δ can be expressed as:

$$\delta = t - y. \tag{3.2}$$

If both the network output and the target output are identical, no further learning is required for this sample pattern. Another sample pattern is fed into the network and the training process is continued. If the match is not perfect, the network needs to improve by adjusting the connection weights so that the network can perform better when the same sample pattern is provided in the future. The adjustment of the weight Δw, is the proportion of both the input to the node x, and the size of the error δ:

$$\Delta w = \eta \delta x, \tag{3.3}$$

where η is the *learning rate* which determines the size of the changes, i.e. high or low, to the weight. The connection weight is then updated with:

$$w_{\text{NEW}} = w_{\text{OLD}} + \Delta w. \tag{3.4}$$

Once the weights on all connections have been adjusted, another sample pattern is taken and the process is continued until all sample patterns have been learned and the error δ in the network's prediction becomes negligible. This training process is commonly known as *backpropagation learning*.

The training process is aimed at diminishing the difference between target output and network output over a large number of sample patterns. The error could be reduced by making a suitable change to the connection weights, as described earlier, or by the incorporation of learning rate and/or momentum.

In an ANN model, the weights are used to store information about sample patterns and this information builds up over time as the training proceeds. If a large adjustment is made to the weights, knowledge learned previously will be jeopardised. However, if a small adjustment is made to the weights, they are only moved a little into the direction of the optimum values and the learning will take too long to converge. Thus, a typical learning rate η of the ANN is below 0.1 (Cartwright, 2008a).

Cartwright (2008a) suggested a sensible compromise solution based on gradually diminishing the value of the learning rate as training proceeds, so that in the early stages a coarse pruning on the weights can be performed, while in the later stages, only gentle adjustments are made to the weights which allow them to be fine-tuned. For this solution, Cartwright suggested the value should be in the range of $0.0 - 1.0$.

ANNs with a large number of interconnected nodes are able to model any continuous function. Consequently, the error will also be highly corrugated, displaying numerous minima and maxima. Once the adjustments to the weights lessen, the network can be easily trapped, making the learning cyclic. This phenomena is known as *local minima* in the context of machine learning. To reduce the chance of trapping in endless oscillation, a *momentum* α, is generally applied to update connection weights. Momentum is the velocity of how fast the network is being trained, and will provide spontaneous speed to the network to pass through the local peaks in the error surface, by adding a proportion of the update of the weight in the previous epoch, $n-1$, to the weight update in the current epoch n:

$$w_{ij}(n+1) = w_{ij}(n) + \eta\delta_j(n)x_{ij} + \alpha[w_{ij}(n) - w_{ij}(n-1)], \tag{3.5}$$

where $0 \leq \alpha < 1.0$.

The effect is to let momentum update the weights as the network travels across the error surface. Consequently, the network is more likely to escape from a local minima on the error surface rather than being trapped within it.

In addition to being trapped in the local minima, ANNs can be easily over-fitted by the data. This arises when the network takes too long to learn or when the network is over-parameterised. As the network learns, connection weights are adjusted so that the network can model general rules that underly the data. If

these general rules are applied to a large proportion of the sample patterns, the network will repeatedly see examples of these rules and learn from them first. Subsequently, when more specialised rules, which occur in a few examples, appear in the network, the network will start to learn these rules. Once it has learned these rules well, if the training is allowed to continue, the network may start to learn specific sample patterns within the data and the network will then be overtrained (over-fitted). This is because the network tried to fit the connection weight closer to the target output so that it can reduce the error rate of the network.

Over-fitting problems in ANNs, generally, can be tackled either by, monitoring the quality of the training process using appropriate validation mechanisms (see Section 2.2.2 on page 33), or by ensuring small and sufficient networks are used to assess the data. The use of a validation mechanism in assessing network performance is commonly used by most studies for pattern recognition problems, sample classification to be exact.

Taking into considerations of over-fitting and trapping problems, we decided to use a simple *feedforward learning* rather than the backpropagation learning and with no additional learning acceleration techniques in our model. The primary objective of this research was to find a feature set that correctly acts to discriminate between the classes. The presumption is, and this is a major assumption of our model, that the feature set will actually be the feature set that in some sense correctly acts to discriminate between the classes. That is to say, that by deliberately not focusing on the quality of the ANN classifier, then the selected feature set will be closer to the true discriminating feature set for the given classes. This has led us to select feedforward learning method in our model.

3.2.2.3 The activation function of the network

In order for a node to calculate its output signal (activation signal), it requires an *activation function* $f(x)$, that determines the state of action potential firing (or idle) of the node. An activation function is the mathematical computation that limits the amplitude of the output signal of a node with a predefined set of mathematical theorems. Figure 3.10 presents four common activation functions implemented in ANNs.

Most activation functions, except the linear function, reflect the firing rate in a normalisable range. Some functions allow the firing occurring on the values falling in the range of two extreme numbers, i.e. positive and negative signs, $f(x)\epsilon[1,-1]$. This is known as a *bipolar range*. Some functions, on the other hand, are based on only positive signs, i.e. *binary range*, $f(x)\epsilon[0,1]$, to determine the firing of a node. There is also a function which allows the firing based on a positive constant number, i.e. a *threshold* value θ. If the output signal is greater than θ, the node will fire the signal, otherwise, the node will switch into idle mode.

As the ANN field has developed, numerous functions have been proposed. The commonly used functions

include threshold function (Beiko and Charlebois, 2005; Cho et al., 2003a), linear function (Keedwell and Narayanan, 2003; Tong, 2009), sigmoid function (Toure and Basu, 2001; Tong, 2009) and tanh function (Valdés and Barton, 2004; Bevilacqua et al., 2006a; Lin et al., 2006; Tong, 2009; Tong et al., 2009). Other less likely used function includes Gaussian function (Shenouda, 2006).

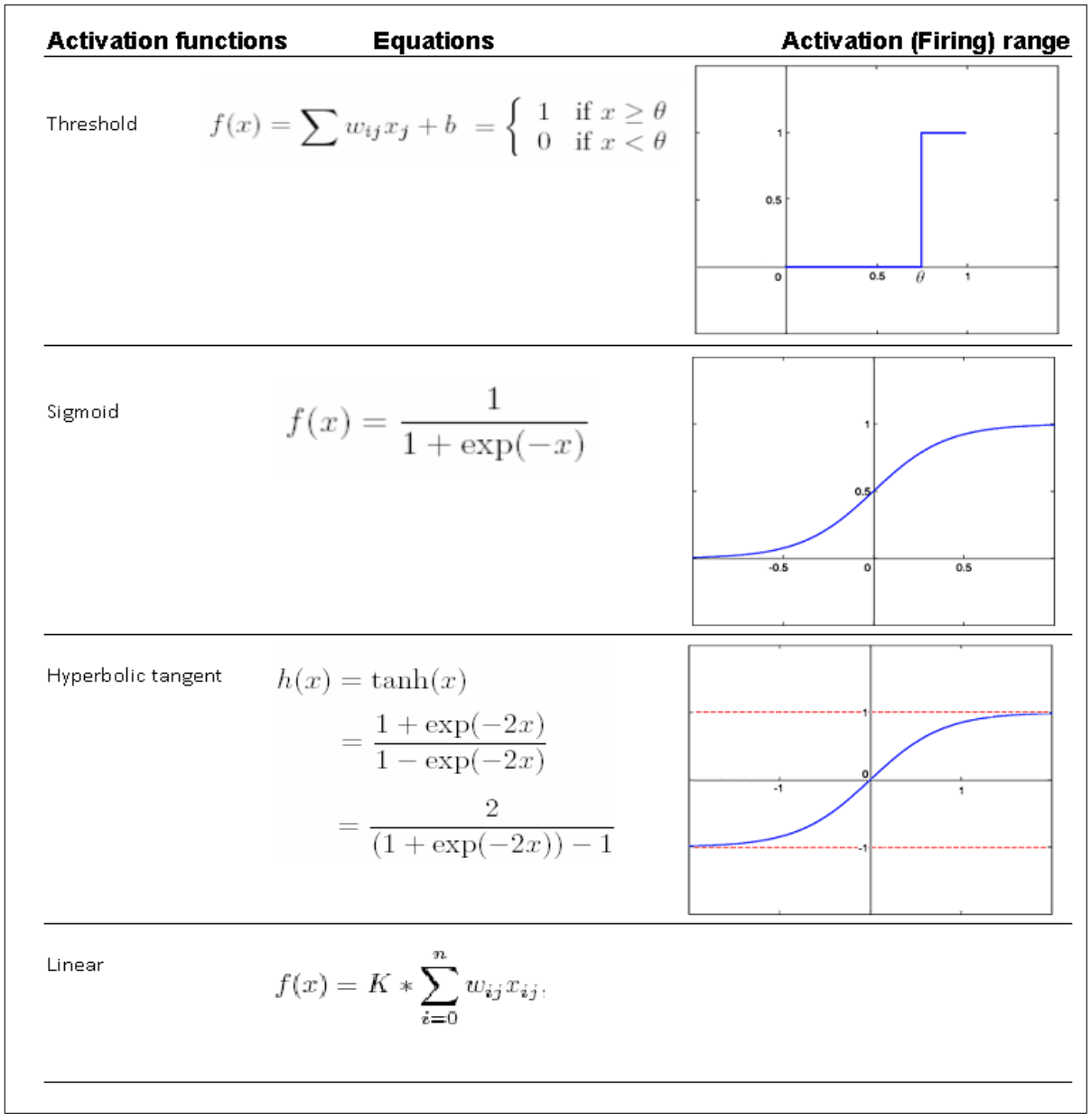


Figure 3.10: Four common activation functions for ANNs. These functions determine the state of action potential firing of a node. When the activation value of a node is favour towards the positive sign (excitatory), the node firing its signal, otherwise the node is static (inhibitory).

For *threshold activation function*, the firing rate of a node is determined by a threshold value θ, usually a positive sign. If the total input of the node is below θ, the node's output is changed to 0 and the node is static. If the summed input is greater than θ, the output value is set to 1 and the node status becomes active which allows it to fire.

Unlike threshold function, *sigmoid* and *tanh activation functions* allow flexible firing zones, provided that an output value is covered within its firing range. The sigmoid function, normally, provides an output in binary range while the tanh function has an output covering the bipolar range. Both the sigmoid and the tanh functions have a different speed advantage during training.

For *linear activation function*, the node passes the summed input signals directly to the output, normally after multiplication by a scaling factor K, usually in the range of $0 < K \leq 1.0$. The simplest linear function is *identify function* in which K is 1.0, thus the output from a node is identical to its input. The output from a linear function may be capped to prevent signals from becoming unreasonably large, in the case of a large network with connection weights greater than 1, so that input and output are linearly related within a restricted range.

In our design, we examined the implication of these four activation functions in computing the fitness values of the chromosomes.

3.2.3 Hybridising GAs and ANNs

Both GAs and ANNs are adaptive, robust and able to deal successfully with a wide range of problems including highly nonlinear models and noisy data. In addition, they do not need priori information to model the problem being studied. Therefore, from a practical perspective, GAs and ANNs appear to work best in combination (Busetti, 2001).

A typical combination is to use ANN as the prime modelling tool with GAs to optimise the network parameters (see Figure 1.3a on page 11). As both of these fields have developed, recent research focuses on using ANNs to model the GA fitness function. Although different implementations of ANN parameters in assessing the quality of a fitness function have been introduced, they can generally be tackled by two computations, i.e. based on the number of correctly labelled sample patterns returned by an ANN (Cho et al., 2003a; Tong, 2009; Tong et al., 2009), or the network error rate computed by an ANN (Lin et al., 2006; Bevilacqua et al., 2006a).

In this thesis, we design a feature extraction model, namely Genetic Algorithm-Neural Network (GANN). In our model, the ANN is used as a fitness score generator to compute the fitness function of a GA in identifying informative features for microarray data, rather than being used as a predictor to model data classes. The philosophy of our approach is to apply the parameter settings that are no more complex than that required for the solution to the problem, i.e. the Ockham's Razor principle.

3.2.3.1 The general description of GANN model

Figure 3.11 on page 84 shows the architectural design of the GANN model. The GANN model consists of three main modules, i.e. initialising population of chromosomes, calculating fitness values for each chromosome in the population and evaluating the quality of each chromosome in the population.

Given an experimental data set to the GANN model, a population containing chromosomes (parent pool), is first initialised and at this stage, the quality of chromosomes in the population are yet to be evaluated as they are not assigned fitness values. Once the population is fully occupied by chromosomes, ANNs are used to calculate the fitness values of each chromosome in the population, as depicted in Figure 3.11b. In the fitness computation phase, a new ANN is constructed each time for each chromosome. The evaluation process soon begins when all chromosomes in the population are assigned fitness values. In the evaluation process, two fitted chromosomes are selected for reproduction and new offspring are produced. The fitness of this new offspring is then calculated by an ANN. To ensure that the fitter chromosomes will not be discarded in future generations, an elitism scheme is adopted, in which only one least fit chromosome in the current population, is replaced by new offspring in each generation. Once the chromosome is substituted by new offspring, the entire population is copied in the new population (new generation) and the new evaluation cycle for the generation begins. The cycle will continue until the termination criteria is met and the GANN stops. Finally, a summary result: the number of correctly labelled samples in each model repetition run and the ranked features based on the entire number of repetitions in accordance to their selection frequency, are produced as the final outcome from the GANN model.

In order to visualise the relationship between features, a genomic analysis tool, i.e. HeatMapViewer, from the GenePattern Software Suites is used to graphically present the feature selection results.

It is worth pointing that unlike most hybrid GA/ANN models which focusing on the quality of the ANN architecture to discriminate data classes, the role of an ANN in our hybrid model is merely a fitness score generator for an GA. We deliberately use simple ANN models with no acceleration parameters to avoid any form of variability that might be incurred by a sophisticated ANN architecture. For the GANN model, the derived ANN is an artefact of the process and is discarded. In the subsequent sections, we will look at each module of the GANN model and its stopping criteria. The summary of GANN parameters is depicted in Table 3.4 on page 90.

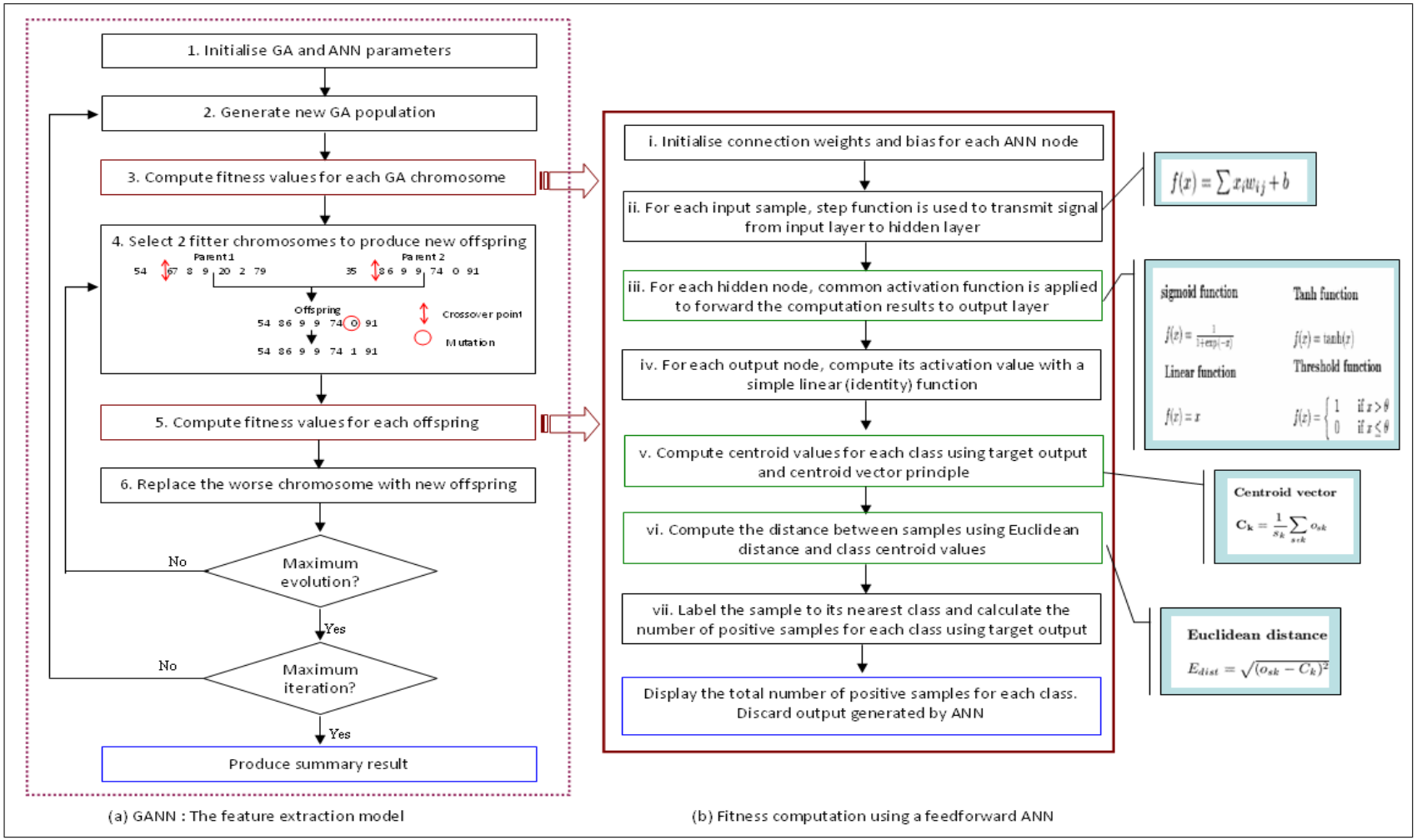


Figure 3.11: GANN: The flowchart design. The diagram (a) shows the overall process of the model. The diagram (b) presents the ANN steps in computing fitness values for GA chromosomes. The parameter settings for GAs and ANNs are first intialised before GANN begins. A population of chromosomes is generated and the fitness for each chromosome is calculated using feedforward ANNs. In the ANN process, the connection weights are initialised randomly based on the range [0,1]. The step function is applied on the input layer to hidden layer and 4 activation functions are used (independently) on the hidden layer to output layer. Based on the result returned by the output layer, the class centroid for each data class is computed using the centroid vector principle and the target output. The distance between each sample to each class centroid is measured using the Euclidean distance technique and the sample is labelled to its nearest class. The correctness of the sample classes are judged by the target output of the samples and the result is used as the fitness value of the chromosome. Once all chromosomes are assigned with fitness scores, 2 fitter chromosomes are selected to produce offspring. The properties of these chromosomes (parent) are cross-overed to produce new offspring, which is then mutated to create a diversity from its parent. In this process, the GA optimise the network weights and evolves the properties of chromosomes. The fitness of the offspring is then calculated by re-run the ANN process using the evolved ANN weights. The least fit chromosome in the population is replaced by the offspring. Through the evolution in many generations, the least fit chromosomes are gradually removed from the population and the optimal solutions are obtained.

3.2.3.2 Population initialisation

A GA normally operates on the basis of a finite-sized population of fixed-length chromosomes. This is to avoid infinite oscillation of a GA in searching the optimal solution, albeit, by limiting the size of population, the optimality of the solution is in question. Even so, a GA is still able to derive sub-optimal solution with limited available information.

In this module, a group of features from the data set are randomly selected by GA and stored in a temporary repository (i.e. population) for further use. The number of features stored is reliant on the size of the population. All features in the population are encoded with real-number representations and are considered as a subset of a member (i.e. GA chromosome) in the population. We set the chromosome size to 10. In other words, there are 10 features in each chromosome. Depending on the size of the population N, the number of features in the population can be expressed as N x 10. Features are allowed to be duplicated so there is potential that a feature can be selected more than one time by the GA.

Cartwright (2008b) recommended small population sizes (40 - 100 chromosomes) as this will lead to a fast convergence of the solution and is time effective. However, DeJong and Spears (1991) discovered that fast convergence does not guarantee the quality of the solution as with smaller population sizes.

In our design, three population sizes: 100, 200 and 300, are used to investigate the implication of different population sizes in finding the correct solution to the problem. We do not consider a population size below 100, as our experimental data features a high dimension (2308 - 10000 features) and a small population size does not have sufficient capacity to accommodate this information. However, larger population sizes could slow down the processing time of GANN, thus, we limited the population size to a maximum of 300 in our model.

3.2.3.3 Fitness computation

Fitness computation is the core component in our design and poor judgement of the fitness function can lead to the deficiency of our model. A penalty function is generally used to assist the function, in assessing the quality of a chromosome, however, this incorporation has added complexity to GAs and is computationally intensive. Over the last few years, research on the incorporation of machine learning methods, such as ANNs and k-nearest neighbours (KNNs) have been introduced to replace penalty functions. These studies reported better performance and delivered promising results. However, this new way of incorporation is not fully-fledged and most studies attempted to use complex parameters in the combination.

In this module, a new ANN model is constructed each time a new chromosome is introduced to the network. Each feature in a chromosome represents an input node to the ANN. In an iterative training process, all

samples in the data set are processed by the network and the activation output for each sample is calculated. Using the target output of the data, the network calculates centroid values of the classes (see Equation 3.8) and the distance between samples for each class is computed using the Euclidean distance technique (see Equation 3.7). The difference between the target output and the network output (see Equation 3.6) is the fitness of the chromosome, i.e. the number of correctly labelled samples per chromosome. This training process is known as *feedforward learning*.

In our design, we used a simple 3-layered feedforward ANN architecture to calculate the fitness values of each chromosome in the population (see Figure 3.11b on page 84). The fitness function of our model is defined as the number of correctly labelled samples for each chromosome. The equation is presented as follow:

$$\text{fitness } f = \sum_{i=1}^{n} \sum_{k=1}^{c} s_{ik}, \tag{3.6}$$

$$s_{ik} = t_{ik} - \sqrt{(A_{ik} - C_k)^2} \quad \begin{cases} \geq f(x), & O_{ik} = T_{ik} \\ < f(x), & O_{ik} \neq T_{ik} \ , \end{cases} \tag{3.7}$$

$$\text{centroid } C_k = \frac{1}{s_k} \sum_{s \epsilon k} A_{sk}, \tag{3.8}$$

where s_{ik} is the sample i in class k, T_{ik} is the target output of sample i, $f(x)$ is the activation function to be used in the ANN, A_{ik} represents the output activation value for sample i, C_k is the centroid value of class k and O_{ik} is the final output value generated by ANNs.

The centroid vector principle and the Euclidean distance ($\sqrt{(A_{ik} - C_k)^2}$) are the most fundamental statistics to construct any computer algorithm. The centroid vector principle is laid on the use of mean (i.e. centroid) and standard deviations of classes to label samples. Since our research is on feature extraction instead of sample classification, we exclude the use of standard deviations of classes in our design (see Equation 3.8). Meanwhile, the Euclidean distance is used to measure how far the distance of individual samples are from each class. Depending on the proximity value, the sample is labelled to its nearest class.

As Schwarzer et al. (2000) said: *"With increasing number of hidden units we fit more and more implausible functions which move away from the true law f, and hence the misclassification probability increases."*. Large network sizes can lead to over-fitting problems as the network tries to fit the connection weight closer to the target output so that it can reduce the error rate of the network. Thus, to reduce the risk of over-fitting in our design, a standard 3-layered network architecture 10-5-O is applied, i.e. 10 input nodes corresponding to the number of features in a chromosome, 5 hidden nodes and O output nodes corresponding to the number of classes in the experimental data.

To make the ANN function, an activation function is required. In our design, despite drawing the finding

based on one type of activation function, we examine the potentialities of four activation functions in calculating the fitness values of each chromosome. These activation functions are *sigmoid, linear, hyperbolic tangent (tanh)* and *threshold* in which their equations and activation range is depicted in Figure 3.10 on page 81. To keep the ANN model simple, we only adopt the feedforward learning algorithm and the bias parameter.

3.2.3.4 Chromosome evolution

In the process of chromosome evolution, the selection mechanism is responsible for selecting two fitter chromosomes for reproduction, i.e. mating. The crossover operator is responsible for introducing new chromosomes (offspring) to the next generation and the mutation operator is responsible for creating new information in the offspring.

The *tournament selection* with the tournament size of 2 is chosen in our design because it often yields a more diverse population (Mitchell, 1997a) which could lead to deeper exploitation of the chromosome search and to prevent premature convergence of homogenous chromosomes.

Cartwright (2008b) commented that the evolution parameters affect the success of a GA, crossover operators to be specific (DeJong and Spears, 1991). Uniform crossover has a high reputation in pattern recognition problems, due to its flexibility on the points to be exchanged in chromosomes and simplicity. However, it is critically reliant on the crossover mask and could be too destructive for high correlated groups of an individual unit in the chromosome. Therefore, instead of using a uniform crossover operator, we use a *single-point crossover* operator (see Figure 3.6 on page 72) in our design as it is least destructive to the relationship of the units in chromosomes and the crossover rate p_c is set to 0.5.

A mutation operator is crucial in the evolution process as it prevents genetic drift caused by the accumulation of stochastic errors on each generation. Mutation refreshes the population by inserting a new value into a small fraction of the chromosome. However, not all of these values are beneficial to the population. Thus, in our design, we use a small mutation rate $p_m = 0.1$, to avoid any drastic changes in the population. Information on the selection methods and GA operators can be found in Sections 3.2.1.3-3.2.1.4.

To ensure that the quality of the chromosomes will not be distorted in the evolution process and to balance the powers of exploitation and exploration, we also apply an *elitism scheme* in our design. In our elitism strategy, only one chromosome is allowed to be replaced by offspring. This is to ensure that a wider exploitation of the search space is provided when the population has almost converged. Additionally, it reduces the disruption which may be caused by mutation operators.

Table 3.2: The summary of the trial based on various sizes of fitness evaluation. These results are based on the use of the tanh activation function with 5000 repetition runs.

Data set	Fitness Evaluation						
	500	1000	5000	10000	15000	20000	25000
Synthetic data set A (2-class data set)							
Fitness accuracy (%)	Nil	Nil	58.36	90.5	93.44	94.62*	95.6*
Predefined features (30)	Nil	Nil	30	30	30	30*	30*
Synthetic data set B (3-class data set)							
Fitness accuracy (%)	Nil	Nil	Nil	4.1	20.24	32.56*	41.28*
Predefined features (30)	Nil	Nil	Nil	29	30	30*	30*

3.2.3.5 Termination criteria

In the iterative process, the algorithm constantly learns new patterns of the data and models general rules that can represent data. However, when the algorithm has been given too much iterations in the modelling process, the algorithm tends to over-fit. Therefore, to prevent the algorithm from over-learning these rules and to stop when the desired solution is achieved, a set of termination criteria is generally required. Cares should be taken in determining these criteria as it affects the generalisation capability of our model in interpreting the problem. Poor decision in the termination criteria may lead to an over- or under-fitting problem and the outcome of the algorithm may be either overly optimistic or overly pessimistic.

In our design, we apply two criteria to stop our model from over-learnt. These criteria examine the effects of our model in two factors, which are as follow:

a. The sufficient amount of freedom for GA to explore and to exploit both the global and the regional feature spaces.

b. The sufficient number of iterations for producing consistent set of results.

The first criteria examines different amount of freedom for our model to deliver optimal results, based on different number of evolution cycles and population sizes, which internally repeating the process of fitness computation for chromosomes in the population. As indicated by Cartwright (2008b), evolution of solutions are crucial in determining the success of a GA rather than the population size. However, there is no clear indication on the number of evolution cycles nor population sizes to be used for microarray studies. A set of trial experiments with an identical set of parameter settings, but different numbers of fitness evaluations, were conducted using synthetic data sets and a summary table is presented in Table 3.2.

The result indicates that the improvement in the identified features are more significant in the increased

Table 3.3: The summary of the trial based on various repetition runs. These results are based on the use of the tanh activation function with 20000 fitness evaluations.

Data set	Repetition Run					
	100	500	1000	5000	10000	15000
Synthetic data set A (2-class data set)						
Fitness accuracy (%)	96*	94.8	95.5	94.62	95.24	95.2
Predefined features (30)	30*	30	30	30	30	30
Synthetic data set B (3-class data set)						
Fitness accuracy (%)	33*	31.2	32.4	32.56	33.64	33.65
Predefined features (30)	26*	28	28	30	30	30

number of fitness evaluations. In fact, the fitness evaluation 20000 can be considered as a minimum setting to differentiate more than 2 classes of samples rather than the fitness evaluation 15000 as it showed a very clear distinction between the degree of fitness confidence on the first 30 extracted features from others features which are also significant but not as important as the first 30. Even though, we decided to examine 10 different evaluation sizes, ranging from 5000 to 50000, to observe the implication of the fitness evolution on delivering consistent results.

The second criteria, on the other hand, assesses the stability of our model in extracting consistent set of features by externally iterating the entire extraction process. Many researchers often ignore the fact that the sufficient number of repetition runs on the computer algorithm can affect the integrity of the reported results. Consequently, most of the published results based on the identical data set vary from one to another. To present the effects of the repetition run on the final result, a set of trial experiments with an identical set of parameter settings, but a different number of repetition runs were conducted, using synthetic data sets as summary shown in the Table 3.3.

Based on the results, there is no significant improvement with incremental repetition runs in response to the fitness accuracy. However, it shows the high accuracy based on the smaller number of repetition runs do not always guarantee that the identified features are the most significant ones. In Table 3.3, the repetition run 100 has a high fitness accuracy in both the binary and the multiclass data sets. It seems reasonable to draw conclusions based on this result, since the result looks promising and a slight improvement with the incremental number of repetition runs does not make a lot of difference in the fitness accuracy, plus, it takes a shorter time to process the results. In fact, this result is unacceptable from the medical and pattern recognition perspectives because it fails to identify all significant features defined in multiclass data set. Amongst the 30 defined features in synthetic data set B, only 26 features were identified. In contrast, with a greater number of repetition runs in the algorithm, all 30 predefined features were identified. Less

repetitions on the algorithm restricts it from exploiting the rules that were learnt (under-fitting), when more repetition runs were performed, the prototype can practise more on the rules and even improvise the rules. When a similar set of rules frequently appear, the algorithm will group these rules as "general" rules, refining them each time when a new sample pattern is introduced. Eventually, the algorithm will formulate a set of logic rules based on these general rules to deal with any unknown samples. Although more repetitions can help the algorithm to formula its own set of rules, too many repetition runs will put the prototype at risk of over-fitting and is computationally intensive. By comparing repetition runs 10000 and 15000 in the synthetic data set A, the performance of the prototype starts decreasing in run 15000, a pre-sign of over-fitting. There is a marginal different on the fitness performance on the repetition runs 5000 and 10000 in both data sets. Therefore, it was decided to iterate the whole process 5000 cycles.

Table 3.4: The summary of the GANN parameters.

Parameter	Setting
GA parameters	
Population size	{100, 200, 300}
Selection	Tournament, tournament size = 2
Crossover operator	Single-point, $P_c = 0.5$
Mutation operator	$P_m = 0.1$
Elitism strategy	Retain $N - 1$ chromosomes in the population, where N is the total number of chromosomes in the population
Fitness function	Number of correctly labelled samples
Fitness evaluation size	{5000, 10000, 15000, 20000, 25000, 30000, 35000, 40000, 45000, 50000}
ANN parameters	
Architecture	10-5-O, where O is ranges from 2-4, for microarray and synthetic data sets 20-10-2 for bioassay data sets
Network Size	67-79 nodes for microarray and synthetic data sets, including 7-9 bias nodes 232 nodes for bioassay data sets, including 12 bias nodes
Learning algorithm	Feedforward
Activation function	{sigmoid, linear, tanh, threshold}
Repetition run	5000

3.3 GenePattern Software Suites - A genomic analysis platform

GenePattern is a powerful scientific genomic analysis platform that provides access to a broad array of computational methods used to analyse genomic data (Reich et al., 2006). It is a freely available software package developed at the Broad Institute of MIT and Harvard. It is designed to enable researchers to develop, capture and reproduce genomic analysis methodologies using the pipelines approach. Figure 3.12 shows the pipeline representation and results that could be derived from GenePattern in a simple form.

In this thesis, the HeatMapViewer module in GenePattern software suites version 3.2 is used to display the relevance of the identified features by our model. Normally, the most relevant features (largest values) are displayed in red (hot) and the least relevant features (smallest values) are displayed in blue (cool). Intermediate values are displayed in different shades of red and blue. This colour-coding scheme provides a quick coherent view of feature correlations. The screen shot of the HeatMapViewer is presented in Figure 4.2 on page 96.

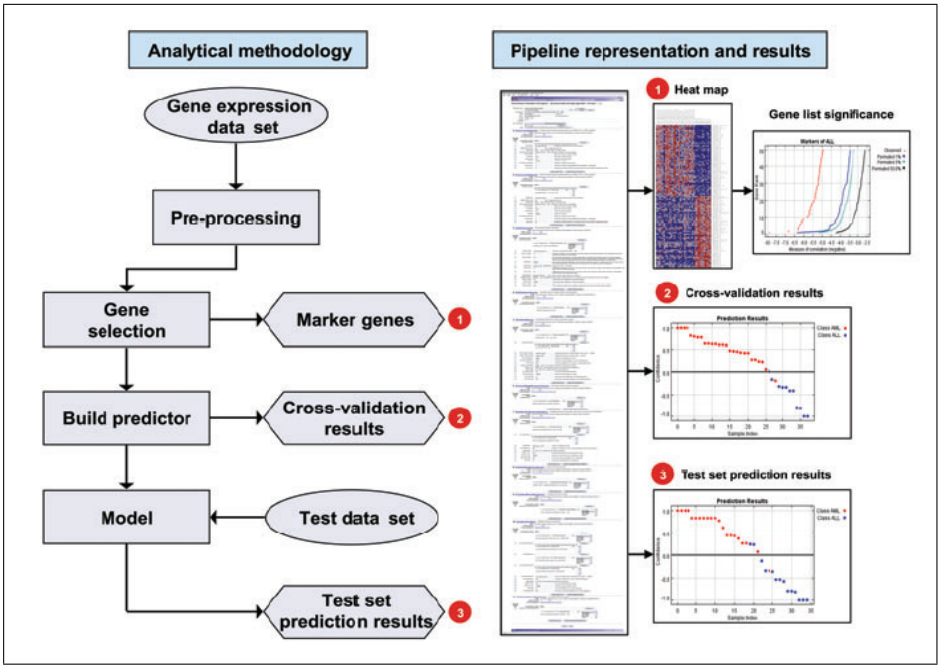


Figure 3.12: The pipeline representation of GenePattern for microarray analysis, which was extracted from Reich et al. (2006). Diagram on the left shows the steps in classifying microarray data and, generally, each step of the analysis invokes manually. In GenePattern (diagram on the right), all steps can be encapsulated in a single, reproducible pipeline that choreographs the entire classification and identification process. The pipeline is then available for modification, with each revision preserved for reproducibility.

3.4 Data Validation - NCBI Genbank & Stanford SOURCE Search System

A cross-reference on the experiment results will be conducted via the National Center for Biotechnology Information (NCBI) Genbank and the SOURCE search and retrieval system by the Stanford University.

The NCBI Genbank sequence database is an annotated collection of all publicly available nucleotide sequences and their protein translocations, i.e. chromosomal structure aberration, which implemented by the NCBI as part of the international collaboration with the European Molecular Biology Laboratory (EMBL) Data Library from the European Bioinformatics Institute (EBI) and the DNA Data Bank of Japan (DDBJ) from the National Institute of Genetics (NIG). The description of the gene entries are retrieved using the Entrez system, a text-based search and retrieval system, supported by the NCBI for all major databases, including

PubMed bibliography, taxonomy project, protein sequences and structures, and many others. In this thesis, the Entrez Gene, one of the several search tools supported by the Entrez system is used to validate our gene findings.

The SOURCE is a unification tool which dynamically collects and compiles data from various scientific databases, such as Gene Ontology (GO), UniGene, SAGE, Genbank and PubMed. It is a text-based search and retrieval system supported by the Stanford University that encapsulates the genetics and molecular biology of genes from variety living organisms, i.e. homo sapiens (human), mus musculus (house mouse), rattus norvegicus (domestic rat), into easy to navigate gene reports.

The screen shot for validating microarray gene expression profile using the NCBI Genbank and the SOURCE system is presented in Figure 4.12 on page 114.

3.5 Summary

In this chapter, we discussed the design of our feature extraction model. We have shown technical perspectives on both GAs and ANNs, how to use ANNs to calculate the fitness values of GA chromosomes. We have applied simple mathematical formulas in fitness computation including the centroid vector principle to calculate mean of each classes, the Euclidean distance to measure the proximity of samples from each classes and network activation functions to determines the potential firing of a node. We have also applied a genomic analysis platform, GenePattern software suite, to demonstrate the gene selection results graphically. The NCBI Genbank and the SOURCE search system have been used to validate the gene findings obtained by our model.

In Chapter 4, the prototype and the experimental study of our model will be presented.

Chapter 4

Prototype and Experimental Study

Chapter 3 presented the conceptual design of our feature extraction model to support the theme of this thesis. This chapter demonstrates the prototype of our model, namely Genetic Algorithm-Neural Network (GANN). The novelty of the GANN model is its simplicity that follows the Ockham's Razor principle which can minimise the potentiality of gene variability errors incurred by data preprocessing which is discussed in Chapter 2. The construction of ANN for computing the fitness values of the chromosomes described in the previous chapter will be explained. Since a standard GA technique, except the fitness computation technique, has been used in the pattern evaluation module, we did not explain the GA construction steps in details, however, we outlined the overview of the implementation for GA evaluation.

The objectives of this chapter are to describe the tools, the prototype and the experimental study for supporting the theme of this thesis. This chapter contains six sections. Section 4.1 presents the software tools used in this thesis, including the programming tool for developing the prototype and the synthetic data sets, the data mining tool for validating the findings of the bioassay data sets and the visualisation tools to present graphically the result findings and the data sets. Section 4.2 explains the needs for the transposition process to preprocess microarray data. The GANN prototype is presented in Section 4.3 and Section 4.4 describes the validation steps conducted using the NCBI Genbank and the SOURCE search system. Section 4.5 describes the research methodology used to test the hypotheses of this thesis and finally, Section 4.6 concludes the chapter.

4.1 Tools used in the Prototype

Five tools: C++, WEKA, GenePattern, Microsoft Excel and R Project, are used to support the theme of this thesis. C++ is an object-oriented programming language that is used for the coding of the GANN

prototype. WEKA is a data mining software that is used to compute the statistical significance of the gene findings and to validate the bioassay findings. GenePattern is a genomic analysis platform that is used to visualise the correlation of the gene findings. Microsoft Office Excel is a spreadsheet software from the Microsoft Office Package Suite that is used to graphically present the findings from the prototype. Lastly, R Project is a language and environment that is used to visualise the data interactions.

4.1.1 Programming language for developing the prototype and the synthetic data

Although C programming language is more commonly used in the machine learning community, C++ is chosen to be the language for developing the prototype since it is less complex in terms of coding commands and rich in variety of built-in functions without a sophisticated programming environment. The screen shots of the important functions in the prototype can be found in Appendix A. In addition to the GANN prototype, two specially written C++ programs are coded to which one is used to transpose the microarray experimental data sets and the other program is used to construct the synthetic data sets. All C++ coding were programmed on the LINUX environment.

For the synthetic data program, the values of genes were designed based on a standard Gaussian distribution, i.e. the mean value of 0 ($\mu = 0$) and the standard deviation of 1 ($\delta = 1$), with the exception on 30 randomly selected genes, from each data set, which were created with different μ values. Table 4.1 presents the settings in the synthetic data sets. In the *synthetic data set 1*, the 30 differentially expressed genes were created with $\mu = 2.0$ and were labelled with the gene indexes 1-15 and 5001-5015. In the *synthetic data set 2*, 10 out of the 30 genes were created with $\mu = 0.5$ (gene indexes 1-10) and the remaining 20 genes were created with $\mu = 2.0$ (gene indexes 11-30).

Table 4.1: The description of the synthetic data sets.

Data set	Description	Significant features
Synthetic data set 1	100 samples equally distributed into 2 class. Each sample has 10000 features which were standardised with $\mu = 0$ and $\delta = 1$.	30 significant features, with the feature indexes 1-15 and 5001-5015, were standardised with $\mu = 2.0$ and $\delta = 1$.
Synthetic data set 2	67 samples distributed into 3 classes, i.e. 20 samples in class 1, 30 in class 2 and 17 in class 3. Each sample has 5000 features which were standardised with $\mu = 0$ and $\delta = 1$.	30 significant features, with the feature indexes 1-10 were standardised with $\mu = 0.5$ and $\delta = 1$ and the feature indexes 11-30 were standardised with $\mu = 2.0$ and $\delta = 1$.

4.1.2 Tool for evaluating the significance of the findings

WEKA (Waikato Environment for Knowledge Analysis) is a data mining software developed by the University of Waikato (Hall et al., 2009). It is a free software that is available to download from its original website (WEKA data mining software). WEKA contains numerous collection of machine learning and data mining tools, such as data preprocessing, classification, regression, clustering, association rules and visualisation. Amongst these tools, the Information Gain (GainRatio) is selected to measure the significant of individual genes extracted from the microarray and synthetic data sets. Four cost-sensitive classifiers, which are naive bayes (NB), support vector machine (SMO), C4.5 tree (J48) and random forest (RF) are used to validate the significance of the attributes extracted from the bioassay data sets. Additionally, the principal component analysis (PCA) is used as a comparative tool in the bioassay data sets.

For GainRatio and PCA, the default parameter settings on the *Attribute Selection* tool were used. To construct the cost-sensitive classifier, the *Cost Sensitive* parameter on the *Meta* option for NB, SMO and RF, and the *MetaCost* parameter for J48 tree were selected. Figure 4.1 shows the screen shot to construct a Cost-Sensitive NB classifier (CSC NB) on the WEKA environment. We used the default WEKA settings for NB and RF classifiers. For SMO classifier, we set the *build logistic models* parameter to *"true"* and for J48 tree, we amended the *Unpruned tree* parameter to *"true"*.

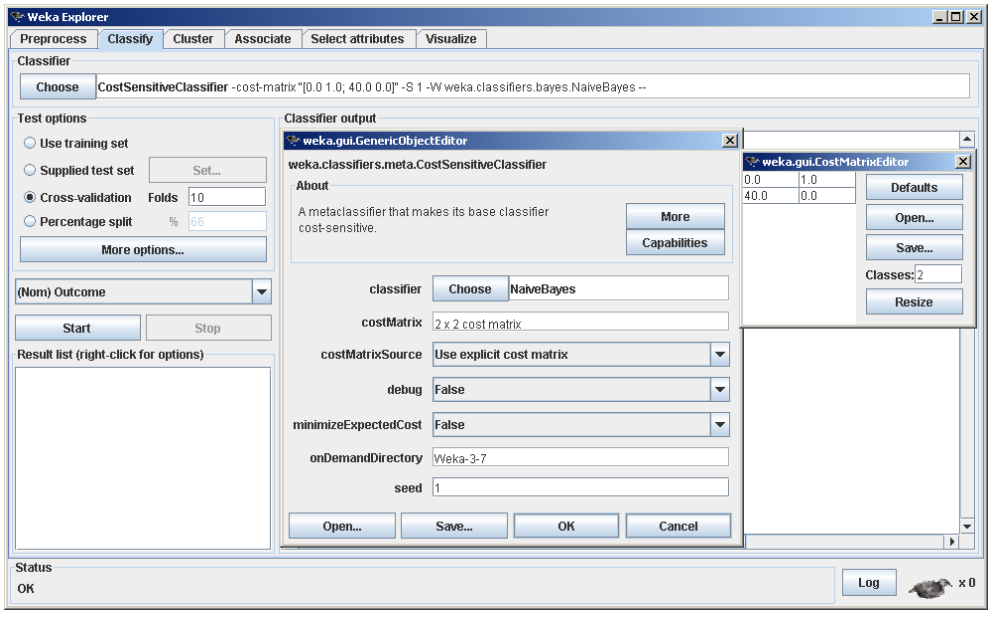


Figure 4.1: The screen shot for constructing a CSC NB on the WEKA environment.

4.1.3 Tool for visualising the significance of the gene findings

GenePattern software suites is a freely available software package developed at the Broad Institute of MIT and Harvard (Reich et al., 2006) that provides access to a broad array of computational methods used to analyse genomic data via the Broad Institute website (GenePattern software suites). *HeatMapViewer*, one of the GenePattern tools, which allows the transformation from the numeric findings into graphical representations and provides a global view on the features interaction without any form of programming syntax, is used to support the findings of the GANN prototype. The colour-coding scheme in the HeatMapViewer provides a quick coherent view of feature correlations.

Figure 4.2 presents the screen shot to produce a heat-map on the GenePattern environment. In the HeatMap Viewer module, the expression values are standardised with the mean value, ranging from -3 to 3, and the standard deviation of 1. These values are presented in 2 different colour shades, i.e. red and blue. High expression values are displayed in red indicating with positive values and the negative value representing low expression values which is displayed in blue. Intermediate expression values are displayed in different shades of red and blue. We used the default HeatMap Viewer settings to generate heat-maps for our findings on microarray data.

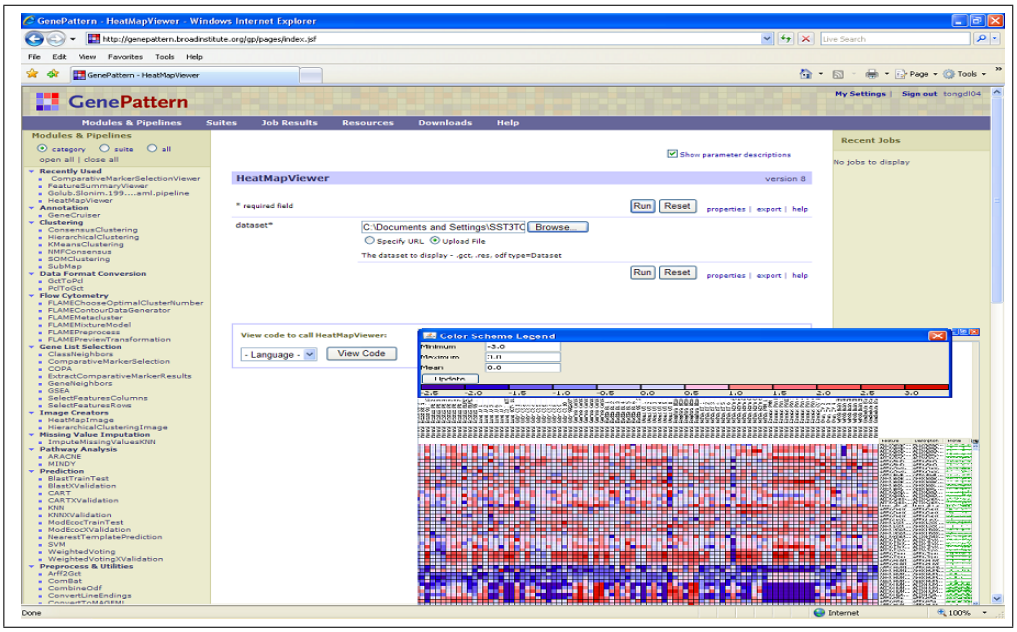


Figure 4.2: The screen shot generating heat-map using HeatMap Viewer.

4.1.4 Tool for visualising the findings

Microsoft Office Excel is a spreadsheet application written and distributed by Microsoft. It is featured with calculation functions, graphing tools, pivot tables and VBA macro programming language. In this thesis,

the calculation functions and graphical tools of the Microsoft Office Excel (version 11.0) are used.

4.1.5 Tool for visualising data sets

R project is a language and environment for statistical computing and graphics developed at the Bell Laboratories. It is available freely under the terms of the Free Software Foundation's GNU General Public License in source code form from the R website (R Development Core Team, 2006). In R, the *cmdscale()* function has been used to visualise sample patterns within the data. Figure 4.3 shows the screen shot on the *cmdscale* programming code in R environment. The *cmdscale* function performs classical multidimensional scaling (MDS) in visualising similarities/dissimilarities between data points (i.e. samples) based on several fitter variable points (i.e. features) to project data points in a two-dimensional graph. The results are evaluated by comparing the distances between data points on the proximity matrix with the Euclidean distances and a measure of goodness-of-fit. In R project, we use the default parameters on the Euclidean distance and goodness-of-fit.

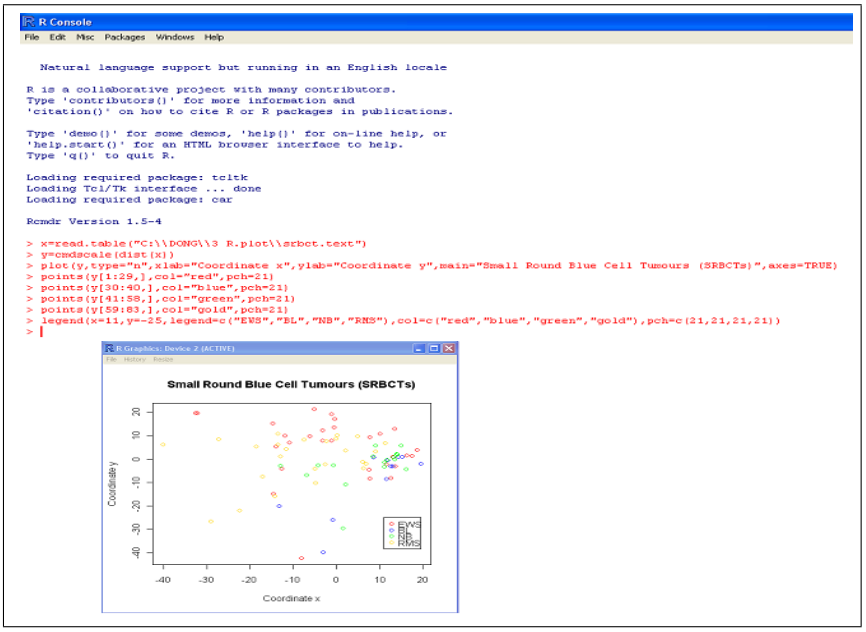


Figure 4.3: The screen shot visualising data pattern using multidimensional scaling (MDS) on the R environment.

4.2 Microarray Data Transposition

Like all kind of data sets, microarray data sets is designed in a two-dimensional table. However, in practice, microarray data sets have different arrangement in the data layout, as showed in Figure 1.1 on page 4.

Typical microarray data sets arranged individual instances, i.e. samples, in the columns of the table and its associated attributes, i.e. genes, in rows. The reason for such data arrangement is that most microarray

software utilise spreadsheet features from the Microsoft Office Package (version 11.0 and below) which can only hold maximally 256 columns, but has an enormous number of rows up to 65536 rows. With the introduction of the new Microsoft Office Package in year 2007 (version 12.0), this new version of Excel software can stores up to 1048576 rows and 16384 columns of data. However, many of the laboratories have not transformed their data to this new version of spreadsheet system. Thus, data transposition is still required when working with current publicly available microarray data.

In this thesis, a C++ program is specially written to transpose the microarray data into the standard format that is compatible to the computing algorithm.

4.3 Architectural Design of the Prototype

Referring to the conceptual design of the GANN model in Figures 3.11 page 84, a high level of architectural design based on the program functions in the prototype is presented in Figure 4.4. The prototype contains three main modules that are coherent with the components in the design phase as discussed in the previous chapter, which include population initialisation, fitness computation and pattern evaluation. These modules are integrated as one whole bespoke program which is implemented using C++. The pseudocode of the GANN prototype is depicted in Figure 4.5 and its coding statements can be found in Appendix A.

4.3.1 Parameter Setting Interface

Before the prototype being executed, the parameters for both GA and ANN need to be defined. As this is only the prototype of our approach, there is no graphical user interface (GUI) environment for such definition. Instead, it was made directly on the coding statement shown in Figure A.1 in Appendix A. Table 4.2 presents the the parameter settings of the prototype.

Table 4.2: The summary of the GANN interface parameters.

Parameter	Description
RUN_COUNT	The whole GANN process cycle. The default value is *5000*.
INPUT_ROWS	The number of samples to be read from the data set.
INPUT_COLS	The number of features to be read from the data set.
CLASS_COUNT	The number of class clusters in the data set.
HIST_COUNT	The number of ranked features produced by GANN. The default value is consistent with the *INPUT_COLS* parameter.
HIST_MIN	The minimum frequency of the correctly labelled samples in each feature produced by GANN. The default value is *0*.

Continued on Next Page...

Table 4.2 – *Continued*

Parameter	Description
HIST_MAX	The maximum frequency of the correctly labelled samples in each feature produced by GANN. The default value is consistent with the *INPUT_ROWS* parameter.
GA_POPSIZE	The population sizes to be examined {100, 200, 300}. User can input integer values that is higher or lower than the defined sizes.
GA_EVALS	The evaluation sizes to be examined {5000, 10000, 15000, 20000, 25000, 30000, 35000, 40000, 45000, 50000}. User can input integer values that is higher or lower than the defined sizes.
GA_PRECISION	The fitness confidence (accuracy) level of GANN. The default value is consistent with the *INPUT_ROWS* parameter.
GA_MUTATIONDIST	The mutation point (the point on the chromosome to be changed) of the string (offspring). The default value is *0.5*.
GA_MUTATIONRATE	The mutation rate. The default value is *0.1*.
GA_XFACTOR	The cut-point on the parent strings. The default value is *2* (single-point crossover).
GA_CROSSOVER	The execution of crossover function. The default setting is *true*. User can de-activates this function by amending the parameter to *false*.
GA_TOURNAMENTSIZE	The size of the tournament selection. The default value is *2*.
MLP_ISIZE	The input nodes in the input layer. The default value is *10*.
MLP_HSIZE	The hidden nodes in the hidden layer. The default value is *5*.
MLP_OSIZE	The output nodes in the output layer. The value must consistent with the *CLASS_COUNT* parameter.
MLP_ACT	The activation function. The available functions include *binary_sigmoid, linear, tanh, threshold*.
MLP_SIZE	The network weights including the bias nodes. User can amend the number of bias nodes in the hidden and output layers.

Three data set parameters, which are *INPUT_ROWS*, *INPUT_COLS* and *CLASS_COUNT*; were used to describe the experimental data set. The INPUT_ROWS is the total number of samples in the data set, the INPUT_COLS is the total number of features in the data set and the CLASS_COUNT is the number of classes in the data set.

The layout of the results is defined by three parameters, i.e. *HIST_COUNT*, *HIST_MIN* and *HIST_MAX*. The HIST_COUNT indicates the total number of ranked features to be displayed by the prototype, the

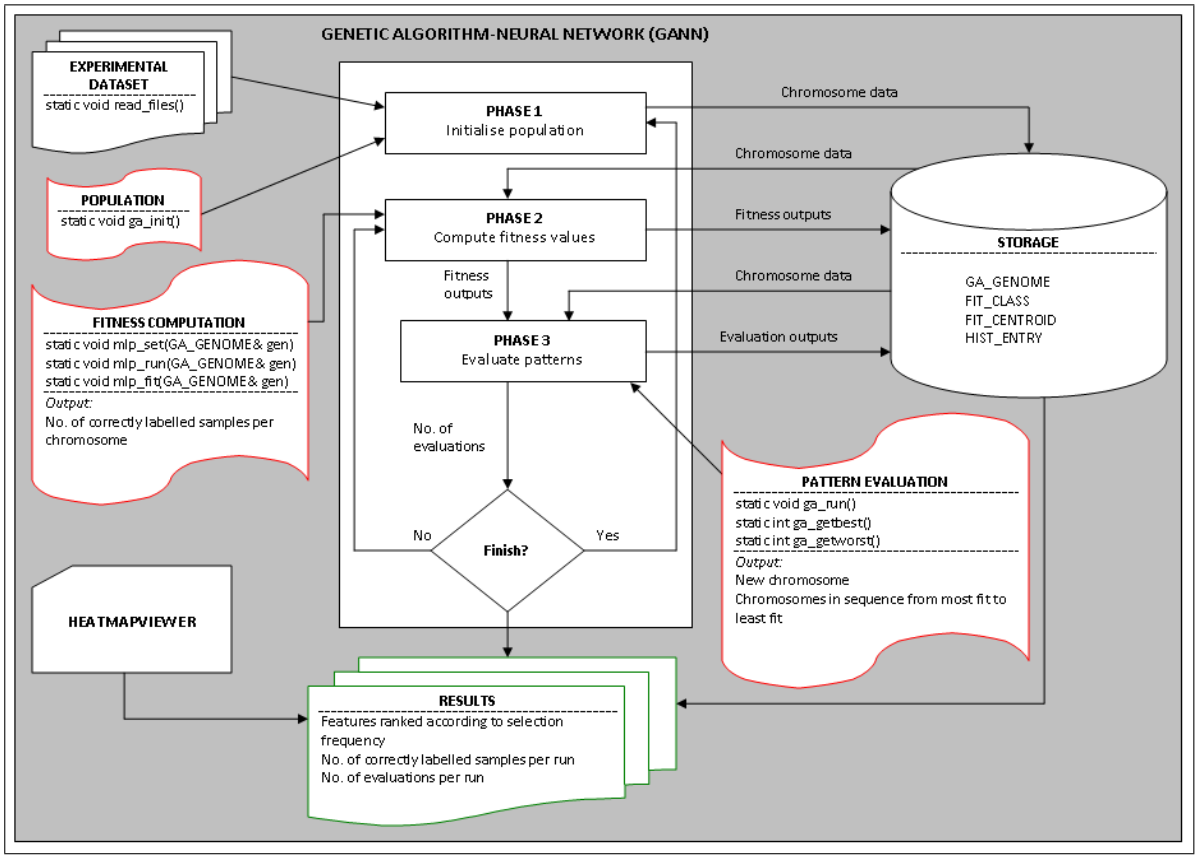


Figure 4.4: GANN Prototype: A high level of architectural design.

HIST_MIN and the HIST_MAX indicates the cut-off range for the minimum and the maximum correct labelled sample frequency for each ranked feature to be displayed by the prototype.

For a GA to optimise chromosomes in the population, eight parameters were included, which are *GA_POPSIZE*, *GA_EVALS*, *GA_PRECISION*, *GA_MUTATIONDIST*, *GA_MUTATIONRATE*, *GA_XFACTOR*, *GA_CROSSOVER* and *GA_TOURNAMENTSIZE*. The GA_POPSIZE indicates the population size, the GA_EVALS indicates the evaluation size, the GA_PRECISION is the fitness accuracy of the prototype, the GA_MUTATIONDIST and GA_MUTATIONRATE are the mutation point and the mutation rate for mutation operator, the GA_XFACTOR is the cut point for crossover operator and the GA_TOURNAMENTSIZE is the number of chromosomes competed in the tournament.

To compute the fitness values for GA chromosomes, five ANN parameters were used, i.e. *MLP_ISIZE*, *MLP_HSIZE*, *MLP_OSIZE*, *MLP_ACT* and *MLP_SIZE*. The first three parameters (MLP_ISIZE, MLP_HSIZE and MLP_OSIZE) indicate the number of nodes for the input, hidden and output layers, respectively. The MLP_ACT is the activation function for the hidden layer and the MLP_SIZE is the total ANN weights, including the bias weights.

The *RUN_COUNT* and *GA_EVALS* parameters are used to stop the prototype when the desired solution is

```
INITIALISE GANN parameters

REPEAT until termination criteria A (Max. no. of iteration) is satisfied {
   Generate GA population
   Calculate fitness values for each string in the population {
      DO WHILE EOF {
         a. Generate network weights
         b. RUN ANN
         c. Compute centroid values for each class using target output
         d. Calculate distance between samples and classes
         e. Label samples to its nearest class
      }
   }
   REPEAT until termination criteria B (Max. no. of fitness evaluation ||
            predefined precision value) is satisfied {
      Select 2 strings as parents for reproduction
      Perform GA operators {
         For network evolution {
            a. Crossover 2 set of network weights to produce new set of network weights
            b. Mutate the new set of weights
         }
        For feature evolution {
            a. Crossover 2 strings to produce new set of string
            b. Mutate the new string
         }
      }
      Calculate fitness value for new string
      Replace the least fit string with the new string
   }
   Calculate the number of correctly labelled samples
}

PRODUCE summary results
```

Figure 4.5: The pseudocode of the GANN prototype.

achieved (*GA_PRECISION*). The RUN_COUNT parameter performs external looping on the entire extraction process and the GA_EVALS parameter internally repeating the fitness computation module.

4.3.2 Population Initialisation Phase

In this module, two program functions are executed, which are the *read_files()* and *ga_init()* functions. Figure 4.6 presents the module's flowchart.

This module begins when the data set is retrieved by the prototype. The *read_files* function reads the entire data set and stores the data in a two-dimensional input matrix with rows indicate samples and columns indicate features. This input matrix is ready for use by the subsequent functions. After the input matrix was created, subsets of the features from the matrix is extracted using the *ga_init* function. The number of extracted features is reliant on the population size and the chromosome size. For example, in the population

size of 300 and the chromosome size of 10, the *ga_init* function will extract 3000 features (300 x 10) from the input matrix. The extracted feature subset is stored in the *GA_GENOME* array for later use. The subsequent program functions will process the feature subsets in the array, rather than the features in the input matrix. The configuration statement for the *GA_GENOME* array and the coding statement for the *ga_init* function are presented in Figures A.2 and A.4a in Appendix A, respectively.

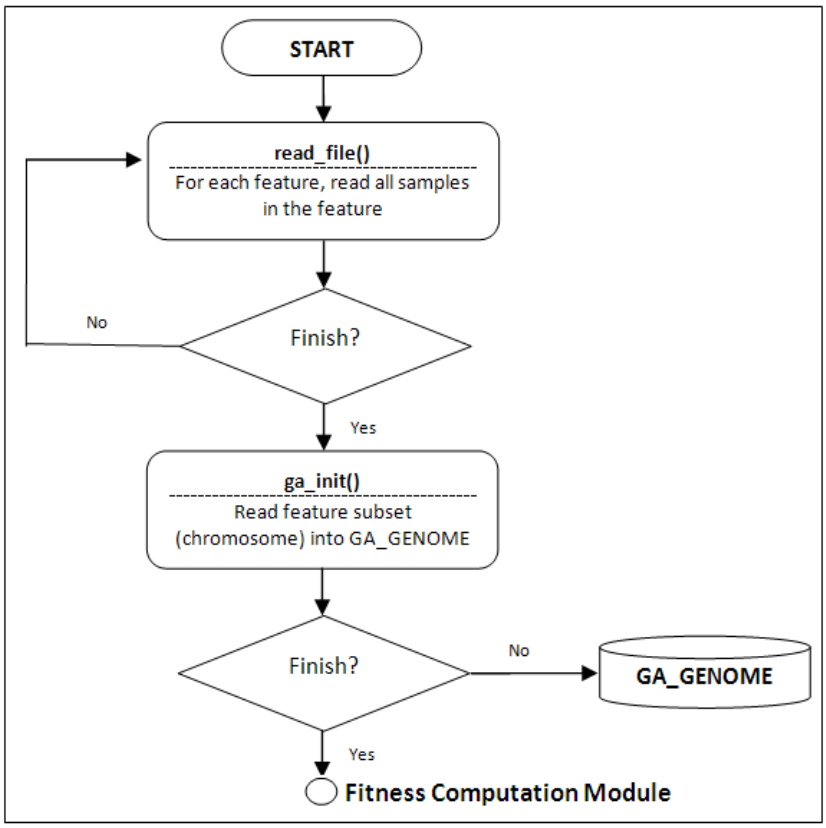


Figure 4.6: Population Initialisation Phase: The system flowchart.

It is important to note that, other than the data transposition for microarray data, there is no data pre-processing steps, such as pre-filtering, imputation techniques and normalisation methods, are applied to the experimental data sets, as is in the traditional way for analysing microarray data. The reason for excluding such traditions is that we would like to avoid any of the sort of errors that may induced by these preprocessing steps.

4.3.3 Fitness Computation Phase

After the feature subsets are loaded in the *GA_GENOME* array, a 3-layered feedforward ANN with the structure of 10-5-O is constructed to calculate the fitness values for each feature subset in the array. Three program functions are created in this module are the *mlp_set(), mlp_run()* and *mlp_fit()* functions. The overall system flowchart of this module is presented in Figure 4.7 and the coding statements of these functions can

be found in Figure A.3 in Appendix A.

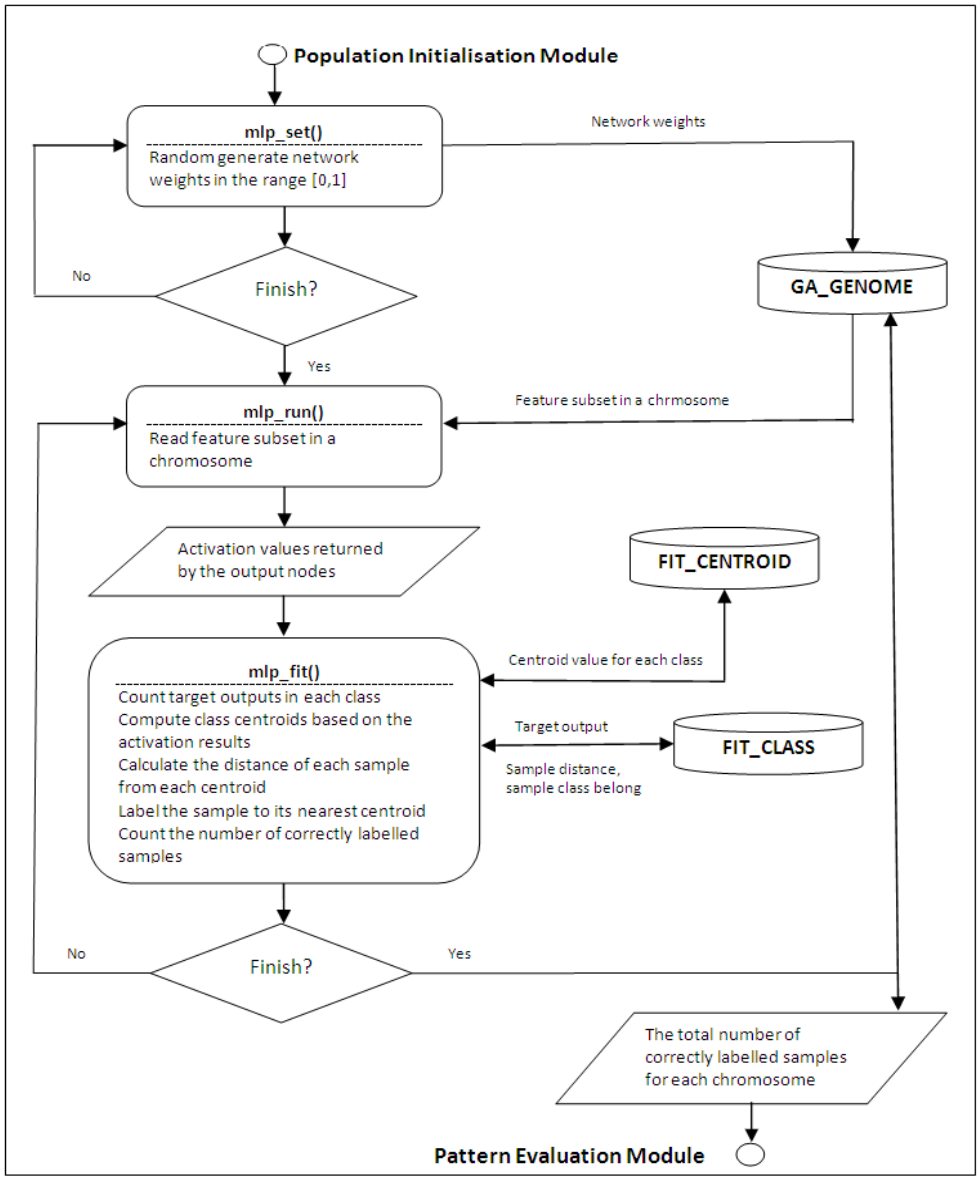


Figure 4.7: Fitness Computation Phase: A high level system flowchart.

In this module, the *mlp_set* function is first executed to initialise the network weights for each network layer and the bias values for the hidden and output layers. The initial weight values are randomly selected from the range of [0,1]. The *mlp_set* function then reads the input nodes from the *GA_GENOME* array. In other words, 10 features will be used as the input nodes in the ANN when the chromosome size is set to 10. The *mlp_run* function is then executed to process each sample of the data set for which the network must be run and calculates activation outputs, which is then used by the *mlp_fit* function to calculate the fitness of the feature subset. The fitness output of the *mlp_fit* function is stored in the *GA_GENOME* array.

4.3.3.1 MLP_SET() FUNCTION

The *mlp_set()* function is the looping function based on the *for* loop command, used to construct the network model based on the network size defined in the parameter setting interface module.

4.3.3.2 MLP_RUN() FUNCTION

After the network is constructed, the *mlp_run()* function processes samples for each chromosome in the *GA_GENOME* array. Once the input activations of the network are set, using the activation value and the element values, the network can be run in sequential order. This process is often called "feedforward" in the context of ANN.

In the *mlp_run* function, as shown in Figure 4.8, the first input weight for the first input node is read from the *GA_GENOME* array to compute the activation value for the input node $f(A_{\text{input}})$. The network is then process the second input weight for the second input node. This process is repeated until all the input nodes have assigned activation values. In the input layer, the *identity* activation function ($f(x) = \Sigma x_i$) is used to calculate input activation values.

The network then process the nodes in the hidden layer by reading the first hidden weight and its bias from the *GA_GENOME* array to calculate its hidden activation value $f(A_{\text{hidden}})$. In the hidden layer, each hidden node is supported by a bias node B, meaning that 5 bias nodes will be generated by the network for 5 hidden nodes. This process is iterated until all the hidden nodes have assigned activation values. In the hidden layer, four commonly used activation functions *f(x)*, i.e. *sigmoid, linear, hyperbolic tangent (tanh)* and *threshold* functions are examined in this thesis. The equations for these activation functions can be found in Figure 3.10 on page 81. The hidden activation function can be expressed as $f(x) = \Sigma x_{ij} + B_j$, in which $f(x)$ is the selected activation function, x_{ij} is the weight for the input node to hidden node, and B_j are the bias weights for the hidden nodes.

Based on the outputs from the hidden layer, the network calculates the activation value $f(A_{\text{output}})$ for each output node in the output layer. A two dimensional matrix A, as shown in Expression 4.1, that holds the activation values from the output layer is produced and this matrix is used by the *mlp_fit* function to calculate the fitness value for each feature subset in the *GA_GENOME* array.

$$A = \begin{bmatrix} a_{1s_1} & a_{2s_1} & \dots & a_{ks_1} \\ a_{1s_2} & a_{2s_2} & \dots & a_{ks_2} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1s_n} & a_{2s_n} & \dots & a_{ks_n} \end{bmatrix}, \tag{4.1}$$

where a_1 is the activation value computed by the first output node (i.e. class 1) and a_2 is the activation value computed by the second output node (i.e. class 2).

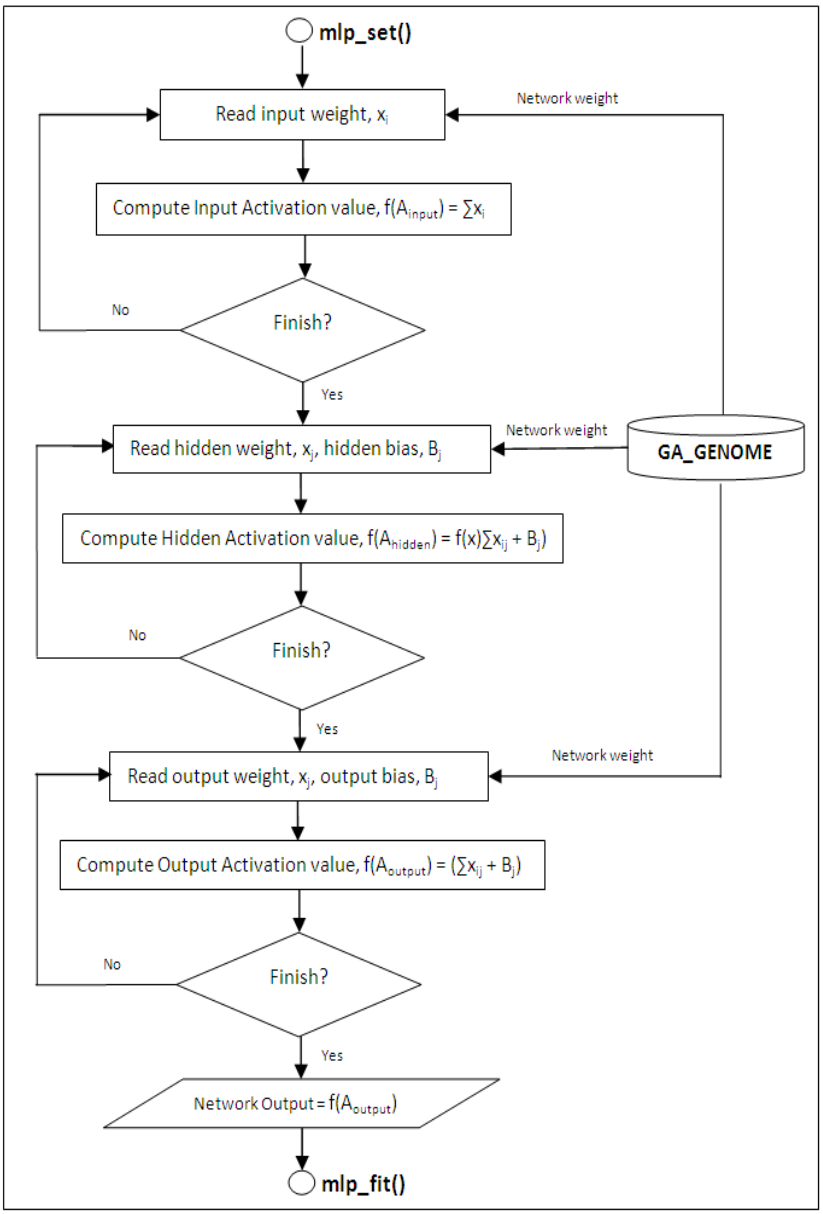


Figure 4.8: Fitness Computation Phase: A low level flowchart on the mlp_run() function.

4.3.3.3 MLP_FIT() FUNCTION

To classify the sample to its nearest class, the *mlp_fit()* function is invoked. Figure 4.9 presents the low level flowchart for *mlp_fit* function. This function plays a key role in calculating the fitness value for each chromosome in the population. In this function, the network firstly reset the network output in the *FIT_CENTROID* array to zero to ensure that the network performance is not interfere by the predecessor network. Using the target output T (see Expression 4.2) from the *FIT_CLASS* array, the network counts the number of

samples in each class and these values are used to calculate the centroid value for each class C_k using the activation results from the *mlp_run* function (see Expression 4.1). The centroid values are stored in the *FIT_CENTROID* array, which will be used for labelling samples to its nearest class. The configuration statements of the *FIT_CENTROID* and *FIT_CLASS* arrays can be found in Figure A.2 in Appendix A.

$$T = \begin{bmatrix} t_{s_1} \\ t_{s_2} \\ \vdots \\ t_{s_n} \end{bmatrix}, \tag{4.2}$$

where t_{s_n} is the target (actual) output for sample n in the data set.

When the centroid of the classes in the data set is calculated, the network computes the sample distance for each class. In the distance proximity measure, the square root for the subtraction of output activation value for each sample to each class centroid is calculated ($\sqrt{(A_{ik} - C_{ik}}^2$) and the results are stored in the *FIT_CLASS* array, which is used to label the sample to the nearest class. Expression 4.3 presents a two-dimensional matrix used in the *FIT_CLASS* array to store the proximity values of samples to each class centroid.

$$O = \begin{bmatrix} o_{1s_1} & o_{2s_1} & \dots & o_{ks_1} \\ o_{1s_2} & o_{2s_2} & \dots & o_{ks_2} \\ \vdots & \vdots & \vdots & \vdots \\ o_{1s_n} & o_{2s_n} & \dots & o_{ks_n} \end{bmatrix}, \tag{4.3}$$

where o_{ks_n} is the network output for sample n for class centroid k.

Based on the proximity values in the matrix O, the network compares values of both the centroid and the sample and labelled sample to the class with the smallest discrepancy value. Thus, the matrix O in the *FIT_CLASS* array is updated as:

$$\Delta O = \begin{bmatrix} o_{1s_1} & o_{2s_1} & \dots & o_{ks_1} & \text{CLASS 1} \\ o_{1s_2} & o_{2s_2} & \dots & o_{ks_2} & \text{CLASS 2} \\ \vdots & \vdots & \vdots & \vdots & \\ o_{1s_n} & o_{2s_n} & \dots & o_{ks_n} & \text{CLASS K} \end{bmatrix}. \tag{4.4}$$

The matrix ΔO is the actual output generated by the network. To count the number of correctly labelled samples for each chromosome, the network output is compared to the target output. For each correctly labelled sample, a constant value of 1 is added to the chromosome's fitness score, which is saved in the *GA_GENOME* array.

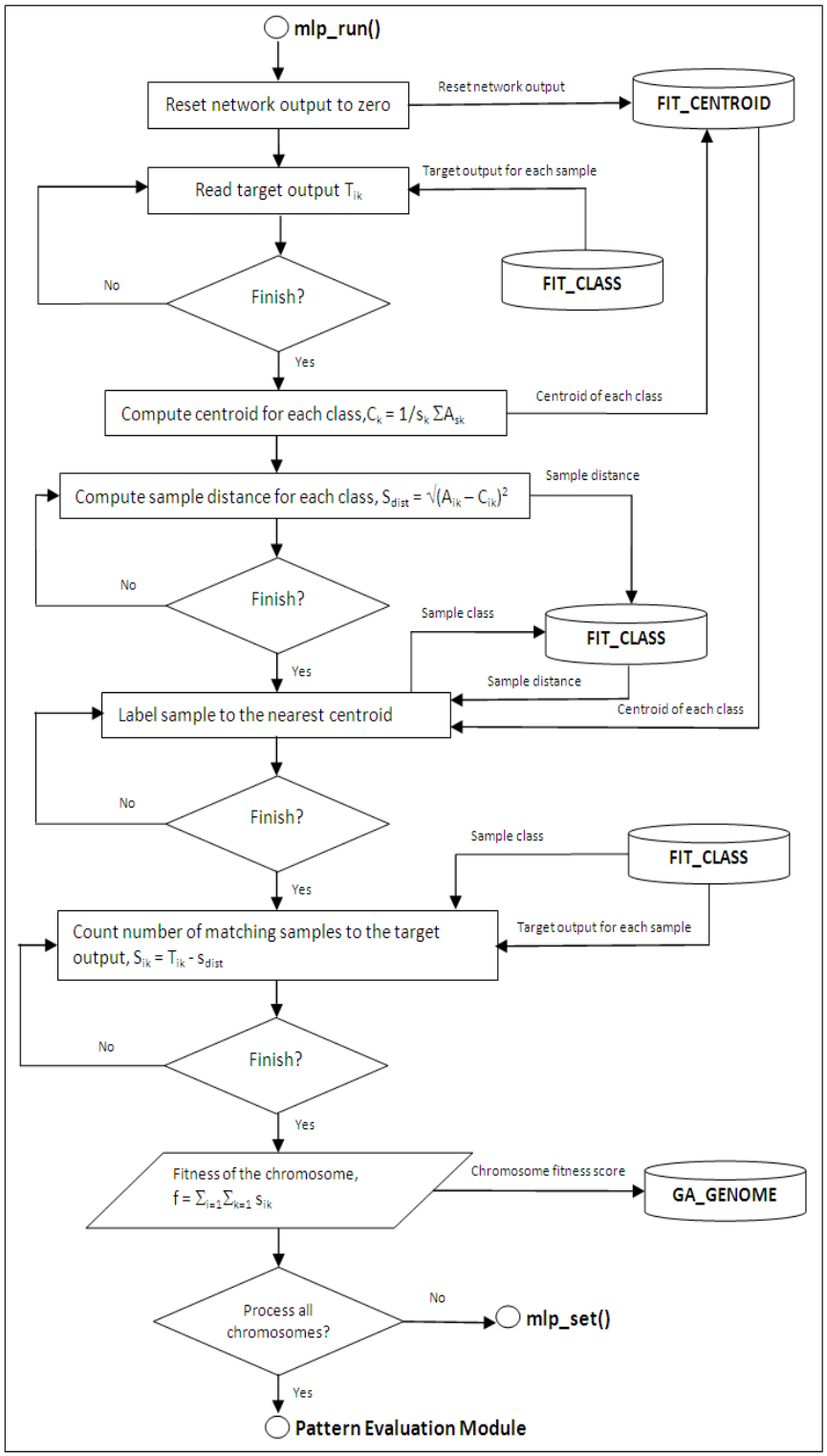


Figure 4.9: Fitness Computation Phase: A low level flowchart on the mlp_fit() function.

The entire fitness computation process is repeated until all chromosomes in the population have been assigned a fitness value and the termination criteria defined in the parameter setting interface module are met.

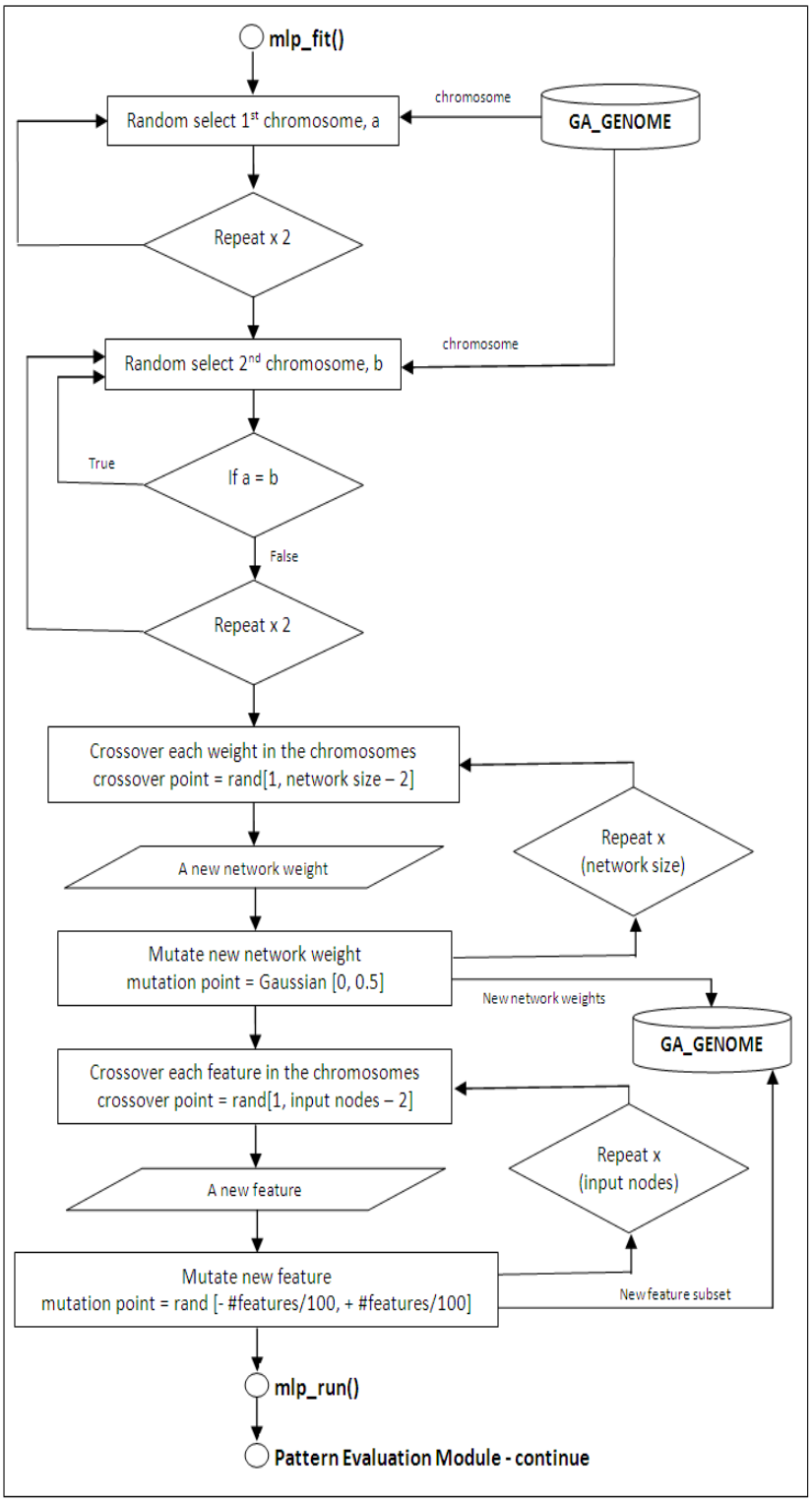
4.3.4 Pattern Evaluation Phase

When all chromosomes have been assigned fitness values, the evaluation process begins by first identifying a pair of chromosome from the *GA_GENOME* array for producing new offspring. The *ga_run()* function is used to execute such process and the ANN process (fitness computation module) is evoked to compute the fitness value for this offspring. The GA compares all chromosomes in the array to identify the least fit chromosome, which is then replaced by the new chromosome; and sorts these chromosomes in sequential order, ranking from the best fit chromosome to the least fit chromosome. The comparison process is performed using the *ga_getworst()* function and the sorting process is invoked by the *ga_getbest()* function. The latter two functions are invoked in the *ga_run* function. In other words, the *ga_getworst* and *ga_getbest* functions are the subfunctions executed by the *ga_run* function. Figure 4.10 presents the system flowchart of the module and the coding statements for this module can be found in Figure A.4 in Appendix A.

4.3.4.1 ga_run() function

The *ga_run()* function plays a key role in evolving chromosomes and it, in fact, represents the entire GA evolution process and to stop the prototype from over-learnt. In the *ga_run* function (see Figure 4.10a), the GA randomly selects a chromosome from the *GA_GENOME* array for comparison. The GA is then selects another chromosome from the array and the fitnesses of these two chromosomes are compared. The chromosome with high fitness value is chosen as the parent chromosome a. Such comparison process is known as *tournament selection* in GA. Similar process is performed to find the parent chromosome b. When the GA has identified parent chromosomes, these chromosomes are cross-overed to produce new offspring. This is known as *reproduction* in the context of GA.

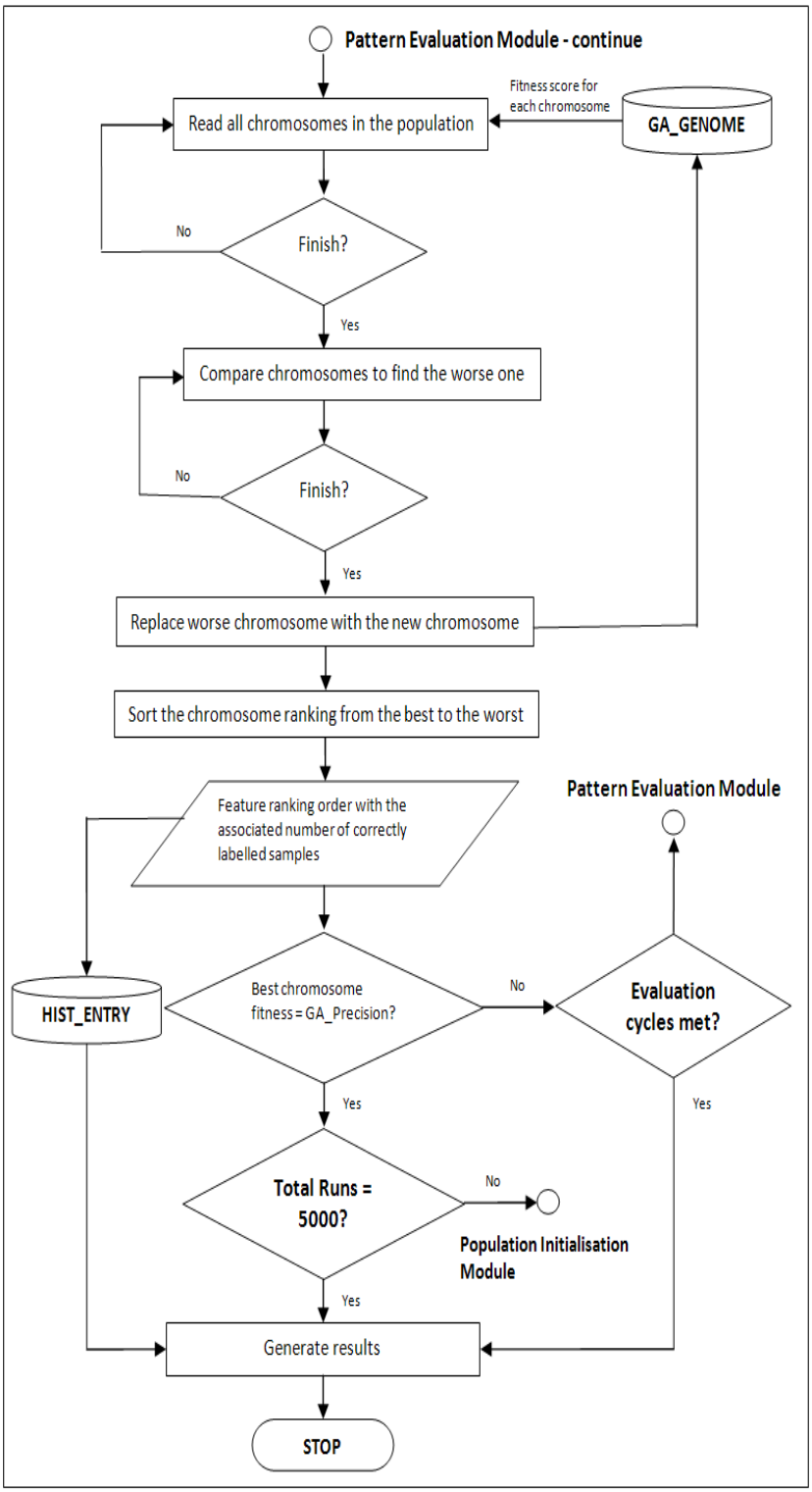
In this module, two types of reproduction process are performed. They are network weight optimisation process and conventional GA process (i.e. feature subset optimisation). For the network weight optimisation process, the weight of the first input node from both parent are cross-overed based on the random crossover point (i.e. cut point) in the range of [1, network size - 2], to obtain the new input weight for the first input node in the new network. This new weight is then mutated with the Gaussian range [0, 0.5] and the mutated weight is saved in the *GA_GENOME* array. The process is repeated until weights for all nodes in both parent are optimised. The similar optimisation process is used to optimise feature subset in the new offspring. For feature subset optimisation, the crossover point of the features is selected randomly in the range of [1, Input nodes - 2] and the mutation point is selected randomly in-between the range [-z, +z], in which the z value is the division of the entire features in the data set over 100. In other words, for the data set containing 7000 features using the network structure 10-5-2, the crossover point for new network is [1, 65], the crossover and mutation points for new feature subset is [1, 8] and [-70, 70], respectively.



(a) The ga_run() function.

Figure 4.10: Pattern Evaluation Phase: A low level flowchart on ga_run() function.

The fitness of this new offspring is calculated by repeating the processes in the *fitness computation module* and the comparison between chromosomes is carried out by the *ga_getworst* function and the feature ranking is performed by the *ga_getbest()*. The processing step in these two functions is presented in Figure 4.10b.



(b) The ga_run() function.

Figure 4.10: – Continued

- **ga_getworst()** In this function, all chromosomes in the *GA_GENOME* array are copied to the temporary matrix M_1. Then the comparison between chromosomes begins by comparing the first two chromosomes. The least fit chromosome is preserved in another temporary matrix M_2 and the other

chromosome is removed from the matrix M_1. The second chromosome from the matrix M_1 is retrieved and compared with the chromosome in the matrix M_2. The process is repeated until there are no chromosome left in the matrix M_1. The chromosome in the matrix M_2 is considered as the worst chromosome in the population and this chromosome is removed from the *GA_GENOME* array. The new offspring is replaced the position of the removed chromosome in the array. The contents of the matrices M_1 and M_2 are emptied so that these matrices can be reused by other functions.

- **ga_getbest()** When the new offspring is introduced to the *GA_GENOME* array, the *ga_getbest* function is invoked by the *ga_run* function. In this function, all chromosomes in the *GA_GENOME* array are copied to the temporary matrix M_1. A chromosome y, randomly selected from the matrix M_1, is copied into the first pointer (i.e. fittest) in the temporary matrix M_2 for comparison. The second chromosome retrieved from the matrix M_1 is compared with the chromosome y in the matrix M_2. If this chromosome has higher fitness score than the chromosome y, the chromosome y is moved one step below the first pointer in the matrix and this new chromosome is allocated in the first pointer. If this chromosome has lower fitness score than the chromosome y, this chromosome is added at the second pointer after the chromosome y. This sorting process is repeated until there are no chromosome left in the matrix M_1 and the ranking of each feature in the chromosomes are stored in the *HIST_ENTRY* array.

4.3.5 Terminating the Prototype

Before the prototype stops running, the fitness score of the best chromosome is compared to the defined value in the GA_PRECISION parameter in the parameter setting interface module. If the fitness score matches the value in the GA_PRECISION parameter, the termination criteria based on the external iterating on the entire GANN process is performed (RUN_COUNT parameter). Otherwise, the termination criteria based on the internal looping on the fitness computation process is invoked (GA_EVALS parameter). For either criteria, a summary result is produced when the prototype stops. This result comprising features ranking based on their selection frequency in different classification accuracy, retrieved from the *HIST_ENTRY* array. The screen shot of the summary result processed by the *HIST_ENTRY* array is presented in Figure 4.11 and the configuration of the *HIST_ENTRY* array can be found in Figure A.2 in Appendix A.

4.4 Data Validation - NCBI Genbank & Stanford SOURCE Search System

Due to the rapid development of microarray annotations, most of the gene description, in both microarray experimental data sets, was not supported by the NCBI Genbank nor can it be found in the Stanford

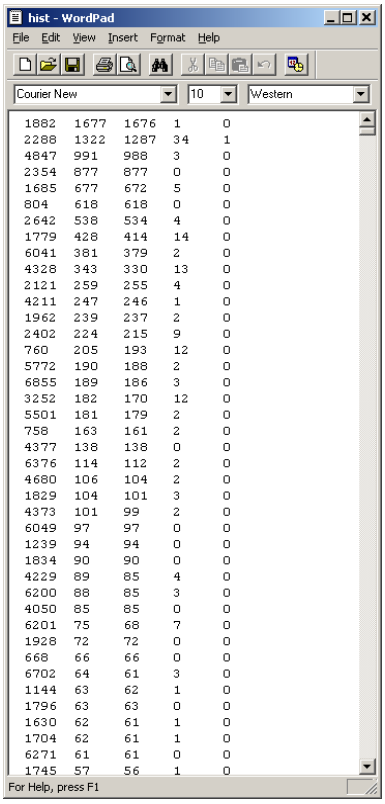


Figure 4.11: The screen shot of the HIST_ENTRY array. Column 1 indicates the indexes of the features in the data set; column 2 represents the total number of frequency selections for each index; column 3 represents the number of times the index has been selected with no mislabelled samples; column 4 shows the number of times the index has been selected with 1 mislabelled sample; and so on.

SOURCE search and retrieval system. Thus, great care is taken in validating the gene findings. Firstly, a list of the unique gene number, i.e. Affymetrix Accession Number for the ALL/AML genes and Clone Image Id for the SRBCTs genes, of the selected genes is uploaded into the SOURCE system. A variety of options on the type of gene information can be selected in the system. Then, the outcome of the SOURCE system is cross-referenced against the NCBI Genbank via the NCBI Entrez Gene system.

There are two main reasons for choosing the SOURCE system as one of the validation mechanisms in this thesis. Firstly, it is far more user friendliness than the NCBI Genbank and it can process multiple gene queries at the same time. Secondly, which is arguably the most important, that is, the SOURCE system can process Clone Id of the cDNA gene, whereas the NCBI Genbank is unable to recognise the Clone Id of the cDNA data.

To validate the gene findings of the GANN prototype, the following steps are performed:

1. A text-based document containing only the unique gene numbers of the selected genes, i.e. Accession Number of the ALL/AML genes and Image Id (clone id) of the SRBCTs genes, was produced.

2. The document is then uploaded to the SOURCE system (Stanford SOURCE search and retrieval

system) and the gene options: Gene Id, Gene Symbol and Gene Cytoband, are selected. When there is more than one possible outcome of a specific gene found in the system, no result will be produced for that gene. Most of the SRBCTs genes containing more than one possible result in the SOURCE system. This is mainly due to different cDNA protocol and annotation being used in different research laboratories.

3. The annotation results obtained by the SOURCE system was then compared with the annotations used in the experimental data sets. For the SRBCTs findings, an additional comparison on the annotation results in the SOURCE system and the up-to-date gene annotations from the original authors was performed. This is to prevent any incorrect annotation being used in our findings.

4. For gene annotations that are not supported by the SOURCE system, the annotation of these genes is entered individually into the NCBI Entrez Gene system (NCBI Genbank).

   (a) For ALL/AML findings, the accession number of the gene is used to perform the search. When more than one possible outcome is found in the Genbank, the annotation based on the term *"Homo Sapiens"* is used. In the case where the accession number is invalid, the gene description is used to perform the search in the system.

   (b) For SRBCTs findings, the gene description of the gene is used to perform the search. Even so, there is always some undefinable genes in the SRBCTs data set due to a generic description being used in the data.

5. A complete list of genes containing the original unique gene number, the gene id and the cytoband of the gene, was produced.

Figure 4.12 presents the steps for validating identified genes on the microarray data set.

4.5 Experimental Study

To support the theme of this research, a research methodology is applied on this thesis. There are several research methodologies in computer science, such as simulation, mathematical proof and experimental study. Our research uses the experimental study to test the hypotheses related to conceptual design and to system performance. We validate our solution by comparing the results from the prototype with the expected results from the synthetic data sets, the findings from the NCBI Genbank and the SOURCE search system, as well as the results reported in the original studies.

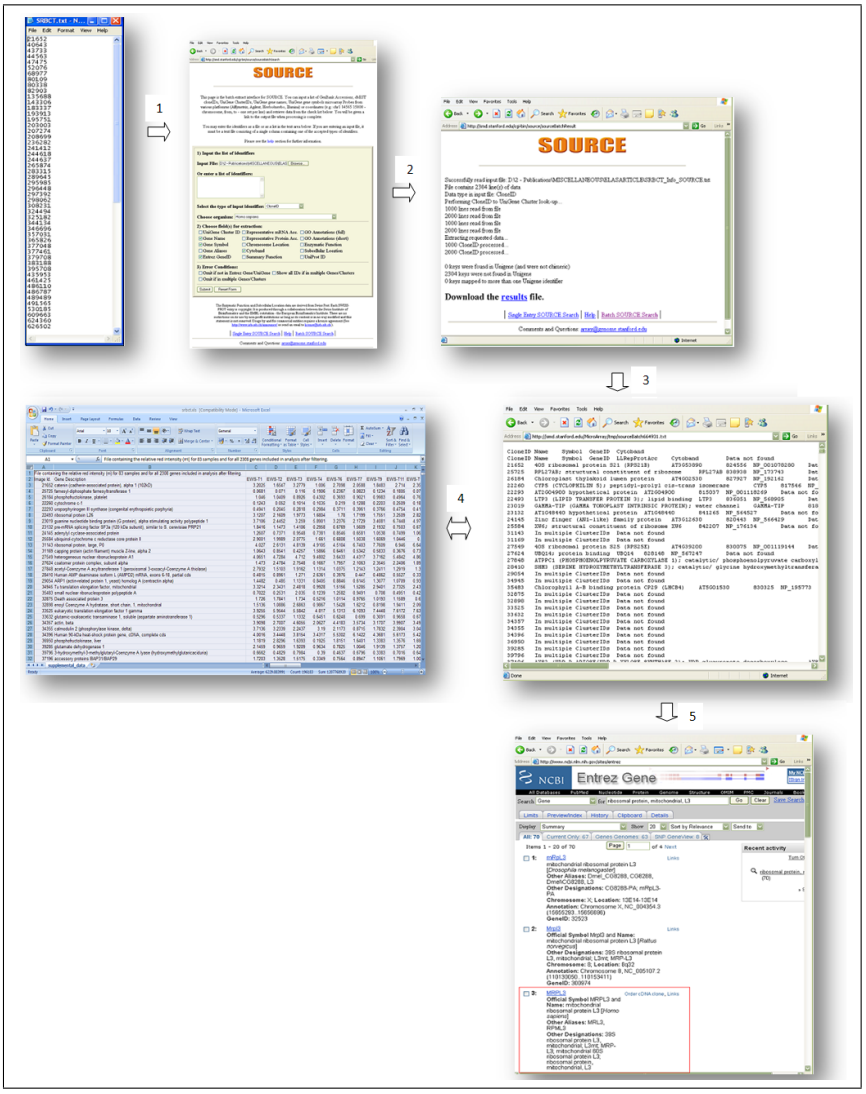


Figure 4.12: The steps for validating identified genes on the microarray data set. [1] A text-based document containing the clone id or accession number of the selected genes is uploaded to the SOURCE entry and retrieval system. Options such as Gene Id, Gene Symbol and Gene Cytoband in the SOURCE system are selected. [2] The SOURCE system processed the clone id and [3] a text-based document containing the defined options is displayed. When there is more than one possible outcome on a specific clone id, the system indicated the words "Data not found" or "Multiple clusters" in the output result. [4] The results was then compared with the gene descriptions in the data set to ensure the correct result is obtained for each identified gene. [5] The annotation for the gene with more than one possible outcome on the SOURCE system is entered into the NCBI genbank for validation. The annotation based on the term "Homo Sapiens" is used for cross-referencing our findings.

4.5.1 Objectives of experimental study

Reference is made to the hypotheses of this research as stated in Section 1.4 on page 12, emphasising the model simplicity, the model generalisability and the normalisation-free model, as well as the formation of biologically relevant results. To test the hypotheses, our experimental study measures the performance of four ANN activation functions: sigmoid (binary sigmoid), linear, tanh and threshold, for which each function is represented by a separate system in the prototype. This is to indicate which activation function has the best or worst performance when they are used to compute the fitness values of features in the same data

sets with similar parameter settings applied. Thus, the experimental study serves five purposes as follows:

The first purpose is to assess the overall performance of each system to handle raw, unprocessed microarray data sets. The term 'raw' used in this section refers to the original microarray data set, obtained from the respective repository, with no preprocessing steps performed on data values. In this experiment, the oligonucleotide microarray data set having large range between the maximum and the minimum values within a gene in the data set is examined.

The second purpose of the study is to assess the overall performance of each system to handle different types of microarray platforms. In this experiment, two microarray data sets with different array platforms and a different number of classes are examined.

The third purpose of the study is to assess the implication of two main GA features: population size and the number of fitness evaluations for analysing microarray data. This objective is to examine which population size is best for data with high feature dimensions and the minimal number of evaluations required for consistent results.

The fourth purpose of the study is to assess the generalisability of the prototype to select the most significant attributes from a large, imbalanced data set that contains multiple data representation. In this experiment, two bioassay data sets with different percentage of minority class (different number of active compounds) are examined.

The last purpose of the experimental study is arguably the most important, that is, to check the accuracy (correctness) of the work described here in extracting the desired features from the data sets. The accuracy of the results is checked by comparing the results obtained via the prototype and our expected results in synthetic data sets.

4.5.2 Experimental Data Sets

Six experimental data sets were used in the experimental study. These data sets comprise of microarray data sets (i.e. ALL/AML and SRBCTs), synthetic data sets (i.e. synthetic data set 1 and synthetic data set 2) and bioassay data sets (i.e. AID362 and AID688). The description of these data sets is presented in Section 3.1 on page 58.

4.5.3 Experiment Design

The experiments are designed according to the objectives of the experimental study, which has been mentioned in Section 4.5.1 and the hypotheses of this thesis, is as follows:

1. Varying sizes of data sets with varying numbers of samples allocated in each class cluster were used in the experiments. We performed trial experiments on several different sizes of data and it was found that data that had a class size between 2 to 3 and a ratio between samples and features is 1:100 showing a significant performance difference between the systems. Thus, the experiments were conducted based on the synthetic data with sample size varying from 67 to 100, feature size varying from 5000 to 10000 and class size varying from 2 to 3. Similar sets of experiments were also performed on the real-world microarray data sets which have a sample size varying from 72 to 83, feature size varying from 2308 to 7129 and class size varying from 2 to 4. This design serves the first two purposes of the experimental study.

2. Varying sizes of population and fitness evaluation on the GA were tested in the experiments. In the GANN prototype, the values of GA parameters, i.e. the population size, the fitness evaluations size, the crossover factor and the mutation rate, can be changed (see GANN interface in Figure A.1 in Appendix A). Trial experiments were conducted on these parameters and it was found that there was a minor influence of the mutation operator to the stability performance of the system, with mutation rate varying from 0.1 to 0.5. When the mutation rate more than 0.5 was applied, the system became unstable and different set of genes were produced in the repeated trial when similar set of parameters were used. This could be a precursor to the over-fitting problem. Therefore, we retained a small mutation rate, i.e. 0.1, in all systems.

   We also conducted trial experiments based on the population size varying from 100 to 700 and the fitness evaluation size varying from 1000 to 50000. The trial results showed that convergence began in most systems when the population size reached 300 and the fitness evaluation size 20000. However, there was not much difference in system performance for population size, ranging from 300 to 700. Therefore, the experiment was conducted on a population size varying from 100, 200 and 300. We performed the experiments on smaller fitness evaluation sizes, started from 5000 and each time, the evaluation was increased another 5000 cycles, until the maximum evaluation size of 50000 is reached. This design supports the third purpose of the experimental study.

3. Varying levels of fitness precision in the prototype were examined in the experiment. The default values of the precision parameter is usually consistent with the sample size, i.e. 100% fitness precision score. This value can be altered to different precision accuracies. The experiments based on three precision levels varying from 95% to 100% were tested. This design argues our statement in Section 1.2.4 on page 9 and supports the objectives of our research theme stated in Section 1.4 on page 12.

4. The comparison study based on the normalised and the raw microarray data set was performed to support our argument concerning the implication of data normalisation process addressed in Section

1.2.2 on page 8. The experiments based on the ALL/AML microarray data set was conducted.

5. The experiments based on two bioassay data sets with the tanh-based GANN system were performed. This design supports the fourth purpose of the experimental study.

All the above experiments were designed for the last purpose of the experimental study. The first and the second experiments were assessed based on the number of significant genes extracted by the system, the fitness accuracy of the system on the extracted genes and the processing time of the system. The integrity of the findings was evaluated based on the comparison studies conducted in previous work and from a molecular perspective.

4.6 Summary

In this chapter, we have discussed the tools used to support the theme of this thesis and explained the prototype of the method outlined in Chapter 3. An experimental study were conducted to evaluate the performance of the prototype from several aspects, including the effect of different precision levels, the population sizes, the fitness evaluation sizes and different activation functions.

With the vast development of microarrays, and there are many 'grey areas' that need to be further investigated, the existing feature selection models are unable to handle these areas. Questions, such as 'which genes trigger the development of specific cancer diseases?' and 'which genes shown the first sign of the recurrence of mutated cells?', are yet to be answered. These questions have been taken into consideration when we constructed the prototype which aims to provide an insight into the elementary genes and triggered genes in malignancy development. A fitness precision accuracy parameter is also built into the prototype, which allow users to closely monitor the pattern of the disease development from the beginning stage to the final stage.

In the next chapter, we carried out a comparative study of our results with the studies reported previously and show how the hybridisation of GAs and ANNs is suitable for analysing microarray data as well as the bioassay data.

Chapter 5

Experimental Results and Discussion

The prototype and the experimental study have been explained in the previous chapter. The objectives of this chapter is to present the findings of the prototype including discussions of the experimental results. Four GANN systems represent different activation functions, based on three population sizes and ten evaluation sizes which were compared in this chapter.

The relevant graphs and tables to support the objectives of the experimental study and the hypotheses of this thesis were produced in this chapter. Additional information on these graphs and tables can be found in Appendix B.

5.1 System performance with different data sets in different population sizes

In this section, we assess the overall system performance based on the synthetic data sets and the microarray data sets. Three figures, each representing an evaluation criteria, were produced. Each figure presents a high level view of system performance in terms of the the average number of the extracted genes in Figure 5.1, the average fitness accuracy of the extracted genes in Figure 5.2 and the average elapsed time (processing time) in Figure 5.3, based on three population sizes, ranging from 100, 200 to 300. The complete list of the extracted gene by each system in the four data sets are presented in Appendix B.

The four systems shown in the figures represent four different ANN activation functions used in the GANN prototype to compute the fitness values for each subset of genes in the population. These systems comprise of sigmoid, linear, tanh and threshold based.

5.1.1 The number of significant genes

With observation on the synthetic data sets in Figure 5.1, all systems have extracted an almost similar number of predefined genes in the synthetic data set 1 and in every population size. The *linear based system* has the highest number of predefined genes found in both synthetic data sets, i.e. on average 30 and 16 with the population size 300 in the synthetic data set 1 and the synthetic data set 2, respectively. The *tanh based system* has the lowest number of predefined genes found in the synthetic data set 2, i.e. on average 11 out of 30 genes in the population size 300.

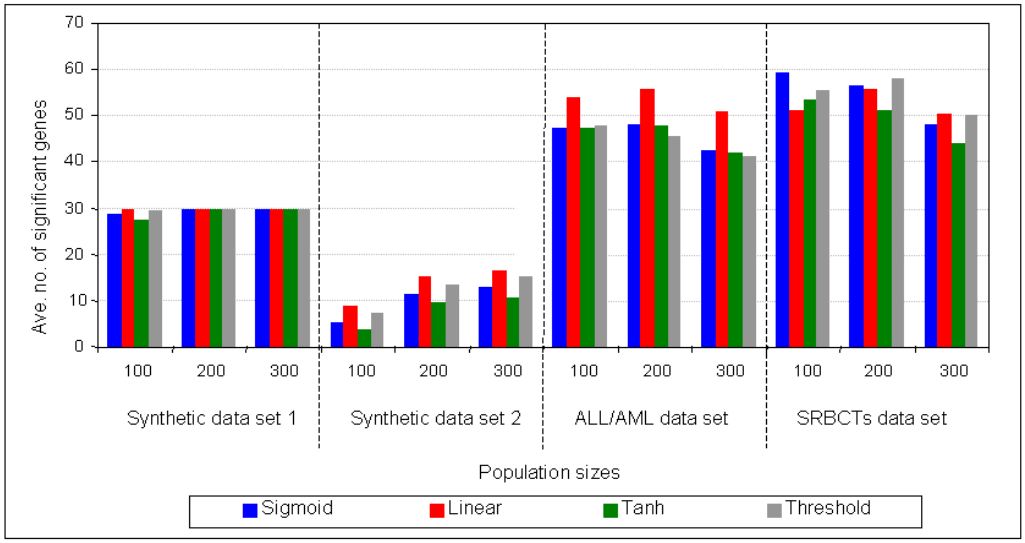


Figure 5.1: The average number of significant genes extracted by each system based on the selection frequency of 50 and above. The linear based system has the highest number of extracted genes in both synthetic data sets and the ALL/AML data set, while the other three systems have a similar performance in these data sets. None of the systems has a significant performance in the SRBCTs data set. In order words, all systems have a comparative performance in the SRBCTs data set.

For microarray data sets, there is a significant difference in the number of genes extracted by the *linear based* system and the other three systems in the ALL/AML data set. The linear based system has found more genes in every population size, while the other three systems have an almost equal number of genes was found in every population size. For SRBCTs data set, a significant fluctuation on the number of extracted genes by each system in every population size. In the population size 100, the sigmoid based system has the highest number of genes found, while the linear based system has the lowest number of genes found when a similar parameter was applied. When the population size is increased to 200, the threshold based system has found a slightly higher number of genes than the sigmoid based system, while the tanh based system showed a significantly decreased number of genes was found. A similar observation was made on the decreased number of extracted genes by the sigmoid and tanh based systems in a population size 300. When a similar population size was applied to the linear and the threshold based systems, both systems had

a higher number of genes extracted than the sigmoid and tanh based systems.

An interesting phenomenon was observed on the number of genes extracted on the data sets when the population size is increased. There are, on average, 11.47% (i.e. 11.64% on sigmoid based, 9.12% on linear based, 11.74% on tanh based and 13.39% on threshold based) and 14.76% (i.e. 18.89% on sigmoid based, 9.16% on linear based, 17.42% on tanh based and 13.57% on threshold based) of decreased number of significant genes extracted by all systems on ALL/AML and SRBCTs data sets, respectively. Conversely, a significant increased number of predefined genes (on average 55.05% on synthetic data set 2) by each system on both the synthetic data sets. This is due to the quality (characteristics) of the data sets. Both the synthetic data sets were generated based on different mean μ values in the Gaussian distribution; as a result, there are no extreme maximum and minimum values in the genes and outliers, which most microarray data suffer. Microarray data sets contain large value interval within a gene, especially the ALL/AML data set that contains large number of suppressed genes expression (i.e. negative values). With the increase in the population size, more chromosomes were exploited by the systems and consequently, more learning patterns (i.e. chromosomes) have been presented to the systems to model a set of general rules from these patterns. As indicated in Figure 5.1, all systems might under-learnt the rules in population size 100 due to insufficient learning patterns to model these rules. With the increase population size to 200, the systems started to explore more learning patterns, however, the rules were still imperfect in some way as microarray data sets are highly imbalanced on the scale of available samples and the number of associated genes per sample, and a high ratio of significant genes and noisy genes in the data sets. When the population size was increased to 300, the systems have been provided with sufficient learning patterns to model a set of better general rules with lesser number of but most significant genes in the data sets. We will discuss the significance of the extracted genes in Sections 5.3-5.4.

5.1.2 The fitness performance

With assessment on the fitness performance of each system in Figure 5.2 for synthetic data sets, there is no significant difference between the systems' performance in the synthetic data set 1 when the population size are 200 and 300. This is because all the 30 predefined genes in the data set have been identified by all systems, as is indicated in Figure 5.1 on page 119. Tables B.1-B.3 in Appendix B show the list of identified genes in the synthetic data set 1.

Even so, some predefined genes may be stronger, i.e. genes with significant expression values than the other genes, these genes have a significant influence on the fitness accuracy. This is indicated by slightly better fitness performance on the linear and the threshold based systems in the population size 300 in the synthetic data set 1. For synthetic data set 2, with a population size of 300, the *linear based* system has the best

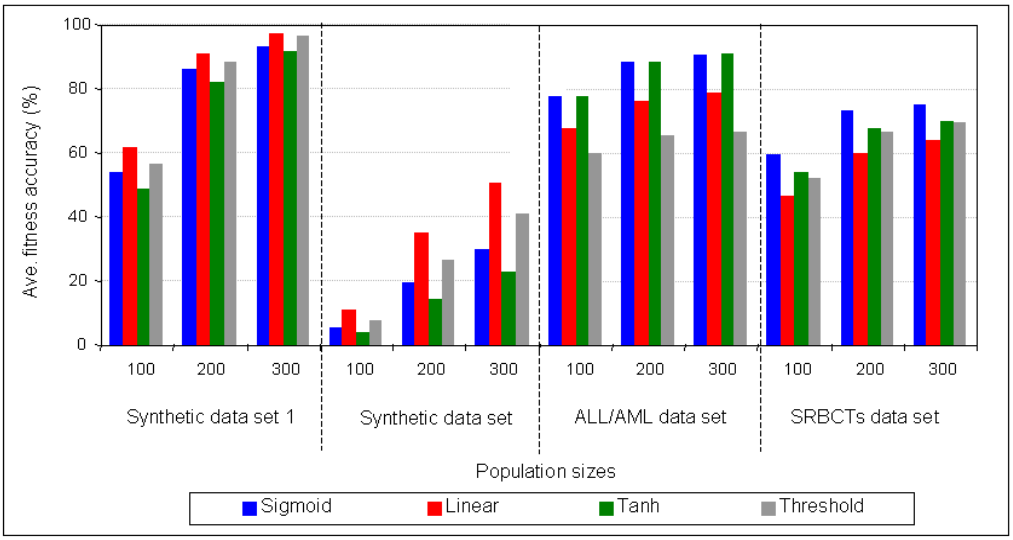


Figure 5.2: The average fitness performance by each system. A significant improvement in the fitness performance in each system with the increased population size. The linear based system has the best fitness performance in both synthetic data sets, whereas the tanh based system has the lowest performance. In the case of microarray data sets, both sigmoid and tanh based systems have the best fitness performance on the ALL/AML data set and the sigmoid based system has the highest fitness performance on the SRBCTs data set.

fitness performance (50.45%), while the *tanh based* system has the lowest performance (22.8%). This is consistent with the smallest number of predefined genes found by the tanh based system in the data set, as is shown in Figure 5.1. Tables B.4-B.6 in Appendix B shows the list of identified genes in the synthetic data set 2. The linear and the threshold based systems have a significant difference in the fitness performance in a population size of 300, although, both systems have a similar number of predefined genes identified. The fitness discrepancy is due to the influence of some stronger (fitter) predefined genes than the other predefined genes, which resulted in a better fitness confidence performance in the system. This inconsistency may be also due to the implication of some strong noisy genes, i.e. indexes 667, 2471, 2816, 2828, 4175, 4377, 4390 and 4883 (see Table B.6b in Appendix B), in the linear based system. This indicates that the linear based system capable in exploring stronger genes more efficiently than the other three systems. The identified genes were important to the subject of interest, however, it cannot assure that the identified genes are genes of interest.

For microarray data sets, the *sigmoid* and the *tanh based* systems have a better overall fitness performance than the linear based system, although more genes have been identified by the latter system, as is indicated in Figure 5.1. For ALL/AML data set, the *threshold based* system has the lowest fitness performance in every population size, while both the *sigmoid* and the *tanh based* systems have the highest performance in every population size. The low performance of threshold based system may be due to the involvement of multiple cancer subtypes within a cancer class in the data set (see Figure 3.1a on page 60). In the population size 300, the linear based system has an average fitness accuracy of 79% on 51 identified genes, compared

to both the sigmoid and the tanh based systems which achieved an average 91% of fitness confidence on 42 identified genes. This has confirmed our observation in the linear based system that it is able to detect the strongest genes, i.e. genes that can discriminate cancer classes, but not these genes might always underly the data. For SRBCTs data set, the linear based system had the lowest fitness performance, while the *sigmoid based* system outperformed the other systems. The sigmoid, the linear and the tanh based systems have a lower fitness performance in the SRBCTs data set than in the ALL/AML data set. Conversely, the threshold based system has a slightly better performance in the SRBCTs data set than in the ALL/AML data set. This might due to the tumour classes in the SRBCTs data set are not dependent while the ALL/AML data set is formed by subtypes of similar cancer classes, and the SRBCTs data set is a multiclass problem while the ALL/AML data set is a binary class problem.

5.1.3 The processing time

Finally we look at the processing cost of each system in each experimental data set. From the Figure 5.3, both the *linear* and the *threshold based* systems have the lowest processing time in all data sets. This is because both systems involved only the basic statistics operations, i.e. addition and multiplication, to calculate the fitness results rather than the sigmoid and the tanh based systems, which adopted more advanced statistics operations in the fitness computation process. This is indicated by a high ratio of elapsed time by both the sigmoid and the tanh based systems in almost all data sets, as is depicted in Figure 5.3.

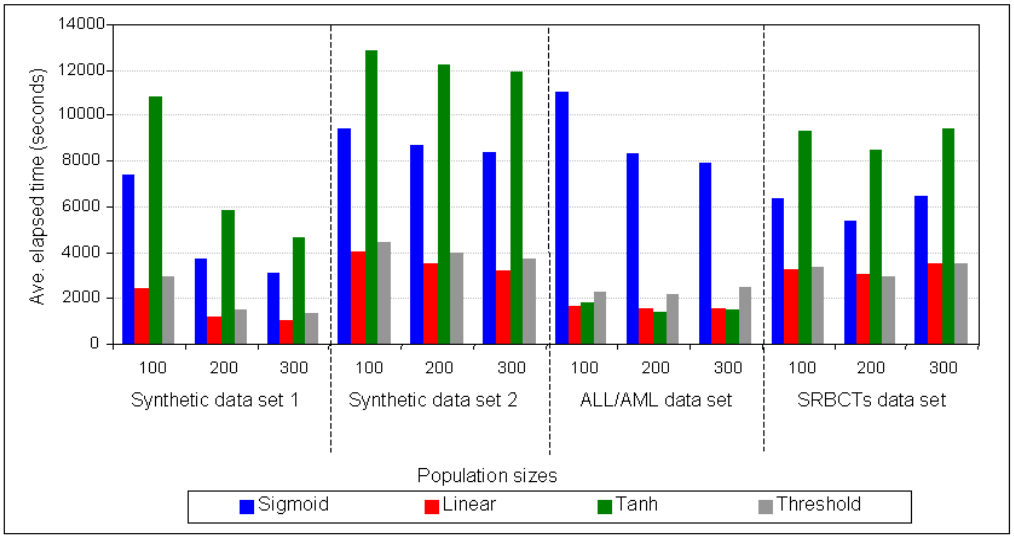


Figure 5.3: The average processing time for each system. Both the linear and the threshold based systems have the lowest amount of elapsed time, while the tanh based system has significantly increased amount of processing time in both synthetic data sets and the SRBCTs data set. The sigmoid based system has an intensive processing time in the ALL/AML data set when compared to the other systems.

The *tanh based system* has the highest processing time in both synthetic data sets and the SRBCTs data

set, and a low processing time in the ALL/AML data set. This is because the ALL/AML data set contains multiple subclasses of cancer sample within a known class (see Figure 3.1a on page 60) and the tanh based system normally worked better in nonlinear problems. The sigmoid based system has the highest processing time in the ALL/AML data set.

5.1.4 Discussion

We would like to bring attention to the performance of the tanh based system. As is indicated in Figure 5.3, the tanh based system has a very low processing time in every population size in the ALL/AML data set. Overall, it outperformed all other systems in the microarray data sets when a high fitness accuracy (see Figure 5.2) on the low number of extracted genes (see Figure 5.1) achieved by the system is taken into consideration. Two main reasons for its efficiency are: (a) the microarray data sets contain subgroup of cancer classes within a known class and have a large value interval within a gene in the data sets; and (b) the bipolar range (-1,1) in the tanh based system has produced two esteem output signals, i.e. positive and negative, in the output of the network, which has expanded the differentiation between the classes in the data sets. This has reduced the chances of mislabelling the sample into the wrong class.

The findings also showed that the performance of each system is very much dependent on the quality of the data set and, to some extent, the population size and the degree of statistics involved in the fitness computation process. Depending on the requirement of the study, each system has its pros and cons in terms of the quality of the extracted genes within a satisfactorily confidence range and in an acceptable processing time. For instance, the linear based system has the lowest processing time in all data sets, however, the identified genes may not underly the data. The sigmoid and the tanh based systems promise a high fitness confidence in larger population size, but they are computationally cost intensive. The threshold based system, meanwhile, is unable to effectively model data with multiple subclasses in a known class.

Even so, we are still able to identify two better systems from the findings in this experiment. The two superior systems are the linear and the tanh based systems. The linear based system has the best performance in the synthetic data sets for which all the 30 predefined genes in the synthetic data set 1 and on average 16 genes in the synthetic data set 2, have been identified and have been associated with high fitness confidence and low processing time for the selected genes in increased population size. The tanh based system has the best performance in microarray data sets for which it achieved a high fitness accuracy with a smaller number of extracted genes and the effective processing cost in the ALL/AML data set and a satisfactory fitness performance in the SRBCTs data set. This indicates that unlike the other three systems, the tanh based system is not restricted by the nature of the microarray platform, such as oligonucleotide-based (ALL/AML) and cDNA-based (SRBCTs); the data distribution and the number of classes in the data set.

In the next section, we will assess the performance of each system on the effects of a specific fitness evaluation, ranging from 5000 to 50000 in three different population size ranging from 100 to 300.

5.2 System performance with different sizes in population and fitness evaluation

In this section, we examined two vital components in the GA which could influence the integrity of the results in supporting the third purpose of our experimental study in Section 4.5.1. These GA components are population and fitness evaluation. Three figures, each containing four graphs representing four different experimental data sets, were produced. Each figure represents the performance of individual systems in terms of the number of genes selected with a selection frequency of 50 and above in Figure 5.4, the fitness performance in Figure 5.5 and the processing time in Figure 5.6, in three population sizes: 100, 200 and 300, and in ten fitness evaluations, ranging from 5000 to 50000. The complete list of the extracted genes by each system is presented in Appendix B.

The four types of the systems shown in the figures represent four different activation functions used in the GANN prototype to compute the fitness values for each subset of genes in the population. These systems comprising sigmoid, linear, tanh and threshold based.

5.2.1 The number of significant genes

Observations on the synthetic data set 1 in Figure 5.4a, in all systems, except the linear based system, were unable to detect all the 30 predefined genes in population size 100. The *linear based* system detected all predefined genes in every population size, except the fitness evaluation size 20000 in population size 100. With increased population size, all systems were able to detect all predefined genes in the data set. Tables B.1-B.3 in Appendix B show the complete list of extracted genes in the synthetic data set 1. A similar observation on the increased number of predefined genes were detected in the synthetic data set 2, as is indicated in Figure 5.4b and Tables B.4-B.6 in Appendix B. A low performance by each system in the population size 100 and the performance was improved when the population size was increased.

The variation of different fitness evaluation sizes in the synthetic data sets has showed a significant impact to the number of predefined genes identified by the system. There was a low number of predefined genes found by all systems in the fitness evaluation size 5000 and when the fitness evaluation size was increased, a larger number of predefined genes was found by each system. The *linear based* system has the best performance when the fitness evaluation size exceeds 20000 in every population size, while the *tanh based* system required at least 30000 fitness evaluations to produce consistent results. For sigmoid and threshold based systems,

there is a discrepancy in the minimum fitness evaluation size for each population. In the population sizes 100 and 300, the *sigmoid based* system required 35000 fitness evaluations for consistent results and in the population size 200, a minimal 30000 fitness evaluations is required for producing consistent results. The *threshold based* system, on the other hand, required minimally 25000 fitness evaluations in the population size 100, 35000 fitness evaluations in the population size 200 and 30000 fitness evaluations in the population size 300.

This observation verified that a small population, i.e. population size 100, is not efficient for microarray analysis. This is due to limited space in the population which is not sufficient to accommodate the enormous possibilities of the combined heterogeneity genes in the data.

For the microarray data, there was a significantly increased number of identified genes by each system when the fitness evaluation size was increased in a larger population size in Figure 5.4(c-d). However, with the comparison on the number of genes found by each system in two population sizes: 200 and 300. In population size 300, all systems have a smaller number of genes identified than in population size 200. There are three reasons: (a) the identified genes in population size 200 may be informative in the development of a cancer, but not crucial in cancer formation,as compared to the genes identified in the population size 300; (b) the complex interaction behaviours of the genes in the microarray data produced the enormous possible combinations of heterogeneity genes which might contribute to cancer development; and (c) the system has been over-fitted by over-sized fitness evaluations and populations, albeit, this reason seems very unlikely in our opinion, as there is no sign of over-fitting in each system when similar parameters were applied in the synthetic data.

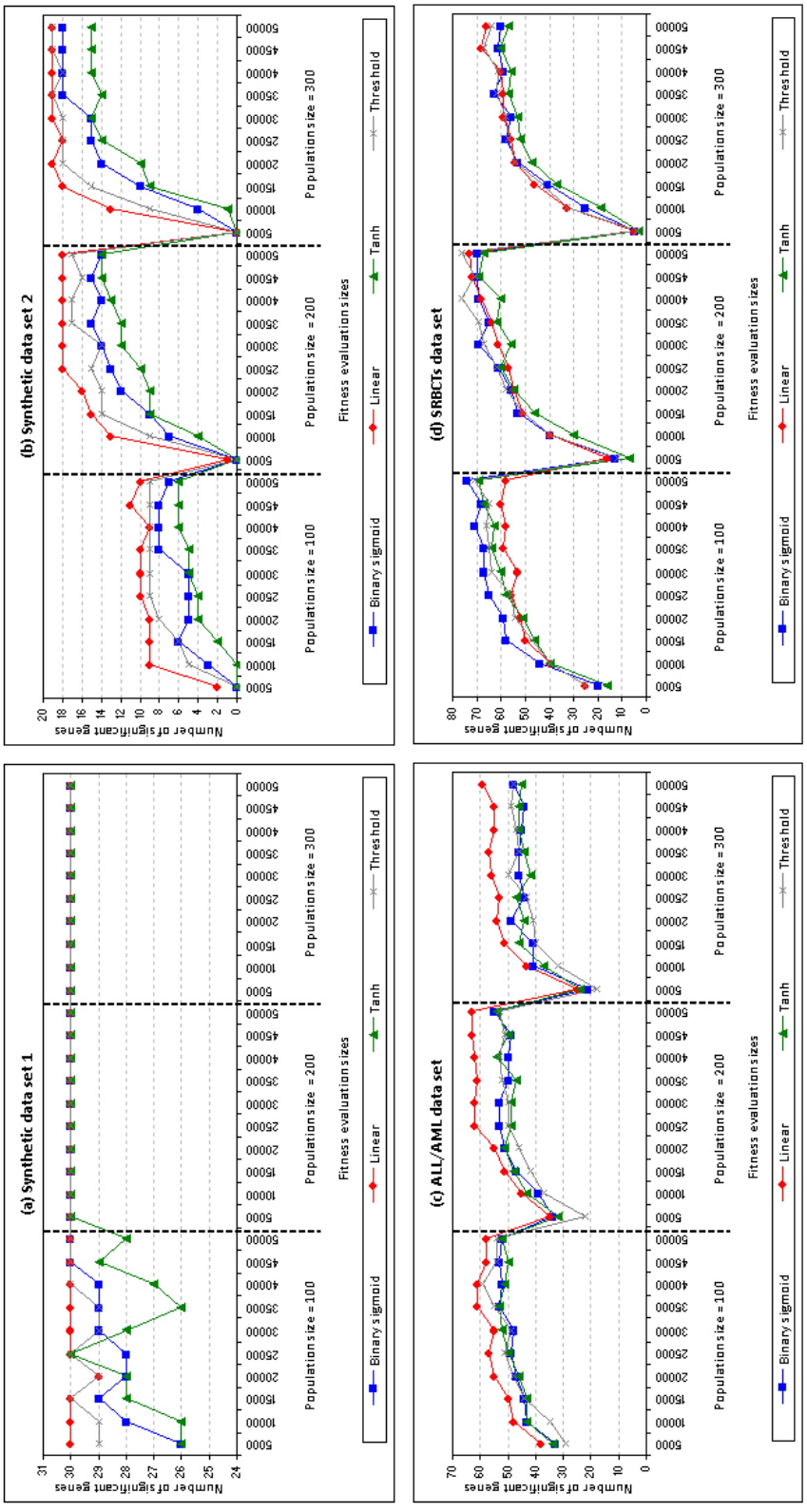


Figure 5.4: The number of significant genes extracted by each system based on the selection frequency of 50 and above in various sizes of population and fitness evaluation. For synthetic data sets, the significantly improved performance of each system on a larger population size and a higher fitness evaluations. For microarray data sets, more genes have been identified in the population size of 200 when compared to the population sizes 100 and 300.

5.2.2 The fitness performance

There is an improvement in the fitness accuracy of the selected genes by each system with the increased fitness evaluations and larger population size, as is indicated in Figure 5.5.

For synthetic data sets, when the population size was increased from 100 to 200, the significantly improved fitness accuracy on the selected genes by each system in every fitness evaluation, as is indicated in Figure 5.5(a-b). The performance of each system had also improved in the synthetic data set 2 when the population size was increased to 300. With the comparison of the fitness performance of each system based on two population sizes: 200 and 300, in the synthetic data set 1, there is no significant performance difference in every fitness evaluation in all systems, except that the threshold based system had a better performance with a fitness evaluation size 5000 in the population size 300 than the other systems.

We would like to draw attention to the fitness performance of each system in the synthetic data set 1. As is shown in Figure 5.4a, with the population sizes 200 and 300, all system have identified all the predefined genes in every fitness evaluation. We observed a discrepancy in the effect of the stronger genes which can severely affect the fitness accuracy of the reported genes. This discrepancy could subsequently affect the decision in the types of therapeutic exercises to be undertaken by patients. All four systems have a lower fitness performance in the population size 200 than in the population size 300, although, all the 30 predefined genes in the data set were detected. Amongst these four systems, the *linear based* system has slightly outperformed the other three systems. This has further confirmed our observation on the linear based system in which it is able to explore the most informative genes than the other systems.

For microarray data sets, as is depicted in Figure 5.5(c-d), in the population size 100, there was a low fitness performance achieved by each system in every fitness evaluation. With the increased population size to 300, the performance of each system was significantly improved. For ALL/AML data set, both the *sigmoid* and the *tanh based* systems have the best fitness performance with minimally 30000 fitness evaluations in the population size 200 and above, while the *linear* and the *threshold based* systems required only 20000 fitness evaluations. For SRBCTs data set, all systems have a better performance in the population size 300 than in the population size 200, with minimally 30000 fitness evaluations. This is due to the multiclass nature of the data set.

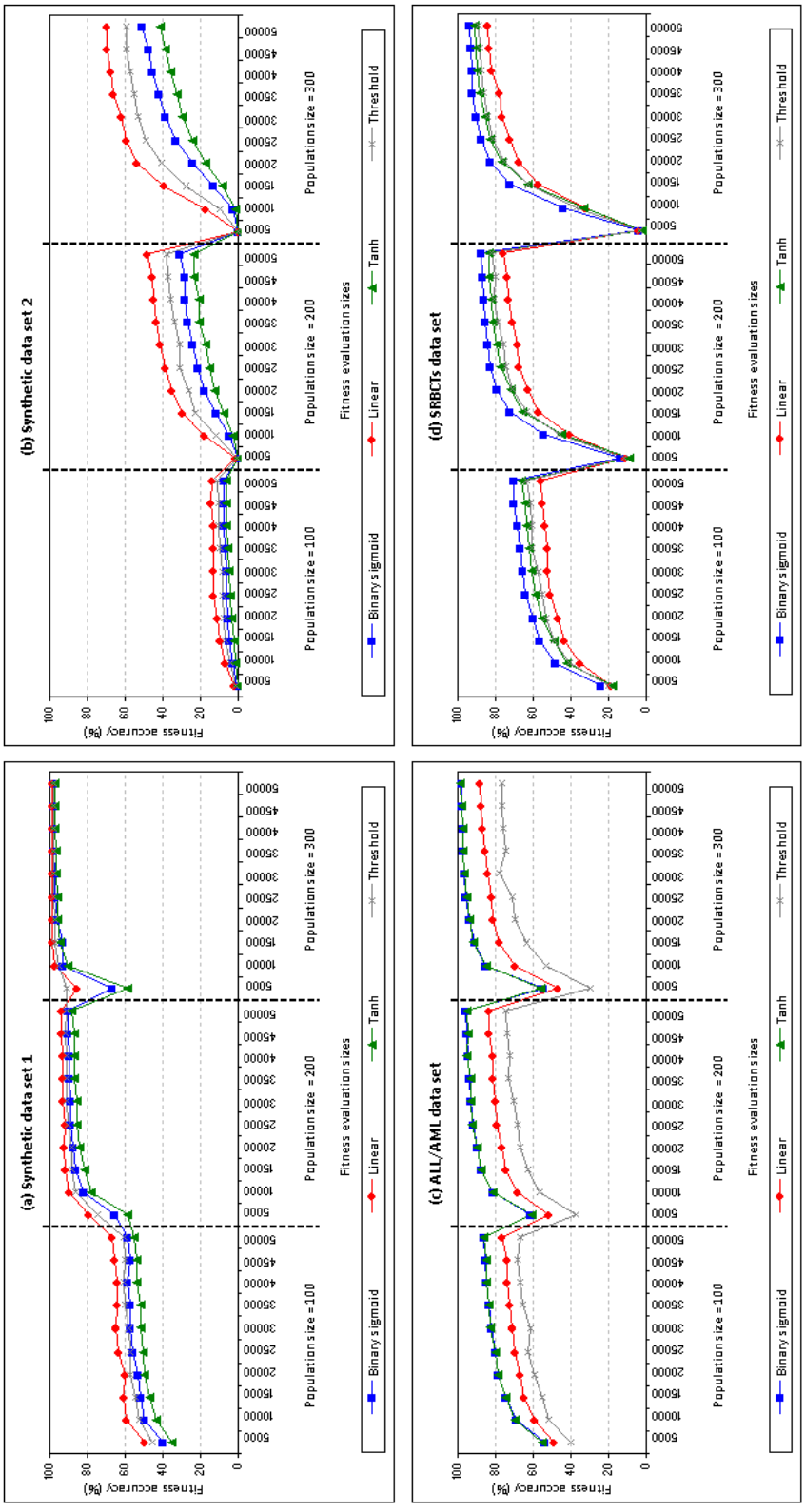


Figure 5.5: The fitness performance by each system in various sizes of population and fitness evaluation. The significantly improved performance by each system with increased sizes in population and fitness evaluation in all data sets. An almost similar fitness performance was achieved when the fitness evaluation size exceeded 10000 in population sizes 200 and 300 for the sigmoid and the tanh based systems in the ALL/AML data set, while the linear and the threshold based systems have a slight improvement in the population size 300 when similar fitness evaluation sizes were applied. For SRBCTs data set, there was a slight improvement in each system when the fitness evaluation exceeding 20000 in population sizes 200 and 300.

5.2.3 The processing time

With observation on the processing time of each system in Figure 5.6, a high ratio of elapsed time by each system was found and this is associated with a high fitness evaluation size rather than the population size. This is indicated by the increased elapsed time in different fitness evaluation sizes within a specific population size and there is no significant processing time difference between similar sets of fitness evaluation in population sizes 200 and 300.

The figure shows that higher processing time in every fitness evaluation in the population size 100 when compared to the identical sets of fitness evaluation were applied in a larger population size. This is because a small population size is not sufficient to accommodate more learning patterns (chromosomes) for the system to model the general rules. The insufficient capacity on the population has significantly reduced the fitness performance of the systems (see Figure 5.5), even though a sufficient number of evaluations is given.

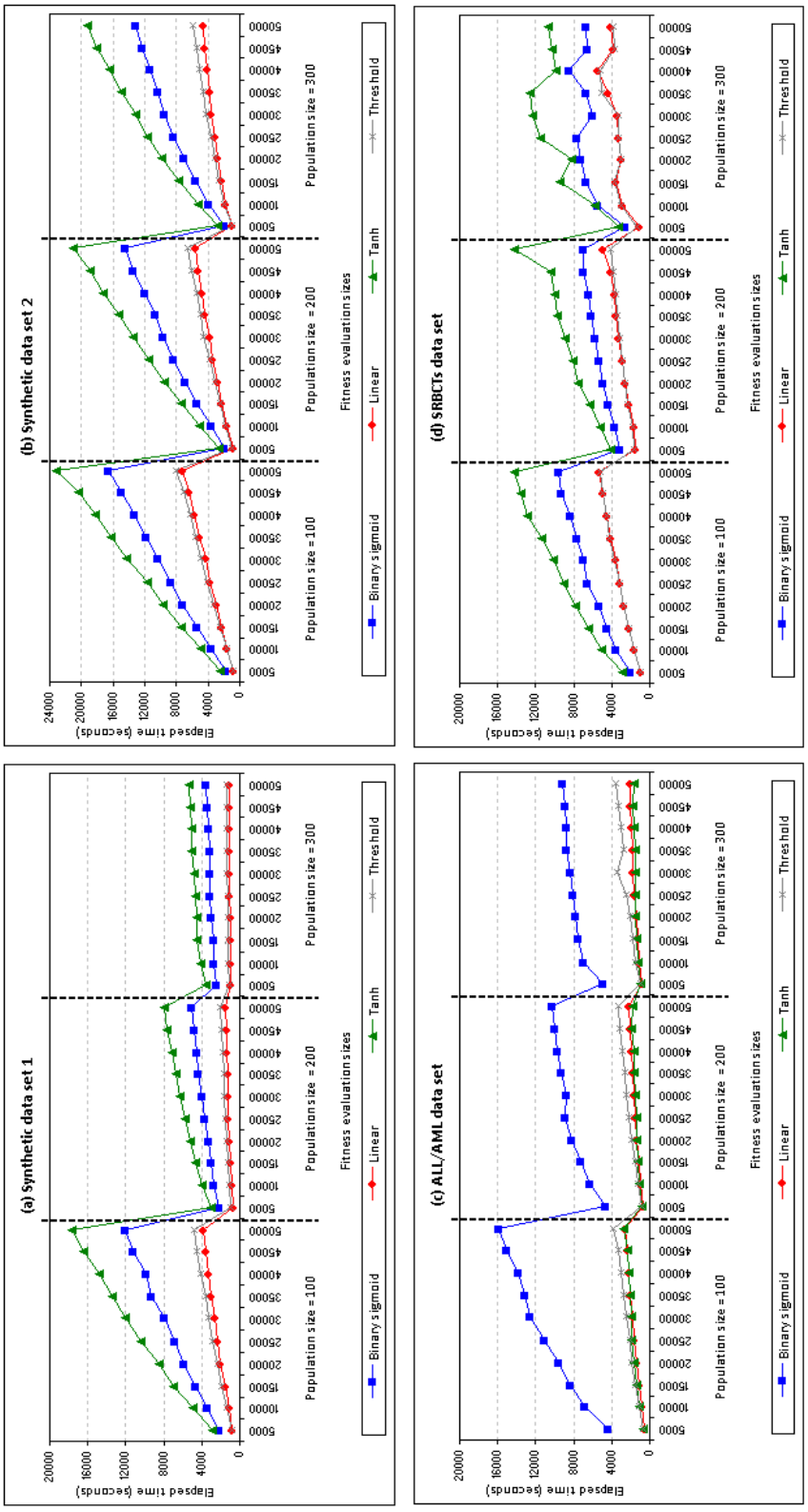


Figure 5.6: The processing time of each system in various sizes of population and fitness evaluation. Both the linear and the threshold based systems have the lowest amount of elapsed time in both synthetic data sets and the SRBCTs data set, while the tanh based system has a significantly decreased amount of processing time in the ALL/AML data set. The sigmoid based system has intensive processing time in the ALL/AML data set when compared to the other systems.

5.2.4 Discussion

The findings reported in this section are based on the performance of each system in the effect of different sizes in population and fitness evaluation . The results indicate that the integrity of the system in finding stronger genes could, indeed, be compromised by inappropriate configuration of the GA population and the fitness evaluation.

Cartwright (2008b) commented on, that the population size is not critically important in the success of a GA,provided that the population size is not unreasonably small (i.e. < 40 chromosomes). Our results had shown the importance of the population in the success of a GA and a strong interaction on the evolution process, as is observed by DeJong and Spears (1991). This is indicated in the population size 100 with the elevated processing time and low fitness accuracy achieved by each system in every fitness evaluation. With the increased population size, better fitness performance and lower processing time are achieved when similar fitness evaluations were applied. DeJong and Spears (1991) made such comments based on the augmentation in the crossover operator and we derived similar conclusions with the increased fitness evaluation sizes.

In addition to the population size, a larger fitness evaluation also promise better fitness confidence in the selected genes and the processing time is not always increased with larger evaluations. This is indicated in Figure 5.6, where a lower or equal ratio of elapsed time was found in each fitness evaluation in population sizes 200 and 300.

Despite the ideal population size, i.e. ranging from 40 to 100, as is suggested by Cartwright (2008b), our results show that the population size 100 is still considerably small for handling microarray data sets. Based on the ALL/AML and SRBCTs data sets, our findings suggest the minimal population size for microarray data should be between 200 and 300. The minimal fitness evaluation size for binary class data should be 20000 and 25000 for multiclass data. The maximal fitness evaluation size should not be exceeds 40000. Our findings also confirmed that the *linear* and the *tanh based* systems are the two most effective ANN activation functions to be used to compute fitness values for GA chromosomes.

5.3 The statistical significance of the extracted genes

In this section, we examine the performance of each system based on the integrity of the extracted genes by each system. This experiment was conducted based on the number of identified genes extracted in each data set and is based on the population size 300 with fitness evaluation size ranging from 20000 to 40000. Four tables based on these data sets were produced and each table presents a list of genes extracted by each system, ordered, according to the selection frequency of the gene, along with its IG value. The IG (gain ratio) method is a type of t-statistics approach that evaluates the significance of a feature by measuring the

gain ratio with respect to the class. This ratio value is used as the comparison on the feature ranking order between the IG method and the GANN systems.

5.3.1 The synthetic data set 1

Table 5.1 on page 133 shows the list of extracted genes in the synthetic data set 1. All four systems have identified all the 30 predefined genes in the data set with an identical set of the strongest genes, i.e. genes 5014 and 12; and the weakest genes, i.e. genes 15, 6, 5011 and 13. Amongst four systems, the *threshold based* system, overall, has a fairly consistent ranking in the selected genes when the fitness evaluation size is 25000 or more (see Table 5.1d). Whilst the other three systems have a significant gene ranking discrepancy in different sizes of fitness evaluations.

Both the *sigmoid* and the *linear based* systems have a significant ranking discrepancy in the fitness evaluation size 40000 and the fitness evaluation size lesser than 40000. For sigmoid based system, as is showed in Table 5.1a, gene 5013 has been highly rated in smaller fitness evaluations, i.e. 20000 to 35000, however, this gene is not frequently selected in the fitness evaluation size 40000. As opposed to gene 5013, gene 5008 has been highly rated in the fitness evaluation size 40000, but not in smaller fitness evaluations. For linear based system in Table 5.1b, gene 5009 has a significant ranking in the fitness evaluation size 40000, when compared to smaller fitness evaluations. The *tanh based* system, in Table 5.1c, however, has a greater ranking discrepancy in the genes in the fitness evaluation size 35000 with the other evaluation sizes. The differentially expressed genes involved genes 8, 5009 and 5008, for which gene 5009 was highly rated in the fitness evaluation size 35000, however, genes 8 and 5008 were less significant.

By comparing the ranking order of the genes extracted by the GANN systems and the IG method, gene 5014 is the most significant gene (i.e. the primary feature for perfectly discriminate data classes) amongst the 30 selected genes by all the GANN systems, followed by genes 12 and gene 13 is the weakest (i.e. the least significant feature for class discrimination) in the ranking. The IG method, however, rated gene 12 as the strongest with the rate of 0.612, followed by gene 5005 with the rate of 0.545 and gene 2 is the lowest ranked in IG. The variability on the gene ranking is due to the IG method ranks genes based on its individual significance to the cancer classes and overlooks its correlation with other genes to the cancer classes. The GANN system, on the other hand, rank the genes based on its correlation with the other genes to the cancer classes, meaning that using the selected genes by the GANN system. It is not sufficient for creating a perfect discrimination between data classes compared to the genes selected by the IG method. However, the genes selected by the GANN system provide more information on the correlation between expressed genes to the cancer classes.

Table 5.1: The list of extracted genes in synthetic data set 1 by each system based on the population size 300. *Freq.* is the number of times that the gene is selected.

(a) Sigmoid-based System

		Fitness Evaluation size									
		20000		25000		30000		35000		40000	
IG rate	Genes	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.534	5014	**1**	*1463*	**1**	*1495*	**1**	*1454*	**1**	*1453*	**1**	*1484*
0.612	12	**2**	*1068*	**2**	*1130*	**2**	*1066*	**2**	*1140*	**2**	*1102*
0.534	10	**3**	*947*	**3**	*922*	**4**	*931*	**5**	*897*	**3**	*992*
0.506	5001	**4**	*903*	**4**	*901*	**3**	*991*	**3**	*919*	**4**	*942*
0.509	11	**5**	*820*	**5**	*841*	**6**	*844*	**4**	*900*	**5**	*885*
0.501	7	**6**	*807*	**6**	*816*	**5**	*866*	**6**	*796*	**6**	*778*
0.509	9	**8**	*701*	**7**	*745*	**7**	*732*	**8**	*751*	**7**	*717*
0.546	5005	**7**	*703*	**8**	*705*	**8**	*700*	**7**	*762*	**8**	*701*
0.423	8	**9**	*653*	**9**	*646*	**10**	*653*	**10**	*638*	**9**	*666*
0.534	4	**12**	*595*	**12**	*580*	**9**	*675*	**9**	*655*	**10**	*634*
0.447	5013	**10**	*606*	**11**	*602*	**11**	*609*	**11**	*624*	**15**	*558*
0.38	5009	**11**	*601*	**13**	*573*	**12**	*590*	**12**	*613*	**13**	*570*
0.515	1	**13**	*588*	**10**	*614*	**15**	*560*	**14**	*560*	**12**	*587*
0.366	5003	**14**	*545*	**14**	*563*	**13**	*588*	**15**	*551*	**16**	*541*
0.341	5008	**16**	*515*	**15**	*557*	**16**	*533*	**13**	*584*	**11**	*597*
0.373	5010	**15**	*527*	**16**	*534*	**14**	*562*	**16**	*551*	**14**	*558*
0.324	3	**17**	*479*	**18**	*474*	**17**	*476*	**17**	*469*	**17**	*489*
0.425	5012	**18**	*463*	**17**	*493*	**18**	*449*	**18**	*457*	**18**	*476*
0.417	14	**20**	*420*	**19**	*440*	**19**	*411*	**19**	*438*	**20**	*412*
0.455	5006	**21**	*416*	**20**	*428*	**20**	*394*	**20**	*430*	**19**	*442*
0.36	5015	**19**	*430*	**21**	*405*	**21**	*394*	**21**	*420*	**21**	*407*
0.289	5004	**22**	*318*	**23**	*321*	**22**	*301*	**22**	*330*	**22**	*324*
0.4	5007	**23**	*305*	**22**	*345*	**24**	*282*	**23**	*286*	**23**	*284*
0.269	2	**24**	*269*	**25**	*263*	**23**	*296*	**24**	*279*	**24**	*270*
0.371	5002	**26**	*238*	**24**	*270*	**25**	*242*	**25**	*256*	**26**	*256*
0.316	5	**25**	*254*	**26**	*253*	**26**	*224*	**26**	*242*	**25**	*264*
0.371	15	**27**	*177*	**27**	*183*	**28**	*164*	**27**	*169*	**27**	*162*
0.352	6	**28**	*167*	**29**	*172*	**27**	*167*	**28**	*147*	**28**	*155*
0.321	5011	**29**	*144*	**28**	*172*	**30**	*133*	**30**	*141*	**30**	*141*
0.345	13	**30**	*139*	**30**	*122*	**29**	*143*	**29**	*142*	**29**	*153*

(b) Linear-based System

		Fitness Evaluation size									
		20000		25000		30000		35000		40000	
IG rate	Genes	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.534	5014	**1**	*1500*	**1**	*1436*	**1**	*1481*	**1**	*1476*	**1**	*1455*
0.612	12	**2**	*1069*	**2**	*1101*	**2**	*1145*	**2**	*1093*	**2**	*1140*
0.534	10	**3**	*1023*	**3**	*996*	**3**	*983*	**3**	*990*	**4**	*918*
0.506	5001	**4**	*926*	**4**	*955*	**4**	*935*	**4**	*926*	**3**	*961*
0.509	11	**5**	*904*	**6**	*818*	**5**	*875*	**5**	*916*	**5**	*902*
0.501	7	**6**	*867*	**5**	*889*	**6**	*825*	**6**	*875*	**6**	*841*
0.509	9	**7**	*770*	**7**	*799*	**7**	*767*	**8**	*731*	**7**	*807*
0.546	5005	**8**	*766*	**8**	*756*	**8**	*762*	**7**	*776*	**8**	*777*
0.534	4	**9**	*684*	**10**	*652*	**9**	*661*	**10**	*677*	**11**	*652*
0.423	8	**10**	*651*	**9**	*661*	**12**	*618*	**9**	*689*	**10**	*659*
0.38	5009	**11**	*628*	**13**	*596*	**10**	*646*	**12**	*621*	**9**	*666*
0.515	1	**12**	*623*	**11**	*644*	**14**	*613*	**14**	*600*	**12**	*644*
0.447	5013	**14**	*585*	**12**	*624*	**13**	*614*	**13**	*601*	**15**	*596*
0.366	5003	**13**	*608*	**14**	*588*	**15**	*597*	**15**	*593*	**13**	*620*
0.341	5008	**15**	*578*	**16**	*561*	**11**	*619*	**11**	*626*	**14**	*616*
0.373	5010	**17**	*508*	**15**	*586*	**18**	*526*	**16**	*593*	**16**	*555*
0.324	3	**16**	*544*	**17**	*540*	**16**	*534*	**17**	*518*	**17**	*500*
0.425	5012	**18**	*494*	**18**	*507*	**17**	*534*	**18**	*514*	**18**	*486*
0.417	14	**21**	*440*	**20**	*461*	**19**	*439*	**19**	*462*	**21**	*421*
0.36	5015	**20**	*442*	**19**	*473*	**21**	*407*	**21**	*402*	**19**	*440*
0.455	5006	**19**	*446*	**21**	*430*	**20**	*436*	**20**	*428*	**20**	*423*
0.289	5004	**22**	*337*	**22**	*360*	**22**	*366*	**22**	*366*	**22**	*392*
0.4	5007	**23**	*330*	**23**	*324*	**23**	*356*	**23**	*346*	**23**	*331*
0.269	2	**24**	*297*	**24**	*279*	**24**	*332*	**25**	*277*	**24**	*313*
0.371	5002	**25**	*292*	**26**	*259*	**25**	*280*	**24**	*292*	**25**	*295*
0.316	5	**26**	*255*	**25**	*268*	**26**	*263*	**26**	*265*	**26**	*267*
0.352	6	**27**	*195*	**27**	*195*	**28**	*186*	**28**	*177*	**28**	*183*
0.371	15	**28**	*181*	**29**	*171*	**27**	*189*	**27**	*190*	**27**	*187*
0.321	5011	**29**	*178*	**28**	*194*	**29**	*172*	**29**	*148*	**29**	*166*
0.345	13	**30**	*143*	**30**	*163*	**30**	*143*	**30**	*123*	**30**	*140*

Continued on Next Page...

Table 5.1 – *Continued*

(c) Tanh-based System											
		Fitness Evaluation size									
		20000		25000		30000		35000		40000	
IG rate	Genes	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.534	5014	**1**	*1483*	**1**	*1400*	**1**	*1486*	**1**	*1513*	**1**	*1455*
0.612	12	**2**	*1075*	**2**	*1069*	**2**	*1111*	**2**	*1108*	**2**	*1017*
0.506	5001	**3**	*930*	**4**	*911*	**4**	*860*	**3**	*933*	**3**	*909*
0.534	10	**4**	*904*	**3**	*912*	**3**	*925*	**5**	*831*	**4**	*885*
0.509	11	**6**	*787*	**5**	*789*	**5**	*837*	**4**	*839*	**6**	*809*
0.501	7	**5**	*794*	**6**	*766*	**6**	*762*	**6**	*798*	**5**	*815*
0.509	9	**8**	*686*	**8**	*702*	**7**	*683*	**7**	*710*	**8**	*720*
0.546	5005	**7**	*695*	**7**	*712*	**8**	*669*	**8**	*663*	**7**	*729*
0.423	8	**10**	*602*	**9**	*644*	**9**	*612*	**11**	*593*	**9**	*630*
0.534	4	**9**	*618*	**10**	*579*	**11**	*569*	**9**	*649*	**10**	*587*
0.447	5013	**11**	*572*	**11**	*570*	**10**	*600*	**12**	*584*	**12**	*565*
0.38	5009	**12**	*570*	**12**	*563*	**13**	*539*	**10**	*598*	**13**	*532*
0.515	1	**13**	*543*	**14**	*544*	**12**	*564*	**13**	*553*	**11**	*566*
0.366	5003	**16**	*496*	**13**	*545*	**16**	*513*	**14**	*550*	**15**	*530*
0.341	5008	**14**	*539*	**15**	*537*	**14**	*519*	**16**	*492*	**14**	*531*
0.373	5010	**15**	*514*	**16**	*500*	**15**	*517*	**15**	*510*	**16**	*511*
0.324	3	**17**	*474*	**18**	*467*	**17**	*474*	**17**	*484*	**17**	*450*
0.425	5012	**18**	*471*	**17**	*470*	**19**	*424*	**18**	*430*	**18**	*440*
0.417	14	**19**	*400*	**21**	*359*	**18**	*454*	**20**	*421*	**19**	*422*
0.455	5006	**20**	*390*	**20**	*405*	**20**	*420*	**19**	*424*	**20**	*409*
0.36	5015	**21**	*383*	**19**	*417*	**21**	*368*	**21**	*341*	**21**	*374*
0.289	5004	**22**	*346*	**22**	*289*	**22**	*321*	**23**	*282*	**22**	*291*
0.4	5007	**23**	*288*	**23**	*287*	**23**	*298*	**22**	*309*	**23**	*281*
0.371	5002	**24**	*267*	**24**	*263*	**24**	*253*	**24**	*263*	**25**	*250*
0.269	2	**25**	*237*	**25**	*254*	**26**	*235*	**25**	*248*	**24**	*273*
0.316	5	**26**	*231*	**26**	*235*	**25**	*239*	**26**	*227*	**26**	*241*
0.371	15	**28**	*153*	**27**	*141*	**28**	*166*	**27**	*142*	**27**	*167*
0.352	6	**27**	*156*	**28**	*135*	**27**	*168*	**28**	*135*	**28**	*154*
0.321	5011	**29**	*134*	**29**	*134*	**29**	*131*	**29**	*132*	**29**	*129*
0.345	13	**30**	*99*	**30**	*98*	**30**	*126*	**30**	*117*	**30**	*113*

(d) Threshold-based System											
		Fitness Evaluation size									
		20000		25000		30000		35000		40000	
IG rate	Genes	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.534	5014	**1**	*1458*	**1**	*1427*	**1**	*1473*	**1**	*1420*	**1**	*1484*
0.612	12	**2**	*1154*	**2**	*1155*	**2**	*1150*	**2**	*1092*	**2**	*1111*
0.534	10	**4**	*938*	**3**	*1005*	**3**	*978*	**3**	*976*	**3**	*959*
0.506	5001	**3**	*974*	**4**	*889*	**4**	*899*	**4**	*925*	**4**	*889*
0.509	11	**5**	*874*	**5**	*814*	**5**	*843*	**5**	*876*	**5**	*856*
0.501	7	**6**	*786*	**6**	*801*	**6**	*822*	**6**	*842*	**6**	*810*
0.509	9	**8**	*717*	**7**	*774*	**7**	*718*	**7**	*785*	**7**	*777*
0.546	5005	**7**	*732*	**8**	*732*	**8**	*698*	**8**	*742*	**8**	*687*
0.534	4	**11**	*612*	**12**	*617*	**9**	*648*	**9**	*651*	**9**	*657*
0.423	8	**9**	*639*	**9**	*650*	**12**	*604*	**10**	*638*	**10**	*653*
0.515	1	**10**	*617*	**11**	*633*	**11**	*611*	**12**	*601*	**11**	*611*
0.38	5009	**12**	*600*	**10**	*641*	**10**	*618*	**13**	*600*	**14**	*592*
0.447	5013	**13**	*592*	**15**	*560*	**13**	*576*	**14**	*588*	**13**	*594*
0.366	5003	**14**	*578*	**14**	*563*	**15**	*574*	**15**	*572*	**12**	*609*
0.341	5008	**15**	*567*	**13**	*575*	**14**	*575*	**11**	*601*	**15**	*555*
0.373	5010	**16**	*522*	**16**	*529*	**16**	*548*	**16**	*535*	**16**	*527*
0.324	3	**18**	*473*	**17**	*498*	**17**	*534*	**17**	*515*	**17**	*496*
0.425	5012	**17**	*516*	**18**	*462*	**18**	*443*	**18**	*467*	**18**	*490*
0.455	5006	**19**	*433*	**20**	*420*	**20**	*434*	**19**	*410*	**20**	*429*
0.417	14	**20**	*398*	**19**	*445*	**19**	*439*	**20**	*399*	**21**	*418*
0.36	5015	**21**	*373*	**21**	*412*	**21**	*376*	**21**	*378*	**19**	*432*
0.289	5004	**22**	*352*	**22**	*327*	**22**	*343*	**22**	*313*	**22**	*330*
0.4	5007	**23**	*315*	**23**	*318*	**23**	*307*	**23**	*307*	**23**	*324*
0.269	2	**25**	*258*	**25**	*284*	**24**	*271*	**24**	*267*	**25**	*296*
0.371	5002	**24**	*277*	**26**	*275*	**25**	*271*	**25**	*247*	**24**	*303*
0.316	5	**26**	*243*	**24**	*285*	**26**	*253*	**26**	*245*	**26**	*270*
0.352	6	**27**	*168*	**27**	*189*	**28**	*172*	**27**	*166*	**27**	*187*
0.371	15	**28**	*164*	**28**	*170*	**27**	*174*	**28**	*162*	**28**	*186*
0.321	5011	**29**	*151*	**29**	*156*	**29**	*165*	**30**	*129*	**29**	*154*
0.345	13	**30**	*139*	**30**	*154*	**30**	*137*	**29**	*141*	**30**	*131*

5.3.2 The synthetic data set 2

For synthetic data set 2 in Table 5.2, the linear based system outperforms the other three system by consistently identifying 19 out of 30 predefined genes in the data set. Meanwhile, the sigmoid, the tanh and the

threshold based systems only identified 15, 14 and 18 predefined genes, respectively.

Table 5.2: The list of extracted genes in synthetic data set 2 by each system based on the population size 300. *Freq.* is the number of times that the gene is selected. Genes highlighted in red are noisy genes.

(a) Sigmoid-based System

		Fitness Evaluation size									
		20000		25000		30000		35000		40000	
IG rate	Genes	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.381	21	**1**	*793*	**1**	*1039*	**1**	*1170*	**1**	*1242*	**2**	*1336*
0.557	11	**2**	*740*	**2**	*971*	**2**	*1134*	**3**	*1183*	**3**	*1295*
0.645	12	**3**	*645*	**3**	*962*	**3**	*1114*	**2**	*1224*	**1**	*1349*
0.385	19	**4**	*617*	**4**	*794*	**4**	*902*	**4**	*929*	**4**	*964*
0.442	27	**5**	*421*	**5**	*569*	**5**	*660*	**5**	*674*	**5**	*758*
0.567	24	**6**	*313*	**6**	*399*	**7**	*387*	**6**	*460*	**6**	*493*
0.609	17	**7**	*306*	**7**	*358*	**6**	*433*	**7**	*437*	**7**	*465*
0.518	16	**8**	*226*	**8**	*354*	**8**	*371*	**8**	*386*	**8**	*438*
0.51	23	**9**	*170*	**9**	*295*	**9**	*359*	**9**	*349*	**9**	*418*
0.371	18	**10**	*91*	**10**	*123*	**10**	*129*	**10**	*153*	**10**	*169*
0.504	26	**14**	*65*	**11**	*119*	**11**	*128*	**12**	*138*	**11**	*148*
0.396	15	**11**	*82*	**13**	*97*	**12**	*122*	**11**	*145*	**12**	*148*
0.265	22	**12**	*75*	**12**	*98*	**13**	*100*	**13**	*113*	**14**	*115*
0.504	20	**13**	*67*	**14**	*83*	**14**	*91*	**14**	*104*	**13**	*125*
0.482	29			**15**	*65*	**15**	*59*	**15**	*67*	**15**	*74*
0.248	667							**16**	*56*	**16**	*54*
0.3	25							**17**	*55*	**17**	*53*
0.272	13							**18**	*55*	**19**	*52*
0.377	14							**19**	*50*	**18**	*52*

(b) Linear-based System

		Fitness Evaluation size									
		20000		25000		30000		35000		40000	
IG rate	Genes	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.645	12	**1**	*1687*	**1**	*1879*	**1**	*1990*	**1**	*2168*	**1**	*2134*
0.381	21	**2**	*1683*	**2**	*1853*	**2**	*1935*	**2**	*1999*	**2**	*2051*
0.557	11	**3**	*1460*	**3**	*1625*	**3**	*1706*	**3**	*1740*	**3**	*1793*
0.385	19	**4**	*1119*	**4**	*1243*	**4**	*1286*	**4**	*1357*	**4**	*1432*
0.442	27	**5**	*1085*	**5**	*1147*	**5**	*1245*	**5**	*1315*	**5**	*1338*
0.518	16	**6**	*803*	**6**	*842*	**6**	*878*	**6**	*909*	**6**	*934*
0.609	17	**7**	*678*	**7**	*755*	**8**	*747*	**8**	*752*	**7**	*839*
0.567	24	**8**	*649*	**8**	*751*	**7**	*748*	**7**	*756*	**8**	*756*
0.51	23	**9**	*576*	**9**	*622*	**9**	*659*	**9**	*707*	**9**	*698*
0.371	18	**10**	*224*	**10**	*227*	**10**	*277*	**10**	*282*	**10**	*250*
0.396	15	**11**	*213*	**13**	*213*	**12**	*208*	**11**	*247*	**12**	*241*
0.504	26	**13**	*188*	**11**	*227*	**13**	*192*	**12**	*243*	**11**	*245*
0.265	22	**12**	*190*	**12**	*221*	**11**	*209*	**13**	*220*	**13**	*195*
0.504	20	**14**	*141*	**14**	*164*	**14**	*181*	**14**	*183*	**14**	*180*
0.482	29	**15**	*130*	**15**	*153*	**15**	*122*	**15**	*146*	**15**	*170*
0.377	14	**17**	*98*	**17**	*89*	**16**	*110*	**16**	*122*	**16**	*115*
0.272	13	**16**	*105*	**16**	*109*	**17**	*95*	**18**	*87*	**18**	*87*
0.3	25	**18**	*75*	**18**	*80*	**18**	*83*	**17**	*99*	**17**	*92*
0.248	667	**21**	*51*	**19**	*68*	**19**	*76*	**20**	*65*	**19**	*71*
0.244	4883	**19**	*57*	**20**	*58*	**21**	*55*	**19**	*69*	**21**	*63*
-	4377	**20**	*52*	**22**	*57*	**20**	*61*	**24**	*59*	**24**	*57*
0.275	28	**22**	*50*			**22**	*51*	**21**	*62*	**23**	*59*
-	2828			**23**	*50*			**22**	*59*	**20**	*63*
-	4175			**21**	*58*					**22**	*60*
-	2816							**23**	*59*		
-	4390									**25**	*54*
-	2471									**26**	*52*

Continued on Next Page...

Table 5.2 – *Continued*

(c) Tanh-based System

		Fitness Evaluation size									
		20000		25000		30000		35000		40000	
IG rate	Genes	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.381	21	**1**	*570*	**1**	*763*	**1**	*872*	**1**	981	**2**	*1064*
0.645	12	**4**	*439*	**2**	*704*	**3**	*833*	**2**	950	**1**	*1072*
0.557	11	**2**	*522*	**3**	*698*	**2**	*843*	**3**	925	**3**	*1004*
0.385	19	**3**	*446*	**4**	*559*	**4**	*651*	**4**	672	**4**	*732*
0.442	27	**5**	*292*	**5**	*365*	**5**	*476*	**5**	505	**5**	*546*
0.567	24	**6**	*177*	**6**	*254*	**6**	*294*	**6**	330	**6**	*352*
0.518	16	**8**	*162*	**7**	*237*	**8**	*272*	**8**	285	**7**	*328*
0.609	17	**7**	*163*	**8**	*229*	**7**	*281*	**7**	300	**8**	*301*
0.51	23	**9**	*117*	**9**	*184*	**9**	*240*	**9**	227	**9**	*265*
0.371	18			**12**	*70*	**12**	*92*	**12**	91	**10**	*126*
0.504	26			**10**	*74*	**11**	*92*	**13**	89	**11**	*113*
0.396	15			**11**	*71*	**10**	*97*	**11**	99	**13**	*93*
0.265	22	**10**	*51*	**14**	*54*	**13**	*77*	**14**	64	**12**	*97*
0.504	20			**13**	*55*	**14**	*75*	**10**	99	**14**	*73*
0.482	29					**15**	*55*			**15**	*50*

(d) Threshold-based System

		Fitness Evaluation size									
		20000		25000		30000		35000		40000	
IG rate	Genes	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.645	12	**2**	*1224*	**2**	*1465*	**1**	*1601*	**1**	*1750*	**1**	*1766*
0.381	21	**1**	*1264*	**1**	*1516*	**2**	*1575*	**2**	*1652*	**2**	*1695*
0.557	11	**3**	*1158*	**3**	*1360*	**3**	*1411*	**3**	*1470*	**3**	*1533*
0.385	19	**4**	*922*	**4**	*999*	**4**	*1107*	**4**	*1073*	**4**	*1106*
0.442	27	**5**	*759*	**5**	*930*	**5**	*985*	**5**	*1006*	**5**	*1013*
0.518	16	**6**	*514*	**6**	*649*	**6**	*648*	**6**	*733*	**6**	*699*
0.609	17	**7**	*476*	**7**	*574*	**7**	*626*	**7**	*623*	**7**	*624*
0.567	24	**8**	*460*	**8**	*497*	**8**	*563*	**8**	*604*	**8**	*614*
0.51	23	**9**	*437*	**9**	*480*	**9**	*513*	**9**	*520*	**9**	*554*
0.371	18	**13**	*136*	**10**	*198*	**10**	*218*	**10**	*217*	**10**	*225*
0.396	15	**11**	*148*	**12**	*149*	**11**	*205*	**11**	*198*	**12**	*204*
0.504	26	**10**	*162*	**11**	*156*	**12**	*191*	**12**	*173*	**11**	*204*
0.265	22	**12**	*139*	**13**	*143*	**13**	*146*	**13**	*173*	**13**	*178*
0.504	20	**14**	*108*	**14**	*114*	**14**	*138*	**14**	*149*	**14**	*139*
0.482	29	**15**	*85*	**15**	*99*	**15**	*106*	**15**	*107*	**15**	*135*
0.272	13	**18**	*57*	**16**	*77*	**17**	*81*	**16**	*92*	**16**	*84*
0.3	25	**16**	*67*	**17**	*66*	**18**	*79*	**18**	*72*	**17**	*82*
0.377	14	**17**	*58*	**18**	*54*	**16**	*87*	**17**	*87*	**18**	*78*
-	4377			**19**	*54*			**19**	*66*		
0.267	30							**20**	*64*		
0.248	667							**21**	*58*		
0.275	28							**22**	*55*		
-	2816									**19**	*54*
-	4175									**20**	*51*

With increased fitness evaluation size, there was a significant improvement in the detection of the number of predefined genes by each system. The *sigmoid based* system identified 18 predefined genes in the fitness evaluation sizes 35000 and 40000, the *linear based* system detected 19 predefined genes in a fitness evaluation size of 30000 or more, the *tanh based* system found 15 identical predefined genes in the fitness evaluation sizes 30000 and 40000, and the *threshold based* system had the highest number of predefined genes found, i.e. 20 genes, in the population size 35000. This would suggested that both the linear and the tanh based systems required a minimal 30000 fitness evaluations for consistent and better selection performance, while the sigmoid and the threshold based systems required at least 35000 fitness evaluations.

The findings also show that none of the systems was able to detect all the 30 predefined genes in synthetic data set 2. This is because two different mean μ values, i.e. 20 stronger genes and 10 weaker genes, were used in creating the data set. All systems are able to detect the stronger genes, but not the weaker genes as they were 'buried' by the other stronger noisy genes, i.e. genes that were not suppose to be selected. Instead, the

presence of the noisy genes was detected in the sigmoid, the linear and the threshold based systems when the fitness evaluation size was increased. These noisy genes are genes 667, 4883, 4377, 2828, 4175, 2816, 4390 and 2471. Gene 667 was found in the sigmoid based system when the fitness evaluation size was increased to 35000 or more (see Table 5.1a), genes 4377, 667, 2816 and 4175 were detected in the threshold based system when higher fitness evaluation sizes were applied (see Table 5.1d) and genes 667, 4883 and 4377 appeared in every fitness evaluation in the linear based system. This would suggested that the linear based system can explore the most significant genes, even with smaller fitness evaluations, however, it cannot promise on whether the selected genes are truly important to the subject of interest. The tanh based system is able to rule out any unrelated genes from those that are highly related, however, it might also dismissed some important genes.

By comparing the ranking order of the genes extracted by the GANN systems and the IG method, genes 21 and 12 are the most significant amongst the 30 selected genes all the GANN systems and for the IG method, genes 12 (IG rate = 0.645) and 17 (IG rate = 0.248) are ranked as the most significant genes. Amongst the predefined genes, the gene 22 (IG rate = 0.265) is the lowest ranked by the IG method. Meanwhile, for all four GANN systems, gene 22 is more significant in terms of its correlation with other genes, compared to gene 28 which is ranked lowly by the linear and the threshold based systems. For the sigmoid/tanh based systems, gene 28 has not been identified as it's correlated gene (i.e. gene 4377) has not been detected by the systems. Amongst the noisy genes detected by the linear based system, genes 667 and 4883 provide a certain level of statistical significance in the cancer classification based on the IG method and the other noisy genes (4377, 2828, 4175, 2816, 4390 and 2471) pose no significance contribution to the cancer classification. This means that the detection of these noisy genes is due to the detection of their correlated genes by the linear based system. Gene 667 may correlates to the predefined genes 25, 13 and 14 based on the comparison between the extracted genes by the sigmoid and the tanh based systems. The detection of gene 4377 is due to the presence of gene 667 in the linear/threshold based systems.

5.3.3 The ALL/AML microarray data set

With reference on Table 5.3 on page 138 for ALL/AML data set, not surprisingly, the *linear based* system has the highest number of extracted genes, i.e. 63 genes in total, and the *tanh based* system has the lowest number of extracted genes, i.e. 53 genes in total. Both the *sigmoid* and the *threshold based* systems have, in total, 54 genes extracted by the fitness evaluation size ranging from 20000 to 40000. Amongst the genes extracted in each system, across the board, 39 genes overlapped in all systems, including the first-17 genes selected by each system.

Table 5.3: The list of extracted genes in ALL/AML data set by each system based on the population size 300. *Freq.* is the number of times that the gene is selected. Genes highlighted in **Boldface** are common genes that were identified by all systems. Genes marked with "*" symbol are genes that matched with the genes reported in the original study.

(a) Sigmoid-based System

			Fitness Evaluation Size									
	Gene	Accession	20000		25000		30000		35000		40000	
IG Rate	Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.742	**1882***	M27891	**1**	*1621*	**1**	*1663*	**1**	*1721*	**1**	*1592*	**1**	*1666*
0.747	**2288***	M84526	**2**	*1167*	**2**	*1180*	**2**	*1268*	**2**	*1273*	**2**	*1221*
0.735	**4847***	X95735	**3**	*959*	**3**	*967*	**3**	*1035*	**3**	*964*	**3**	*1001*
0.568	**2354***	M92287	**4**	*916*	**4**	*902*	**4**	*903*	**4**	*953*	**4**	*930*
0.628	**1685**	M11722	**5**	*664*	**5**	*690*	**5**	*679*	**5**	*678*	**5**	*674*
0.429	**804**	HG1612-HT1612	**6**	*632*	**6**	*656*	**6**	*673*	**6**	*632*	**6**	*653*
0.488	**2642***	U05259	**7**	*511*	**7**	*501*	**7**	*518*	**7**	*534*	**7**	*547*
0.376	**1779**	M19507	**8**	*391*	**8**	*427*	**8**	*390*	**8**	*467*	**8**	*416*
0.674	**6041**	L09209	**9**	*351*	**9**	*366*	**9**	*328*	**9**	*364*	**9**	*368*
0.512	**4328***	X59417	**10**	*293*	**10**	*295*	**10**	*306*	**10**	*307*	**10**	*328*
0.475	**2121***	M63138	**11**	*245*	**11**	*248*	**11**	*249*	**11**	*250*	**11**	*251*
0.275	**4211**	X51521	**12**	*218*	**12**	*243*	**12**	*220*	**12**	*232*	**12**	*214*
0.306	**1962**	M33680	**13**	*216*	**13**	*210*	**13**	*217*	**16**	*189*	**14**	*206*
0.747	**760**	D88422	**15**	*198*	**16**	*191*	**16**	*178*	**13**	*225*	**13**	*208*
0.562	**2402***	M96326	**14**	*206*	**15**	*205*	**15**	*183*	**14**	*211*	**17**	*194*
0.388	**5772***	U22376	**16**	*187*	**17**	*191*	**14**	*199*	**15**	*203*	**15**	*202*
0.678	**6855***	M31523	**17**	*152*	**14**	*206*	**17**	*176*	**17**	*176*	**16**	*200*
0.718	**3252**	U46499	**18**	*149*	**18**	*155*	**19**	*146*	**20**	*143*	**19**	*155*
0.512	**758**	D88270	**20**	*130*	**21**	*128*	**21**	*137*	**18**	*173*	**20**	*153*
0.403	**5501***	Z15115	**22**	*113*	**20**	*135*	**20**	*142*	**19**	*170*	**18**	*160*
0.547	**4377**	X62654	**19**	*140*	**19**	*136*	**18**	*148*	**21**	*135*	**21**	*149*
0.643	**6376***	M83652	**21**	*116*	**23**	*109*	**22**	*115*	**22**	*121*	**32**	*85*
0.261	**6049**	U89922	**23**	*108*	**24**	*99*	**27**	*88*	**23**	*115*	**24**	*108*
0.341	**1239**	L07633	**28**	*91*	**22**	*119*	**23**	*105*	**27**	*94*	**25**	*104*
0.543	**1829**	M22960	**26**	*97*	**29**	*85*	**24**	*98*	**24**	*105*	**23**	*111*
0.601	**4373**	X62320	**27**	*93*	**28**	*91*	**26**	*91*	**26**	*97*	**22**	*117*
0.429	**1928***	M31303	**29**	*90*	**25**	*95*	**25**	*93*	**25**	*103*	**31**	*86*
0.424	4680	X82240	**24**	*100*	**27**	*92*	**29**	*80*	**29**	*88*	**26**	*99*
0.403	**6200***	M28130	**25**	*97*	**26**	*94*	**28**	*84*	**28**	*92*	**33**	*83*
0.735	**1834***	M23197	**32**	*76*	**32**	*77*	**36**	*64*	**31**	*81*	**28**	*90*
0.403	**6201***	Y00787	**30**	*84*	**31**	*77*	**44**	*53*	**30**	*85*	**29**	*87*
0.493	4229	X52056	**31**	*79*	**30**	*80*	**30**	*72*	**33**	*68*	**30**	*86*
0.201	4050	X03934	**33**	*73*	**33**	*71*	**34**	*66*	**32**	*77*	**27**	*91*
-	**1796**	M20902	**34**	*70*	**41**	*56*	**32**	*71*	**36**	*64*	**39**	*60*
0.26	**668**	D86967	**39**	*57*	**34**	*64*	**31**	*72*	**38**	*62*	**38**	*64*
0.209	**412**	D42043	**41**	*56*	**35**	*61*	**35**	*65*	**42**	*54*	**35**	*67*
0.392	1704	M13792	**48**	*50*	**36**	*60*	**37**	*59*	**34**	*67*	**37**	*64*
0.309	6702	X97267	**38**	*59*	**38**	*58*	**33**	*68*	**40**	*59*	**41**	*55*
0.266	6271	M33493	**40**	*56*	**42**	*53*	**38**	*59*	**45**	*51*	**34**	*70*
0.395	1630	L47738	**37**	*61*	**43**	*52*	**42**	*57*	**35**	*65*	**44**	*51*
0.428	**1745***	M16038	**36**	*61*			**39**	*59*	**41**	*58*	**36**	*66*
0.513	1144	J05243	**35**	*67*	**40**	*57*			**39**	*61*	**43**	*52*
-	**1975**	M34344			**37**	*59*	**43**	*57*	**37**	*63*		
0.227	6510	U23852	**43**	*54*			**41**	*57*	**43**	*54*		
0.353	**4951**	Y07604	**42**	*55*					**44**	*51*	**42**	*53*
-	1809	M21624	**49**	*50*			**40**	*58*				
-	5445	X04526	**45**	*53*			**45**	*53*				
-	5543	D00749	**46**	*53*					**46**	*51*		
0.332	4438	X66401			**39**	*58*						
-	1941	M31994									**40**	*57*
0.453	**4196***	X17042	**44**	*53*								
0.465	2335	M89957			**44**	*51*						
-	**5950**	M29610	**47**	*51*								
-	**6079**	U59632									**45**	*51*

Continued on Next Page...

Table 5.3 – *Continued*

(b) Linear-based System															
			Fitness Evaluation Size												
	Gene	Accession	20000		25000		30000		35000		40000				
IG Rate	Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.			
0.735	**4847***	X95735	**1**	*1515*	**1**	*1489*	**1**	*1508*	**1**	*1545*	**1**	*1486*			
0.747	**2288***	M84526	**2**	*781*	**2**	*793*	**2**	*792*	**2**	*898*	**2**	*875*			
0.568	**2354***	M92287	**3**	*775*	**3**	*766*	**3**	*732*	**3**	*765*	**3**	*791*			
0.475	**2121***	M63138	**4**	*622*	**4**	*658*	**4**	*629*	**4**	*648*	**5**	*718*			
0.376	**1779**	M19507	**5**	*610*	**5**	*603*	**5**	*612*	**5**	*590*	**4**	*719*			
0.429	**804**	HG1612-HT1612	**6**	*538*	**6**	*497*	**6**	*525*	**6**	*558*	**6**	*549*			
0.742	**1882***	M27891	**7**	*420*	**7**	*463*	**7**	*446*	**7**	*429*	**7**	*476*			
0.674	**6041**	L09209	**8**	*355*	**10**	*340*	**8**	*420*	**8**	*380*	**10**	*366*			
0.488	**2642***	U05259	**9**	*348*	**8**	*368*	**10**	*329*	**9**	*356*	**8**	*372*			
0.512	**4328***	X59417	**10**	*309*	**9**	*360*	**9**	*369*	**10**	*343*	**9**	*366*			
0.306	**1962**	M33680	**11**	*247*	**11**	*238*	**11**	*270*	**12**	*247*	**12**	*264*			
0.275	**4211**	X51521	**12**	*228*	**12**	*233*	**12**	*255*	**11**	*281*	**11**	*265*			
0.353	**4951**	Y07604	**14**	*202*	**13**	*212*	**14**	*200*	**13**	*208*	**14**	*196*			
0.628	**1685**	M11722	**13**	*209*	**17**	*177*	**17**	*176*	**15**	*198*	**13**	*219*			
0.678	**6855***	M31523	**15**	*191*	**16**	*183*	**15**	*197*	**14**	*203*	**18**	*177*			
0.453	**4196***	X17042	**17**	*166*	**15**	*188*	**13**	*213*	**16**	*194*	**17**	*180*			
0.718	**3252**	U46499	**18**	*166*	**14**	*196*	**16**	*189*	**17**	*177*	**15**	*194*			
0.547	**4377**	X62654	**16**	*182*	**19**	*145*	**20**	*149*	**20**	*162*	**16**	*183*			
0.341	**1239**	L07633	**19**	*160*	**18**	*151*	**18**	*172*	**19**	*167*	**19**	*161*			
0.543	**1829**	M22960	**20**	*140*	**20**	*141*	**19**	*160*	**18**	*167*	**21**	*144*			
0.403	**6200***	M28130	**23**	*105*	**22**	*127*	**21**	*135*	**22**	*126*	**20**	*147*			
0.403	**5501***	Z15115	**22**	*111*	**21**	*130*	**22**	*130*	**21**	*133*	**23**	*127*			
0.388	**5772***	U22376	**25**	*101*	**23**	*115*	**26**	*108*	**25**	*112*	**22**	*128*			
0.572	3320*	U50136	**21**	*112*	**28**	*93*	**23**	*121*	**27**	*101*	**28**	*113*			
0.462	6215	M19508	**26**	*96*	**26**	*97*	**25**	*109*	**23**	*117*	**29**	*105*			
-	**1796**	M20902	**28**	*86*	**24**	*110*	**28**	*98*	**28**	*101*	**24**	*127*			
0.462	2111*	M62762	**24**	*105*	**27**	*95*	**29**	*90*	**26**	*104*	**25**	*124*			
0.209	**412**	D42043	**27**	*95*	**25**	*103*	**27**	*102*	**29**	*101*	**26**	*115*			
0.429	**1928***	M31303	**29**	*82*	**29**	*87*	**24**	*114*	**24**	*115*	**31**	*96*			
-	5952	U05255	**30**	*79*	**30**	*80*	**32**	*83*	**31**	*95*	**27**	*114*			
0.562	**2402***	M96326	**34**	*73*	**35**	*69*	**31**	*84*	**37**	*75*	**30**	*104*			
0.493	2020*	M55150	**41**	*65*	**31**	*78*	**30**	*88*	**33**	*81*	**34**	*82*			
0.403	**6201***	Y00787	**39**	*69*	**38**	*66*	**36**	*77*	**30**	*96*	**36**	*79*			
-	**1975**	M34344	**31**	*77*	**40**	*62*	**35**	*80*	**32**	*84*	**32**	*83*			
0.512	**758**	D88270	**37**	*70*	**33**	*73*	**40**	*72*	**34**	*80*	**33**	*83*			
0.643	**6376***	M83652	**35**	*73*	**50**	*55*	**34**	*82*	**35**	*78*	**35**	*81*			
0.601	**4373**	X62320	**33**	*74*	**36**	*69*	**39**	*72*	**38**	*74*	**37**	*79*			
0.747	**760**	D88422	**32**	*75*	**37**	*67*	**37**	*74*	**41**	*72*	**39**	*74*			
0.26	**668**	D86967	**38**	*70*	**39**	*64*	**41**	*71*	**40**	*74*	**40**	*73*			
0.392	6539*	X85116	**43**	*60*	**32**	*75*	**33**	*82*	**43**	*62*	**41**	*71*			
-	**5950**	M29610	**40**	*68*	**51**	*53*	**38**	*74*	**36**	*78*	**46**	*63*			
0.735	**1834***	M23197	**36**	*72*	**41**	*61*	**50**	*53*	**39**	*74*	**44**	*66*			
0.23	6184	M26708	**44**	*60*	**34**	*71*	**48**	*54*	**44**	*59*	**49**	*56*			
-	5445	X04526	**47**	*58*	**43**	*60*	**55**	*51*	**56**	*51*	**55**	*51*			
0.201	4050	X03934	**51**	*54*	**45**	*58*	**52**	*52*	**54**	*53*	**51**	*53*			
0.443	3258*	U46751			**52**	*52*	**43**	*62*	**47**	*55*	**38**	*78*			
0.309	6702	X97267	**48**	*57*			**44**	*60*	**48**	*55*	**45**	*63*			
-	6796	J02982	**50**	*57*	**47**	*56*			**55**	*52*	**50**	*54*			
-	4409	X64594			**44**	*59*	**47**	*55*	**52**	*54*	**54**	*51*			
-	1941	M31994	**54**	*50*	**53**	*50*	**45**	*60*	**49**	*55*					
0.428	**1745***	M16038	**45**	*59*			**42**	*63*			**42**	*68*			
-	**6079**	U59632			**42**	*60*	**56**	*50*	**42**	*67*					
0.493	4229	X52056					**46**	*57*	**45**	*57*	**47**	*59*			
0.256	2408	M96803	**46**	*58*	**46**	*58*	**51**	*53*							
0.332	4438	X66401	**53**	*52*					**50**	*55*	**48**	*58*			
0.513	1144	J05243	**49**	*57*			**54**	*51*	**57**	*50*					
0.465	2335	M89957					**49**	*53*	**53**	*53*	**52**	*52*			
0.257	4291	X56468	**42**	*63*	**49**	*55*									
0.261	**6049**	U89922			**48**	*55*	**53**	*51*							
0.387	6225	M84371	**52**	*52*							**53**	*52*			
0.395	1674	M11147									**43**	*67*			
0.36	7119*	U29175							**46**	*56*					
0.314	3984	U94855							**51**	*54*					

Continued on Next Page...

Table 5.3 – *Continued*

(c) Tanh-based System											
			Fitness Evaluation Size								
	Gene	Accession	20000		25000		30000		35000		40000	
IG Rate	Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.742	**1882***	M27891	**1**	*1599*	**1**	*1644*	**1**	*1620*	**1**	*1665*	**1**	*1662*
0.747	**2288***	M84526	**2**	*1203*	**2**	*1164*	**2**	*1254*	**2**	*1229*	**2**	*1291*
0.735	**4847***	X95735	**3**	*973*	**3**	*956*	**3**	*982*	**3**	*1003*	**4**	*922*
0.568	**2354***	M92287	**4**	*946*	**4**	*936*	**4**	*969*	**4**	*940*	**3**	*934*
0.628	**1685**	M11722	**5**	*655*	**5**	*712*	**5**	*691*	**5**	*709*	**5**	*695*
0.429	**804**	HG1612-HT1612	**6**	*647*	**6**	*653*	**6**	*629*	**6**	*659*	**6**	*643*
0.488	**2642***	U05259	**7**	*520*	**7**	*512*	**7**	*513*	**7**	*530*	**7**	*500*
0.376	**1779**	M19507	**8**	*414*	**8**	*447*	**8**	*406*	**8**	*455*	**8**	*440*
0.674	**6041**	L09209	**9**	*351*	**9**	*363*	**9**	*366*	**9**	*378*	**10**	*319*
0.512	**4328***	X59417	**11**	*263*	**10**	*309*	**10**	*305*	**10**	*308*	**9**	*331*
0.475	**2121***	M63138	**10**	*267*	**11**	*249*	**11**	*236*	**11**	*249*	**11**	*263*
0.306	**1962**	M33680	**13**	*211*	**14**	*203*	**13**	*224*	**13**	*214*	**12**	*237*
0.562	**2402***	M96326	**14**	*193*	**12**	*223*	**12**	*225*	**12**	*218*	**13**	*221*
0.275	**4211**	X51521	**12**	*233*	**16**	*194*	**14**	*217*	**14**	*214*	**15**	*198*
0.747	**760**	D88422	**16**	*170*	**17**	*193*	**15**	*206*	**17**	*190*	**14**	*213*
0.388	**5772***	U22376	**15**	*181*	**13**	*205*	**16**	*181*	**15**	*199*	**16**	*188*
0.678	**6855***	M31523	**18**	*154*	**15**	*201*	**17**	*169*	**16**	*193*	**17**	*172*
0.403	**5501***	Z15115	**17**	*161*	**21**	*124*	**18**	*162*	**19**	*152*	**19**	*144*
0.547	**4377**	X62654	**20**	*136*	**19**	*150*	**19**	*150*	**21**	*137*	**20**	*143*
0.718	**3252**	U46499	**19**	*137*	**20**	*142*	**21**	*128*	**18**	*152*	**18**	*146*
0.512	**758**	D88270	**21**	*123*	**18**	*157*	**20**	*137*	**20**	*138*	**21**	*129*
0.643	**6376***	M83652	**24**	*102*	**26**	*92*	**22**	*115*	**24**	*94*	**22**	*105*
0.341	**1239**	L07633	**26**	*92*	**22**	*108*	**25**	*95*	**22**	*109*	**27**	*92*
0.424	4680	X82240	**23**	*103*	**23**	*103*	**26**	*94*	**25**	*94*	**24**	*98*
0.261	**6049**	U89922	**22**	*109*	**25**	*92*	**24**	*101*	**28**	*87*	**29**	*89*
0.403	**6200***	M28130	**28**	*86*	**32**	*79*	**23**	*106*	**26**	*91*	**23**	*104*
0.601	**4373**	X62320	**25**	*95*	**24**	*92*	**28**	*90*	**29**	*86*	**25**	*96*
0.201	4050	X03934	**29**	*81*	**28**	*88*	**30**	*78*	**23**	*99*	**30**	*87*
0.543	**1829**	M22960	**30**	*77*	**33**	*79*	**27**	*91*	**27**	*88*	**28**	*91*
0.429	**1928***	M31303	**27**	*87*	**29**	*87*	**29**	*86*	**31**	*82*	**32**	*79*
0.493	4229	X52056	**33**	*70*	**27**	*90*	**31**	*77*	**30**	*86*	**26**	*92*
-	**1796**	M20902	**32**	*71*	**34**	*65*	**34**	*74*	**34**	*72*	**31**	*83*
0.403	**6201***	Y00787	**31**	*76*	**30**	*83*	**35**	*71*	**38**	*56*	**33**	*74*
0.735	**1834***	M23197	**35**	*70*	**31**	*80*	**33**	*74*	**33**	*76*	**41**	*55*
0.392	1704	M13792	**37**	*63*	**37**	*61*	**36**	*65*	**35**	*69*	**35**	*67*
0.26	**668**	D86967	**34**	*70*			**32**	*76*	**32**	*79*	**34**	*74*
0.209	**412**	D42043	**38**	*62*	**41**	*56*	**40**	*53*	**39**	*56*	**38**	*63*
0.395	1630	L47738	**41**	*56*	**40**	*56*	**37**	*65*	**37**	*58*	**42**	*53*
0.309	6702	X97267	**42**	*55*	**36**	*63*			**36**	*65*	**37**	*64*
0.266	6271	M33493	**43**	*53*	**35**	*63*	**38**	*63*			**39**	*60*
0.513	1144	J05243	**39**	*58*	**38**	*61*			**42**	*53*	**36**	*65*
0.428	**1745***	M16038	**40**	*57*	**42**	*55*			**41**	*53*	**40**	*57*
-	1809	M21624	**36**	*64*	**44**	*54*	**39**	*56*				
0.353	**4951**	Y07604			**43**	*55*			**43**	*53*		
0.453	**4196***	X17042			**45**	*53*					**44**	*52*
-	5543	D00749					**41**	*51*	**40**	*54*		
-	**1975**	M34344							**44**	*52*	**45**	*50*
0.36	7119*	U29175			**47**	*52*					**46**	*50*
-	**6079**	U59632			**39**	*57*						
0.256	2408	M96803									**43**	*53*
-	5542	M37271			**46**	*52*						
0.204	6388	S54005	**44**	*51*								
-	**5950**	M29610					**42**	*50*				

Continued on Next Page...

Table 5.3 – *Continued*

(d) Threshold-based System										
			Fitness Evaluation Size							
	Gene	Accession	20000		25000		30000		35000		40000	
IG Rate	Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.735	**4847***	X95735	**1**	*1220*	**1**	*1196*	**1**	*1311*	**1**	*1239*	**1**	*1294*
0.747	**2288***	M84526	**2**	*712*	**2**	*698*	**2**	*858*	**2**	*806*	**2**	*779*
0.376	**1779**	M19507	**5**	*516*	**4**	*574*	**3**	*714*	**3**	*657*	**3**	*642*
0.742	**1882***	M27891	**3**	*599*	**3**	*587*	**4**	*599*	**4**	*606*	**4**	*628*
0.568	**2354***	M92287	**4**	*576*	**5**	*546*	**6**	*565*	**6**	*529*	**6**	*539*
0.475	**2121***	M63138	**6**	*493*	**6**	*532*	**5**	*577*	**5**	*568*	**5**	*568*
0.429	**804**	HG1612-HT1612	**7**	*327*	**7**	*351*	**7**	*358*	**7**	*390*	**7**	*357*
0.674	**6041**	L09209	**8**	*237*	**8**	*292*	**9**	*252*	**8**	*274*	**8**	*304*
0.453	**4196***	X17042	**9**	*213*	**9**	*249*	**8**	*286*	**9**	*245*	**9**	*260*
0.512	**4328***	X59417	**12**	*200*	**10**	*218*	**10**	*248*	**10**	*243*	**11**	*223*
0.353	**4951**	Y07604	**11**	*203*	**12**	*200*	**11**	*238*	**12**	*196*	**10**	*243*
0.488	**2642***	U05259	**10**	*207*	**11**	*203*	**12**	*221*	**11**	*226*	**12**	*207*
0.628	**1685**	M11722	**13**	*164*	**15**	*147*	**13**	*174*	**13**	*176*	**14**	*159*
0.275	**4211**	X51521	**15**	*158*	**16**	*139*	**14**	*171*	**15**	*160*	**13**	*190*
0.718	**3252**	U46499	**14**	*159*	**13**	*167*	**15**	*166*	**14**	*175*	**15**	*148*
0.306	**1962**	M33680	**16**	*133*	**14**	*161*	**16**	*159*	**16**	*148*	**17**	*143*
-	**1796**	M20902	**21**	*118*	**18**	*122*	**18**	*151*	**17**	*139*	**16**	*147*
-	5952	U05255	**23**	*103*	**20**	*116*	**17**	*158*	**18**	*131*	**19**	*140*
0.572	3320*	U50136	**17**	*126*	**19**	*121*	**19**	*137*	**21**	*119*	**20**	*133*
0.462	6215	M19508	**18**	*122*	**17**	*123*	**20**	*137*	**22**	*119*	**21**	*133*
0.543	**1829**	M22960	**19**	*119*	**27**	*96*	**24**	*117*	**20**	*122*	**18**	*141*
-	**1975**	M34344	**24**	*92*	**23**	*102*	**21**	*131*	**25**	*109*	**22**	*131*
0.547	**4377**	X62654	**22**	*105*	**22**	*105*	**23**	*119*	**19**	*126*	**26**	*102*
0.678	**6855***	M31523	**20**	*118*	**24**	*100*	**27**	*107*	**23**	*114*	**23**	*118*
0.403	**6200***	M28130	**32**	*72*	**21**	*108*	**22**	*123*	**24**	*112*	**24**	*113*
0.462	2111*	M62762	**29**	*82*	**28**	*93*	**25**	*108*	**26**	*99*	**25**	*111*
0.403	**5501***	Z15115	**26**	*91*	**29**	*87*	**28**	*100*	**32**	*83*	**27**	*100*
0.341	**1239**	L07633	**28**	*86*	**26**	*97*	**29**	*90*	**34**	*82*	**28**	*99*
0.493	2020*	M55150	**27**	*86*	**30**	*85*	**32**	*81*	**27**	*89*	**30**	*80*
0.562	**2402***	M96326	**31**	*73*	**34**	*75*	**26**	*107*	**28**	*86*	**34**	*73*
0.388	**5772***	U22376	**25**	*92*	**25**	*98*	**43**	*61*	**33**	*82*	**32**	*76*
0.209	**412**	D42043	**30**	*82*	**32**	*85*	**33**	*81*	**29**	*86*	**36**	*71*
-	**5950**	M29610	**34**	*60*	**33**	*79*	**31**	*86*	**37**	*73*	**29**	*89*
-	**6079**	U59632	**35**	*57*	**31**	*85*	**30**	*86*	**38**	*69*	**33**	*75*
0.403	**6201***	Y00787	**33**	*70*	**36**	*70*	**34**	*77*	**36**	*73*	**35**	*71*
0.392	6539*	X85116			**35**	*71*	**35**	*74*	**30**	*84*	**31**	*78*
0.26	**668**	D86967	**39**	*53*	**39**	*58*	**39**	*68*	**41**	*59*	**38**	*61*
0.429	**1928***	M31303			**37**	*63*	**42**	*64*	**35**	*77*	**37**	*64*
0.517	1674	M11147	**37**	*54*			**38**	*68*	**31**	*83*	**44**	*53*
-	1941	M31994			**38**	*59*	**41**	*64*	**39**	*66*	**46**	*51*
-	6796	J02982			**43**	*52*	**40**	*66*	**40**	*61*	**39**	*61*
0.735	**1834***	M23197			**41**	*54*	**37**	*69*	**42**	*56*	**41**	*58*
0.443	3258*	U46751					**36**	*70*	**43**	*55*	**47**	*50*
0.261	**6049**	U89922					**47**	*52*	**45**	*52*	**42**	*56*
0.601	**4373**	X62320	**38**	*54*			**45**	*53*	**46**	*51*		
0.643	**6376***	M83652					**46**	*53*			**40**	*59*
0.512	**758**	D88270	**41**	*52*			**44**	*56*				
0.747	**760**	D88422	**40**	*52*							**43**	*55*
-	4409	X64594			**42**	*52*	**49**	*51*				
0.252	4095	X06948					**48**	*51*			**45**	*51*
0.428	**1745***	M16038			**40**	*57*						
-	5445	X04526	**36**	*55*								
0.23	6184	M26708							**44**	*53*		
-	7128	M71243					**50**	*50*				

Both the sigmoid and the tanh based systems have 49 genes in common and have an identical set of the first-11 selected genes. This might be due to both sigmoid and tanh based systems being non-linear functions, which able to explore the correlation between features within the data. Furthermore, both the sigmoid and the tanh based systems used the logistic curve (i.e. S-shape curve, see Figure 3.10 on page 81) for squashing the activation value of each set of genes to a specific activation range before the output is generated by the network. A logistic curve relates to the growth in the learning process. At the initial stage of the learning, the growth is exponential, then as saturation begins (at the middle stage of the learning), the growth slows and at the final stage of the learning (i.e. maturity), growth stops. This curve provides better discrimination

between data classes. Whilst, the linear and the threshold based systems have 52 common genes. This is due to both systems performing simple linear computation on the activation value for each set of genes and not squashing the activation results. Both the linear and the threshold functions utilised a straight line (see Figure 3.10) to discriminate data classes rather than the logistic curve. However, they do not have genes in identical rankings, mainly because the threshold based system restricted the activation of the network node only when it exceeds the defined threshold value in the system.

A comparison between the genes extracted by each system and the original work by Golub et al. (1999) was conducted (see Table 5.3). Amongst the selected genes in each system, both the *sigmoid* and the *tanh based* systems have 20 genes, including the top 4-ranked genes in the systems, which were consistent with the top-50 genes reported by Golub et al. Meanwhile, the *linear* and the *threshold based* system have 24 matching genes when compared to the reported genes by Golub et al. Amongst these common genes, 18 were overlapped in all systems. This indicates that our method is effective in extracting informative genes from ALL/AML data set and the data set is not being normalised. In Golub et al. (1999) work, the ALL/AML data set had been normalised with zero mean and unit standard deviation. Some relevant works on the ALL/AML data set is presented in Table C.1 in Appendix C.

By comparing the ranking order of the genes extracted by the GANN systems and the IG method, genes 1882, 2288, 4847 and 2354 are the top-4 most significant genes selected by both the sigmoid and the tanh based systems, and these genes were consistent with the top-50 genes reported by Golub et al. (1999). For the linear based system, the top-4 significant genes are 4847, 2288, 2354 and 2121, which were also consistent with the top-50 genes reported by Golub et al.. Meanwhile the threshold based system identified gene 1779 as one of the top-4 important genes, instead of gene 2354 that was highly rated by the sigmoid, the linear and the tanh based systems. For the IG method, the top-4 significant genes are 2288 and 760 (both genes have the equal IG rate of. 0.747), 1882 (IG rate = 0.742), 4847 and 1834 (both genes have equal IG rate of 0.735) and 3252 (IG rate = 0.718). Amongst the IG selected genes, gene 3252 is ranked in-between the top-11 and the top-20 significant genes by all the GANN systems and gene 1834 is the least significant genes in all the GANN systems as it does not have much correlation with other selected genes in the systems. For gene 760, both the linear and the threshold based systems have poor ranking on this genes, however, in both the sigmoid and the tanh based systems, gene 760 is ranked in-between the top-13 and the top-17 significant genes. The gene ranking discrepancy between the sigmoid/tanh based systems and the linear/threshold based systems confirmed our observation on the sigmoid and the tanh based systems in which the use of logistic curve provides features which benefits data classification and simultaneously, these features pose a certain degree of correlation with other selected features. The main reason for such gene ranking discrepancy between GANN system and the IG method is due to the fact that the IG method measures the distance

between features to its nearest class independently, meaning that each of these features can be used as an independent primary feature to categorise data classes as each feature provides a high classification accuracy in the data set. Unlike the IG method, GANN explores the correlation between features to the data classes. Therefore, a feature in the GANN system may not provides high classification accuracy in the data set; however, using a group of features extracted by the GANN system, a certain level of classification accuracy might be achieved.

5.3.4 The SRBCTs microarray data set

For SRBCTs data set in Table 5.4, surprisingly, both the *sigmoid* and the *threshold based* systems have the highest number of extracted genes, i.e 70 genes in total, compared to the *linear based* system which has 69 extracted genes and the *tanh based* system which has extracted 68 genes from the data set. Amongst these genes, 49 genes were overlapped in all systems, including the first-17 genes selected by each system. Both the sigmoid and the tanh based systems have 61 genes in common, while the linear and the threshold based systems have 65 common genes.

By comparing our findings with the original work conducted by Khan et al. (2001), the linear based system has the lowest number of overlapping genes with the top 96 genes reported by Khan et al. Even though, the top-10 ranked genes in all systems were consistent with the genes reported by Khan et al. Amongst the 69 genes identified by the linear based system, 39 genes were consistent with the genes reported by Khan et al., meanwhile, the sigmoid and the threshold based systems have 43 and 41 matching genes, respectively. The tanh based system has the highest number of matching genes, i.e. 44 genes, to the genes reported by Khan et al. Amongst these common genes, 34 were overlapped in all systems. This would suggest that the tanh based system is the most effective system to be used in this study. Some relevant works on the SRBCTs data set is presented in Table C.2 in Appendix C.

We also observed that more matching genes with the genes reported by Khan et al. were found in the fitness evaluation size 35000.

Table 5.4: The list of extracted genes in SRBCTs data set by each system based on the population size 300. *Freq.* is the number of times that the gene is selected. Genes highlighted in **Boldface** are common genes that were identified by all systems. Genes marked with "*" symbol are genes that matched with the genes reported in the original study.

(a) Sigmoid-based System

			Fitness Evaluation Size									
	Gene	Image	20000		25000		30000		35000		40000	
IG Rate	Index	Id	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.655	**742***	812105	**1**	*2320*	**1**	*2305*	**1**	*2365*	**1**	*2449*	**1**	*2366*
0.64	**1389***	770394	**3**	*1860*	**3**	*1915*	**3**	*1979*	**2**	*2073*	**2**	*2113*
0.572	**509***	207274	**2**	*1900*	**2**	*1968*	**2**	*1991*	**3**	*1979*	**3**	*1989*
0.858	**1955***	784224	**4**	*1546*	**4**	*1619*	**4**	*1700*	**4**	*1766*	**4**	*1802*
0.736	**246***	377461	**5**	*1520*	**5**	*1575*	**5**	*1632*	**5**	*1572*	**5**	*1602*
0.723	**545***	1435862	**6**	*1362*	**6**	*1423*	**6**	*1500*	**6**	*1485*	**6**	*1506*
0.484	**187***	296448	**7**	*1021*	**7**	*1064*	**7**	*1115*	**7**	*1198*	**7**	*1142*
0.708	**1601***	629896	**8**	*932*	**8**	*960*	**8**	*986*	**8**	*987*	**8**	*982*
0.784	**1954***	814260	**9**	*731*	**9**	*772*	**9**	*770*	**9**	*807*	**9**	*817*
0.607	**2046***	244618	**10**	*503*	**10**	*523*	**10**	*573*	**10**	*603*	**10**	*555*
0.592	**1645***	52076	**11**	*419*	**11**	*424*	**14**	*406*	**12**	*434*	**13**	*425*
0.798	**846***	183337	**12**	*402*	**14**	*393*	**11**	*439*	**13**	*420*	**12**	*440*
0.548	**153***	383188	**14**	*361*	**13**	*406*	**12**	*433*	**11**	*434*	**11**	*448*
0.669	**255***	325182	**13**	*373*	**12**	*422*	**13**	*415*	**14**	*411*	**14**	*417*
0.62	**1319***	866702	**16**	*266*	**15**	*288*	**15**	*341*	**16**	*287*	**15**	*313*
0.724	**554***	461425	**15**	*275*	**16**	*246*	**16**	*310*	**15**	*292*	**16**	*311*
0.658	**1606**	624360	**17**	*217*	**17**	*240*	**17**	*273*	**17**	*264*	**17**	*264*
0.851	**1003***	796258	**18**	*207*	**18**	*220*	**18**	*226*	**18**	*215*	**18**	*243*
0.509	**1158***	814526	**20**	*168*	**21**	*162*	**19**	*179*	**20**	*184*	**19**	*205*
0.567	**1916***	80109	**19**	*171*	**19**	*173*	**21**	*163*	**21**	*173*	**21**	*169*
0.453	**335***	1469292	**23**	*147*	**22**	*155*	**20**	*176*	**19**	*194*	**23**	*150*
0.365	**1084**	878652	**21**	*153*	**20**	*163*	**22**	*158*	**22**	*160*	**22**	*168*
0.756	**1386**	745019	**22**	*151*	**24**	*144*	**23**	*153*	**23**	*156*	**20**	*172*
0.482	**1***	21652	**24**	*133*	**25**	*132*	**24**	*149*	**25**	*148*	**24**	*144*
0.448	**1207**	143306	**25**	*121*	**23**	*151*	**25**	*147*	**24**	*149*	**26**	*131*
0.766	129*	298062	**28**	*106*	**27**	*121*	**26**	*130*	**27**	*124*	**25**	*133*
0.561	**976**	786084	**26**	*118*	**26**	*125*	**29**	*108*	**29**	*113*	**28**	*124*
0.562	**85***	297392	**31**	*92*	**33**	*92*	**30**	*105*	**28**	*122*	**27**	*126*
0.89	**836***	241412	**27**	*107*	**34**	*91*	**33**	*98*	**26**	*124*	**30**	*112*
0.705	1764*	44563	**32**	*89*	**30**	*97*	**27**	*121*	**31**	*104*	**31**	*104*
0.512	**236***	878280	**29**	*98*	**32**	*92*	**32**	*99*	**37**	*88*	**29**	*118*
0.587	**1055***	1409509	**30**	*93*	**31**	*96*	**38**	*83*	**30**	*112*	**34**	*101*
0.817	**783***	767183	**37**	*71*	**29**	*98*	**28**	*110*	**35**	*93*	**33**	*101*
0.49	**1434***	784257	**33**	*80*	**38**	*87*	**35**	*88*	**32**	*99*	**35**	*97*
0.521	**417***	395708	**34**	*77*	**40**	*67*	**31**	*104*	**33**	*98*	**36**	*95*
0.544	**1613***	80338	**36**	*72*	**35**	*91*	**36**	*87*	**39**	*86*	**32**	*102*
0.546	**1884***	609663	**38**	*68*	**28**	*102*	**34**	*90*	**36**	*92*	**39**	*83*
0.749	**2050***	295985	**40**	*65*	**37**	*89*	**37**	*85*	**38**	*86*	**38**	*87*
1	**123**	236282	**35**	*74*	**39**	*68*	**40**	*74*	**34**	*97*	**37**	*91*
0.567	**1116**	626502	**42**	*59*	**36**	*89*	**39**	*79*	**40**	*84*	**41**	*67*
0.514	**165**	283315	**41**	*63*	**42**	*64*	**47**	*57*	**41**	*82*	**40**	*78*
0.502	**2186**	208699	**39**	*66*	**47**	*60*	**43**	*62*	**42**	*79*	**45**	*60*
0.755	**1387**	770394	**46**	*55*	**41**	*66*	**42**	*63*	**45**	*66*	**43**	*64*
0.615	1708*	43733	**44**	*56*	**52**	*54*	**44**	*60*	**43**	*71*	**49**	*58*
0.441	**1327**	491565	**45**	*55*	**49**	*56*	**53**	*52*	**44**	*67*	**42**	*65*
0.479	1932	782811	**43**	*58*	**55**	*52*	**52**	*53*	**53**	*54*	**44**	*62*
0.814	**842**	810057	**49**	*52*	**43**	*64*	**51**	*54*	**63**	*50*	**51**	*57*
0.474	**188**	435953	**52**	*51*	**53**	*54*	**48**	*57*	**52**	*55*	**48**	*59*
0.444	430	379708			**45**	*62*	**50**	*55*	**46**	*65*	**56**	*53*
0.519	1700	796475			**44**	*63*	**46**	*60*	**47**	*62*	**59**	*50*
0.479	1662*	377048	**53**	*50*			**49**	*56*	**59**	*50*	**52**	*56*
0.539	368*	1473131	**47**	*54*	**56**	*52*	**41**	*65*				
0.373	**2157**	244637	**48**	*53*					**48**	*62*	**55**	*53*
0.673	**107***	365826			**51**	*55*			**51**	*55*	**53**	*54*
0.43	**380***	289645	**50**	*51*					**54**	*54*	**46**	*59*
0.401	**365**	1473131					**55**	*51*	**61**	*50*	**47**	*59*
0.533	1536	530185			**54**	*54*			**57**	*52*	**57**	*52*
0.452	1497	203003			**48**	*58*	**45**	*60*				
0.373	1301	346696							**49**	*58*	**54**	*54*
0.398	2199*	135688	**51**	*51*					**50**	*57*		
0.432	**951***	841620			**50**	*55*					**58**	*51*
0.582	1738	771323			**57**	*52*			**58**	*51*		
0.397	1105*	788107					**54**	*51*	**62**	*50*		
0.572	566*	357031			**46**	*60*						
0.571	1915*	840942									**50**	*57*
0.446	1776	768246							**56**	*53*		
0.323	1991	740554							**55**	*53*		
0.328	139	729964			**58**	*51*						
-	407	195751					**56**	*50*				
0.267	847	265874							**60**	*50*		

Continued on Next Page...

Table 5.4 – *Continued*

(b) Linear-based System												
			Fitness Evaluation Size									
	Gene	Image	20000		25000		30000		35000		40000	
IG Rate	Index	Id	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.655	**742***	812105	**1**	*2220*	**1**	*2342*	**1**	*2449*	**1**	*2424*	**1**	*2606*
0.64	**1389***	770394	**2**	*1637*	**2**	*1737*	**2**	*1898*	**2**	*1879*	**2**	*2044*
0.858	**1955***	784224	**3**	*1554*	**3**	*1671*	**3**	*1793*	**3**	*1811*	**3**	*1894*
0.572	**509***	207274	**4**	*1203*	**4**	*1156*	**4**	*1233*	**4**	*1230*	**4**	*1300*
0.723	**545***	1435862	**5**	*1025*	**5**	*1124*	**5**	*1142*	**5**	*1170*	**5**	*1157*
0.736	**246***	377461	**6**	*770*	**6**	*822*	**6**	*884*	**6**	*889*	**6**	*997*
0.708	**1601***	629896	**7**	*707*	**7**	*818*	**7**	*857*	**7**	*848*	**7**	*842*
0.798	**846***	183337	**8**	*564*	**8**	*607*	**8**	*658*	**8**	*671*	**8**	*710*
0.548	**153***	383188	**9**	*544*	**9**	*586*	**10**	*612*	**9**	*639*	**9**	*664*
0.484	**187***	296448	**10**	*472*	**10**	*516*	**9**	*623*	**10**	*619*	**10**	*625*
0.479	1932	782811	**11**	*441*	**11**	*494*	**11**	*519*	**11**	*512*	**11**	*569*
0.658	**1606**	624360	**12**	*440*	**12**	*483*	**12**	*477*	**13**	*485*	**12**	*514*
0.607	**2046***	244618	**14**	*390*	**14**	*441*	**13**	*461*	**12**	*500*	**13**	*511*
0.851	**1003***	796258	**13**	*418*	**13**	*450*	**14**	*445*	**14**	*454*	**14**	*473*
0.784	**1954***	814260	**16**	*330*	**15**	*374*	**15**	*441*	**15**	*401*	**15**	*442*
0.669	**255***	325182	**15**	*347*	**16**	*311*	**16**	*345*	**16**	*362*	**16**	*379*
0.509	**1158***	814526	**19**	*246*	**19**	*256*	**17**	*279*	**17**	*308*	**17**	*338*
0.567	**1916***	80109	**18**	*278*	**17**	*270*	**19**	*256*	**18**	*290*	**19**	*295*
0.453	**335***	1469292	**17**	*279*	**20**	*252*	**18**	*265*	**20**	*269*	**18**	*309*
0.448	**1207**	143306	**20**	*226*	**18**	*263*	**20**	*255*	**21**	*248*	**21**	*260*
0.562	**85***	297392	**21**	*218*	**21**	*241*	**23**	*237*	**19**	*288*	**20**	*261*
0.546	**1884***	609663	**22**	*212*	**22**	*217*	**24**	*237*	**22**	*241*	**22**	*249*
0.482	**1***	21652	**24**	*188*	**24**	*210*	**21**	*247*	**23**	*210*	**23**	*245*
0.724	**554***	461425	**25**	*182*	**23**	*215*	**22**	*241*	**25**	*209*	**24**	*245*
0.567	**1116**	626502	**23**	*201*	**25**	*204*	**25**	*217*	**26**	*203*	**25**	*243*
0.756	**1386**	745019	**27**	*172*	**27**	*186*	**26**	*217*	**24**	*209*	**27**	*198*
0.89	**836***	241412	**26**	*173*	**28**	*185*	**27**	*202*	**29**	*171*	**26**	*209*
0.749	**2050***	295985	**31**	*126*	**26**	*187*	**28**	*164*	**28**	*171*	**32**	*150*
0.592	**1645***	52076	**30**	*134*	**30**	*147*	**29**	*145*	**27**	*173*	**28**	*197*
0.817	**783***	767183	**29**	*138*	**29**	*159*	**31**	*142*	**31**	*152*	**29**	*169*
0.561	**976**	786084	**28**	*141*	**31**	*146*	**30**	*145*	**30**	*156*	**30**	*160*
0.514	**165**	283315	**32**	*115*	**32**	*138*	**33**	*115*	**32**	*146*	**31**	*155*
0.62	**1319***	866702	**33**	*113*	**36**	*111*	**32**	*136*	**35**	*128*	**35**	*124*
0.814	**842**	810057	**37**	*83*	**35**	*111*	**35**	*112*	**34**	*130*	**33**	*137*
0.587	**1055***	1409509	**34**	*107*	**34**	*115*	**38**	*91*	**33**	*132*	**34**	*127*
0.365	**1084**	878652	**36**	*88*	**33**	*115*	**36**	*104*	**36**	*120*	**38**	*100*
0.538	758	47475	**39**	*73*	**38**	*98*	**34**	*112*	**37**	*118*	**36**	*122*
0.441	**1327**	491565	**40**	*68*	**37**	*109*	**37**	*92*	**41**	*79*	**37**	*106*
0.755	**1387**	770394	**35**	*89*	**39**	*86*	**40**	*89*	**38**	*94*	**40**	*93*
0.512	**236***	878280	**44**	*63*	**42**	*68*	**39**	*90*	**43**	*75*	**39**	*95*
1	**123**	236282	**41**	*68*	**41**	*72*	**41**	*87*	**46**	*70*	**41**	*87*
0.452	1497	203003	**38**	*76*	**47**	*60*	**42**	*76*	**39**	*85*	**44**	*83*
0.582	1738	771323	**43**	*64*	**43**	*66*	**44**	*73*	**45**	*72*	**45**	*80*
0.521	**417***	395708	**50**	*54*	**40**	*79*	**57**	*51*	**40**	*81*	**43**	*85*
0.368	1067	489489	**49**	*55*	**51**	*57*	**43**	*75*	**42**	*77*	**47**	*73*
0.432	**951***	841620	**45**	*62*	**48**	*59*	**45**	*67*	**48**	*66*	**46**	*73*
0.49	**1434***	784257	**48**	*56*	**46**	*60*	**52**	*55*	**52**	*59*	**42**	*87*
0.502	**2186**	208699	**52**	*51*	**44**	*64*	**46**	*65*	**44**	*73*	**52**	*61*
0.533	1536	530185	**47**	*57*	**50**	*59*	**47**	*64*	**51**	*60*	**49**	*68*
0.526	74	193913	**42**	*64*	**54**	*54*	**49**	*62*	**57**	*54*	**48**	*70*
0.544	**1613***	80338	**46**	*60*	**49**	*59*			**49**	*62*	**50**	*62*
0.571	1915*	840942			**55**	*54*	**50**	*58*	**50**	*62*	**51**	*62*
0.373	**2157**	244637			**53**	*54*	**54**	*53*	**47**	*68*	**53**	*59*
0.401	**365**	1434905	**51**	*52*			**55**	*53*	**53**	*56*	**54**	*59*
0.497	1626*	811000	**54**	*50*			**59**	*50*	**56**	*55*	**57**	*56*
0.588	585	68977			**56**	*50*	**56**	*53*	**54**	*55*	**59**	*51*
0.33	1295	344134	**53**	*50*							**58**	*55*
0.419	251*	486787			**45**	*62*						
0.289	276*	868304					**48**	*62*				
0.279	937	789204									**55**	*58*
0.648	589	769657			**52**	*57*						
0.474	**188**	435953					**51**	*56*				
0.387	166	897177									**56**	*56*
0.265	94	809603					**53**	*55*				
0.539	**380***	289645							**55**	*55*		
0.327	1066*	486110							**58**	*53*		
0.673	**107***	365826							**59**	*51*		
0.373	1301	346696					**58**	*50*				
0.489	1911	898219									**60**	*50*

Continued on Next Page...

Table 5.4 – *Continued*

(c) Tanh-based System												
			Fitness Evaluation Size									
	Gene	Image	20000		25000		30000		35000		40000	
IG Rate	Index	Id	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.655	**742***	812105	**1**	*1911*	**1**	*1963*	**1**	*2019*	**1**	*2065*	**1**	*2123*
0.572	**509***	207274	**2**	*1796*	**2**	*1870*	**2**	*1865*	**2**	*1941*	**2**	*1982*
0.64	**1389***	770394	**3**	*1621*	**3**	*1774*	**3**	*1763*	**3**	*1873*	**3**	*1866*
0.736	**246***	377461	**4**	*1468*	**4**	*1545*	**4**	*1576*	**4**	*1571*	**4**	*1640*
0.858	**1955***	784224	**5**	*1279*	**5**	*1376*	**5**	*1452*	**5**	*1521*	**5**	*1490*
0.723	**545***	1435862	**6**	*1093*	**6**	*1152*	**6**	*1251*	**6**	*1347*	**6**	*1307*
0.484	**187***	296448	**7**	*934*	**7**	*1058*	**7**	*1141*	**7**	*1254*	**7**	*1233*
0.708	**1601***	629896	**8**	*918*	**8**	*990*	**8**	*955*	**8**	*1047*	**8**	*1012*
0.784	**1954***	814260	**9**	*662*	**9**	*745*	**9**	*784*	**9**	*802*	**9**	*825*
0.607	**2046***	244618	**10**	*458*	**10**	*465*	**10**	*516*	**10**	*521*	**10**	*528*
0.592	**1645***	52076	**11**	*378*	**11**	*423*	**12**	*398*	**11**	*457*	**12**	*433*
0.669	**255***	325182	**13**	*332*	**12**	*387*	**11**	*420*	**12**	*419*	**11**	*446*
0.548	**153***	383188	**12**	*338*	**14**	*309*	**13**	*371*	**13**	*364*	**13**	*395*
0.62	**1319***	866702	**14**	*287*	**15**	*300*	**14**	*335*	**14**	*335*	**14**	*350*
0.798	**846***	183337	**15**	*279*	**13**	*315*	**15**	*321*	**15**	*308*	**15**	*312*
0.724	**554***	461425	**16**	*279*	**16**	*300*	**16**	*284*	**16**	*295*	**16**	*284*
0.851	**1003***	796258	**17**	*180*	**17**	*197*	**17**	*210*	**18**	*196*	**17**	*239*
0.705	1764*	44563	**18**	*145*	**18**	*186*	**18**	*193*	**17**	*207*	**18**	*207*
0.365	**1084**	878652	**21**	*128*	**19**	*173*	**20**	*175*	**23**	*137*	**19**	*197*
0.658	**1606**	624360	**19**	*139*	**20**	*140*	**19**	*175*	**19**	*158*	**21**	*147*
0.756	**1386**	745019	**23**	*106*	**23**	*126*	**21**	*141*	**21**	*141*	**20**	*160*
0.766	129*	298062	**20**	*133*	**21**	*138*	**22**	*123*	**22**	*137*	**25**	*121*
0.509	**1158***	814526	**22**	*113*	**27**	*111*	**24**	*117*	**20**	*142*	**23**	*132*
0.567	**1916***	80109	**24**	*104*	**22**	*130*	**27**	*109*	**26**	*130*	**22**	*133*
0.561	**976**	786084	**27**	*93*	**25**	*114*	**23**	*118*	**27**	*115*	**24**	*121*
0.512	**236***	878280	**29**	*88*	**28**	*106*	**26**	*116*	**24**	*134*	**27**	*111*
0.587	**1055***	1409509	**40**	*59*	**29**	*100*	**25**	*117*	**25**	*134*	**29**	*110*
0.482	**1***	21652	**25**	*103*	**24**	*117*	**30**	*93*	**30**	*101*	**38**	*88*
0.453	**335***	1469292	**28**	*90*	**34**	*90*	**29**	*95*	**33**	*99*	**26**	*112*
0.448	**1207**	143306	**30**	*83*	**30**	*100*	**35**	*85*	**32**	*100*	**30**	*109*
0.615	1708*	43733	**33**	*76*	**26**	*112*	**34**	*88*	**29**	*102*	**33**	*96*
0.49	**1434***	784257	**35**	*74*	**33**	*92*	**33**	*88*	**28**	*106*	**28**	*110*
0.521	**417***	395708	**31**	*79*	**31**	*96*	**28**	*95*	**35**	*95*	**31**	*102*
0.562	**85***	297392	**26**	*93*	**32**	*94*	**36**	*82*	**34**	*98*	**36**	*91*
0.89	**836***	241412	**34**	*74*	**35**	*87*	**32**	*89*	**39**	*70*	**32**	*102*
0.544	**1613***	80338	**36**	*65*	**36**	*78*	**31**	*92*	**31**	*100*	**40**	*81*
0.749	**2050***	295985	**32**	*76*	**37**	*77*	**38**	*76*	**37**	*77*	**35**	*94*
1	**123**	236282	**37**	*61*	**41**	*68*	**40**	*67*	**36**	*87*	**39**	*82*
0.817	**783***	767183	**45**	*50*	**42**	*64*	**42**	*63*	**41**	*69*	**34**	*96*
0.572	566*	357031	**38**	*60*	**45**	*60*	**39**	*72*	**40**	*69*	**41**	*78*
0.444	430	379708	**42**	*57*	**47**	*54*	**37**	*80*	**44**	*63*	**44**	*72*
0.546	**1884***	609663	**44**	*53*	**39**	*73*	**46**	*57*	**49**	*57*	**47**	*63*
0.539	368*	1473131	**41**	*57*	**44**	*62*			**38**	*72*	**37**	*89*
0.479	1662*	377048			**38**	*77*	**43**	*62*	**46**	*60*	**42**	*74*
0.398	2199*	135688	**39**	*59*			**48**	*55*	**42**	*66*	**46**	*68*
0.673	**107***	365826			**48**	*52*	**41**	*64*	**50**	*56*	**43**	*73*
0.519	1700	796475			**49**	*52*	**53**	*50*	**53**	*52*	**45**	*68*
0.401	**365**	1434905	**47**	*50*			**45**	*57*	**47**	*58*	**50**	*55*
0.567	**1116**	626502					**44**	*60*	**43**	*63*	**51**	*53*
0.502	**2186**	208699	**46**	*50*	**40**	*68*			**55**	*51*		
0.446	1776	768246	**43**	*55*			**47**	*56*			**53**	*53*
0.373	**2157**	244637					**51**	*52*	**48**	*57*	**55**	*52*
0.814	**842**	810057			**52**	*51*	**49**	*55*	**56**	*50*		
0.432	**951***	841620							**51**	*54*	**56**	*50*
0.328	139	729964							**45**	*62*		
0.755	**1387**	770394			**43**	*62*						
0.474	**188**	435953			**46**	*57*						
0.441	**1327**	491565									**48**	*56*
0.514	**165**	283315									**49**	*55*
0.497	1263*	324494							**52**	*53*		
0.344	1721	40643									**52**	*53*
0.4	257	740801			**50**	*52*						
0.43	**380***	289645									**54**	*52*
0.323	1991	740554			**51**	*52*						
0.402	2144*	308231					**50**	*52*				
0.588	585	68977					**52**	*51*				
0.609	1980*	841641							**54**	*51*		
0.327	1066*	486110							**57**	*50*		

Continued on Next Page...

Table 5.4 – *Continued*

(d) Threshold-based System												
			Fitness Evaluation Size									
	Gene	Image	20000		25000		30000		35000		40000	
IG Rate	Index	Id	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
0.655	**742***	812105	**1**	*2248*	**1**	*2370*	**1**	*2439*	**1**	*2491*	**1**	*2443*
0.64	**1389***	770394	**2**	*1625*	**2**	*1862*	**2**	*1908*	**2**	*1913*	**2**	*2009*
0.858	**1955***	784224	**3**	*1618*	**3**	*1751*	**3**	*1797*	**3**	*1794*	**3**	*1908*
0.572	**509***	207274	**4**	*1565*	**4**	*1586*	**4**	*1573*	**4**	*1585*	**4**	*1615*
0.723	**545***	1435862	**5**	*1254*	**5**	*1297*	**5**	*1342*	**5**	*1389*	**5**	*1317*
0.736	**246***	377461	**6**	*1053*	**6**	*1124*	**6**	*1193*	**6**	*1204*	**6**	*1233*
0.708	**1601***	629896	**7**	*782*	**7**	*858*	**7**	*920*	**7**	*927*	**7**	*898*
0.484	**187***	296448	**8**	*708*	**8**	*839*	**8**	*898*	**8**	*902*	**8**	*893*
0.798	**846***	183337	**9**	*513*	**9**	*559*	**9**	*599*	**10**	*579*	**9**	*594*
0.548	**153***	383188	**10**	*507*	**10**	*515*	**11**	*508*	**9**	*584*	**10**	*576*
0.784	**1954***	814260	**11**	*432*	**11**	*486*	**10**	*537*	**12**	*526*	**11**	*564*
0.607	**2046***	244618	**12**	*421*	**12**	*478*	**12**	*475*	**11**	*537*	**12**	*531*
0.658	**1606**	624360	**13**	*410*	**13**	*389*	**13**	*420*	**13**	*395*	**13**	*447*
0.669	**255***	325182	**14**	*328*	**14**	*361*	**15**	*357*	**14**	*393*	**14**	*421*
0.402	1932	782811	**15**	*283*	**15**	*352*	**16**	*342*	**15**	*364*	**15**	*370*
0.851	**1003***	796258	**17**	*266*	**16**	*306*	**14**	*359*	**16**	*342*	**16**	*352*
0.567	**1916***	80109	**16**	*274*	**17**	*283*	**17**	*288*	**17**	*312*	**19**	*276*
0.592	**1645***	52076	**18**	*243*	**18**	*239*	**19**	*251*	**19**	*259*	**17**	*301*
0.509	**1158***	814526	**19**	*242*	**19**	*227*	**20**	*238*	**18**	*290*	**18**	*286*
0.453	**335***	1469292	**20**	*236*	**24**	*207*	**21**	*237*	**21**	*241*	**20**	*262*
0.562	**85***	297392	**22**	*196*	**21**	*214*	**18**	*252*	**20**	*253*	**23**	*215*
0.724	**554***	461425	**21**	*203*	**23**	*209*	**22**	*217*	**23**	*213*	**21**	*256*
0.482	**1***	21652	**25**	*171*	**20**	*222*	**23**	*213*	**22**	*223*	**24**	*210*
0.546	**1884***	609663	**23**	*187*	**22**	*211*	**28**	*176*	**26**	*190*	**22**	*216*
0.756	**1386**	745019	**24**	*178*	**27**	*181*	**24**	*206*	**24**	*201*	**25**	*207*
0.567	**1116**	626502	**26**	*170*	**25**	*200*	**25**	*190*	**25**	*192*	**26**	*205*
0.62	**1319***	866702	**27**	*157*	**28**	*174*	**26**	*181*	**29**	*172*	**27**	*190*
0.448	**1207**	143306	**31**	*137*	**29**	*171*	**27**	*179*	**27**	*185*	**28**	*188*
0.89	**836***	241412	**29**	*138*	**26**	*182*	**29**	*166*	**28**	*176*	**29**	*179*
0.561	**976**	786084	**28**	*138*	**30**	*155*	**30**	*159*	**30**	*162*	**30**	*166*
0.365	**1084**	878652	**30**	*138*	**32**	*128*	**33**	*124*	**31**	*147*	**31**	*156*
0.817	**783***	767183	**34**	*107*	**31**	*138*	**31**	*132*	**32**	*141*	**33**	*145*
0.749	**2050***	295985	**33**	*117*	**34**	*110*	**32**	*130*	**33**	*130*	**32**	*145*
0.514	**165**	283315	**32**	*119*	**36**	*99*	**34**	*113*	**34**	*127*	**34**	*126*
0.814	**842**	810057	**38**	*83*	**33**	*123*	**36**	*108*	**36**	*114*	**37**	*112*
0.587	**1055***	1409509	**35**	*91*	**35**	*109*	**35**	*110*	**37**	*105*	**35**	*125*
0.538	758	47475	**44**	*64*	**37**	*94*	**37**	*100*	**35**	*120*	**36**	*123*
0.571	1915*	840942	**39**	*82*	**43**	*66*	**39**	*96*	**39**	*88*	**38**	*101*
0.512	**236***	878280	**36**	*89*	**46**	*65*	**38**	*100*	**38**	*91*	**42**	*83*
0.755	**1387**	770394	**40**	*82*	**41**	*73*	**43**	*78*	**40**	*87*	**39**	*91*
1	**123**	236282	**42**	*73*	**45**	*65*	**40**	*89*	**41**	*87*	**41**	*87*
0.441	**1327**	491565	**37**	*88*	**39**	*76*	**44**	*77*	**44**	*78*	**43**	*78*
0.452	1497	203003	**46**	*61*	**38**	*78*	**41**	*81*	**43**	*81*	**44**	*76*
0.521	**417***	395708	**41**	*77*	**51**	*58*	**42**	*78*	**45**	*72*	**48**	*70*
0.49	**1434***	784257	**43**	*68*	**44**	*66*	**46**	*74*	**42**	*82*	**54**	*62*
0.544	**1613***	80338	**48**	*57*	**47**	*63*	**49**	*60*	**52**	*62*	**40**	*87*
0.502	**2186**	208699	**52**	*54*	**42**	*68*	**48**	*62*	**57**	*57*	**46**	*72*
0.533	1536	530185	**47**	*59*	**48**	*61*	**56**	*54*	**49**	*66*	**51**	*69*
0.526	74	193913	**51**	*54*	**50**	*59*	**55**	*55*	**47**	*68*	**53**	*64*
0.373	**2157**	244637	**45**	*62*	**52**	*57*			**46**	*69*	**45**	*73*
0.289	276*	868304	**49**	*56*			**47**	*70*	**55**	*57*	**50**	*69*
0.368	1067	489489			**49**	*60*	**50**	*60*	**62**	*51*	**47**	*72*
0.33	1295	344134			**54**	*56*	**53**	*56*	**50**	*63*	**52**	*65*
0.432	**951***	841620					**45**	*76*	**48**	*67*	**55**	*60*
0.582	1738	771323			**40**	*75*			**59**	*54*	**57**	*55*
0.588	585	68977					**54**	*55*	**53**	*61*	**56**	*57*
0.474	**188**	435953			**56**	*51*	**58**	*50*			**49**	*70*
0.497	1626*	811000	**50**	*56*	**57**	*51*			**61**	*52*		
0.387	166	897177					**51**	*59*			**59**	*52*
0.401	**365**	1434905							**56**	*57*	**58**	*53*
0.43	**380***	289645			**55**	*55*					**60**	*51*
0.419	251*	486787	**54**	*52*							**61**	*51*
0.373	1301	346696							**51**	*63*		
0.648	589	769657							**54**	*60*		
0.766	129*	298062					**52**	*57*				
0.479	1662*	377048			**53**	*57*						
0.673	**107***	365826							**58**	*55*		
0.519	1700	796475	**53**	*53*								
0.398	2199*	135688							**60**	*53*		
0.413	1634	82903					**57**	*51*				

By comparing the ranking order of the genes extracted by the GANN systems and the IG method, common genes 742, 1389, 509 and 1955 are the top-4 most significant genes selected by all the GANN systems and

these genes were matched with the top-96 genes reported by Khan et al. (2001). Genes 246, 545 and 187 were also highly ranked by both the sigmoid and the tanh based systems. Gene 545 was ranked at the 4th place in both the linear and the threshold based systems. Using the IG method, the top-5 most significant genes are genes 836 (IG rate = 0.89), 1955 (IG rate = 0.858), 1003 (IG rate = 0.851), 783 (IG rate = 0.817) and 842 (IG rate = 0.814). Genes 742, 1389, 509 and 545 were lowly ranked by the IG method, with the IG rates of 0.655, 0.64, 0.572 and 0.723, respectively. This is because these genes did not pose any classification benefit when they were evaluated independently. Amongst the top-5 significant genes selected by the IG method, gene 1003 is ranked in-between the top-11 to the top-20 significant genes by all GANN systems, and the remaining genes 1003, 783 and 842 are lowly ranked by all GANN systems, meaning that these genes does not have strong correlation with other selected genes. This shows that the GANN system was a better approach for analysing microarray data as it explores the correlation between features in the data sets.

5.3.5 Discussion

The findings reported in this section are based on the performance of each system in extracting informative genes in the effect of different fitness evaluations. For synthetic data sets, all systems have equivalent performance in identifying all the 30 predefined genes in the synthetic data set 1. The linear based system outperformed the other three system in the synthetic data set 2, by consistently identifying 18 out of 30 predefined genes in almost all fitness evaluation sizes. However, the linear based system also identified, consistently, the highest number of noisy genes compared to both sigmoid and threshold based systems. The tanh based system is the only system that did not select any noisy gene in the synthesis data set 2.

For microarray data sets, both linear and threshold based systems outperformed the sigmoid and the tanh based systems in the ALL/AML data set by having 26 genes overlapped with the genes reported in the original study. In the case of SRBCTs data set, the tanh based system outperformed the other three systems with 44 matching genes to the original study. In terms of the number of extracted genes in the microarray data, the linear based system, however, has the highest number of extracted genes in both data sets and the tanh based system outperformed the other three system with the lowest number of extracted genes in both microarray data sets.

The findings suggested that the tanh based system is, amongst all systems, the most effective systems to be used for microarray data sets.

5.4 The biological sensible of the extracted genes

In Sections 5.1 and 5.2, we discussed the performance of each system based on different data sets: synthetic data sets and microarray data sets, and the implication of two vital GA parameters: population and evolution; to extract the most relevant genes from the respective data sets. Based on the findings in these sections, we found that the sufficient population size for microarray studies should range from 200 to 300 and the fitness evaluation sizes should range from 20000 to 40000.

This section presents a global view on the biological relevance of extracted genes presented in Section 5.3, i.e. based on the population size 300 and the fitness evaluation sizes, ranging from 20000 to 40000. The relevant studies of the disease will be reviewed and followed by a discussion on the gene findings of the GANN prototype. A total of 74 and 90 genes extracted from the ALL/AML and SRBCTs data sets, respectively, as is indicated in Tables 5.3 and 5.4 in Section 5.3 will be discussed.

In this section, the term *"gene"* indicates the identified features from the respective data sets and does not possess biological context in the medical field. The term *"chromosome'* represents a threadlike strand of DNA in the nucleus of a biological cell.

5.4.1 The ALL/AML microarray data

Leukaemia is the blood cancer disease that is caused by the immaturity of blood cells in the bone marrow. There are generally two main groups of leukaemia cancer, i.e. lymphoblastic leukaemia, derived from the abnormal growth of lymphoblasts, these are the primitive progenitor cells originating in the bone marrow; and myelogeneous leukaemia which arises from the immaturity of myeloid precursor cells in the bone marrow. These two types of leukaemia cancer can be either chronic or acute. Due to the leukaemia data set that we used in this thesis being acute leukaemia, we only discuss acute-based leukaemia cancers, i.e. *Acute Lymphoblastic Leukaemia (ALL)* and *Acute Myelogeneous Leukaemia (AML).*

Leukaemia cancer is normally incurred by an abnormality of chromosomes in a cell. Several chromosomal abnormalities has been reported in both ALL and AML which leads to tremendous advances in leukaemia research. Most of the identified abnormalities involve translocation of different chromosomes in leukaemia cells. *Translocations* is a type of "Structural Chromosomal Aberrations" that often cause human infertility as they interfere with the normal distribution of chromosomes during meiosis (Robinson, 2003). In cancer cases, it is a chromosome abnormality caused by re-arrangement of parts between two non-homogenous chromosomes.

A standard designation *t(A;B)(p1;q2)* outlined by the International System for Human Cytogenetic Nomen-

clature (ISCN) is used to denote a translocation between chromosome A and chromosome B. The second part of information, i.e. (p1;q2), denotes the precise location (i.e. regions, bands, and so on) of a gene, within chromosome A and chromosome B, respectively. The (p1;q2) can be interpreted as band p1, is located in chromosome A and band q2, is located in chromosome B. The terms *"p"* and *"q"* indicate the length of a chromosome in where "p" indicating the short arm of the chromosome, i.e. short DNA sequence, while "q" indicating the long arm of the chromosome. Table 5.5 shows some chromosomal translocations in leukaemia cancer.

Table 5.5: Some characteristic translocations in Leukaemias.

Translocation	Genes involved	Leukaemia Type
t(1;19)(q23;q13)	TCF3 (E2A), PBX1	ALL - L1/L2; B-ALL
t(7;9)(q35;q34.3)	TCR@ (TCRB), NOTCH1	T-ALL
t(8;21)(q22;q22)	AML1, ETO	AML - M2
t(9;11)	Various	AML - M5; B-ALL
t(9;22)(q34;q11)	BCR, ABL1	AML - M1/M2 (CML); B-ALL
t(12;21)(p13;q22)	ETV6 (TEL), RUNX1 (AML1)	B-ALL
t(15;17)(q22;q11-12)	PML, RARα	AML - M3 (APL)
t(v;11)(v;q23) v = 9-19	Various	AML - M1/M4/M5
del(5q34)	EBF1	B-ALL
MLL rearrangement (11q23)	MLL, ALL1, HRX	B-ALL

Chromosomal translocations in varying chromosomes from 8 to 19, chromosome 21 and chromosome 23 are normally associated with leukaemia disease. The AML1-ETO fusion in $t(8;21)(q22;q22)$ causes a higher remission rate in AML patients with M2 band (based on French-American-British (FAB) classification) than those without it (Golub et al., 1999; Sheer and Shipley, 2005). The AML1 protein contains a DNA- and protein-binding region that is homogenous to the CBFA component and the Drosophila segmentation gent runt. The ETO gene is not normally expressed in myeloid cells.

The BCR3 type PML-RARα fusion in $t(15;17)(q22;q11.2-12)$ is the signature marker for AML M3 band, which is more commonly known as *Acute Promyelocytic Leukaemia (APL)*. RARα is a transcription factor that normally binds to all-trans retinoic acid (ATRA) which then binds to retinoic acid response element (RARE) in the promoters of many genes and transcriptional activation domain (Sheer and Shipley, 2005). A potential role for ATRA has been advocated in APL patients (Sanz et al., 2000).

The Philadelphia (Ph) chromosome, i.e. BCR-ABL fusion in $t(9;22)(q34;q11)$ is commonly found in Chronic Myelogeneous Leukaemia (CML) patients and AML patients in M1 and M2 subtypes. The immaturity of the Ph chromosome can also occur in ALL (Sheer and Shipley, 2005). BCR gene encodes a serine kinase. The ABL protein is a tyrosine kinase that contains a DNA binding domain and it functions as a negative regulator of cell growth when the cell cycle is over-expressed.

A translocation between chromosomes 8 and 21 and a vary translocation of chromosomes varying from 9 to 19, may also possibly invoke AML. A translocation between chromosomes 8 and 21 will cause the elevation of the c-myc activity, a gene involved in cell replication (Robinson, 2003). A translocation between chromosomes 9 and 19 contributes to the AML patients with M1, M4 and M5 bands.

Deletion of the gene EBF1 in del(5q34), an early B-cell factor, could also contribute to leukaemogenesis (Mullighan et al., 2007). The EBF1 gene is required for the development of B-cells and with the gene TCF3 regulates the expression of B-lineage genes.

In addition to chromosomal translocations, up-regulation of certain genes expressed in leukaemia cells is due to chromosomal rearrangement and could contribute to the development of the cancer.

The positive expression in the genes CD1, CD3, CD4, CD8 and terminal deoxynucleotidyl transferase (TdT) correspond to an intermediate thymocyte stage of T-cell ALL differentiation (Strauchen, 2001). The expression of TdT is normally restricted to lymphoid precursors, especially in T-cell and the non-positive TdT (i.e. suppressed) might be detected in B-cells when it is associated with the down-regulation of genes CD10 and CD34; a high WBC count and the rearrangement of gene MLL (Mixed Lineage Leukaemia) (Liu et al., 2004b).

Regulation of the gene CD179a/b in chromosome 22 is useful as a marker for differentiating the immature and mature B-cell precursors, in which the up-regulation of genes CD179a/b were specifically expressed in precursor B-cell lymphoblastic lymphomas, but not in mature B-cell lymphomas of childhood (Kiyokawa et al., 2004).

Suppression in the gene myeloperoxidase (MPO) in adult ALL patients reduces the survival rate of the patient (Arber et al., 2001).

The gene podocalyxin, a CD34 family member, is a useful hematopoietic blast marker for patients with abnormal hematopoietic cells. The increased level of podocalyxin is commonly expressed in both AML and ALL, as well as in cutaneous myeloid sarcoma disease (Kelley et al., 2005).

The translocations in chromosomes with t(8;21) is a distinct clinicopathologic entity for patients who have been diagnosed with AML cancer (Khoury et al., 2004). The translocations t(8;21) displayed a higher level of CD34, HLA-DR and MPO expression, and a lower level of CD13 and CD33 expression.

The gene CD99 (MIC2) is characteristically expressed in precursor B- and T-cell lymphoblastic leukaemias and lymphomas, as well as in Ewing sarcoma and primitive neuroectodermal tumours (Kang and Dunphy, 2006). It is most intensely expressed by immature thymus T-lineage cells and in the earliest stage of precursor B-cells.

A total of 74 genes were extracted from the ALL/AML data set. Among 74 genes, 39 genes were found by all systems, 6 were found by 3 systems (i.e. 4 in sigmoid/linear/tanh based; 2 in sigmoid/tanh/threshold based), 21 genes were identified by 2 systems (i.e. 6 in sigmoid/tanh based; 11 in linear/threshold based; 2 in linear/tanh based; 2 in sigmoid/linear based), 3 genes were identified by only the linear based system, 3 genes by the tanh based system and 2 genes by the threshold based system. Table 5.6 presents all 74 extracted genes ordered by the cytoband of the gene. The complete list of the genes extracted by each system is presented in Table 5.3 on page 138. The correlation between genes within the sample and the gene interaction within the sample is presented in Figure 5.7.

Table 5.6: The summary list of ALL/AML genes. The genes are ordered according to the cytoband of the selected gene. The *Index* denotes the row number of the selected gene. The *Gene Accession* is the id number of the selected gene. The *Symbol* represents the official abbreviation of the selected gene based on the NCBI Genbank (as of October 2009). The *GeneID* is the NCBI genbank number of the selected gene. The *Cytoband* is the location of the selected gene. The *Group* is the cancer group to which the selected gene belongs. Genes marked with "§" are genes that were match with the genes reported by Golub et al. (1999). Genes marked with "†" were genes identified by all systems in the GANN prototype.

Index	Gene Accession	Symbol	GeneID	Cytoband	Group
Index_6510	U23852	LCK	3932	1p34.3	ALL
Index_804	HG1612-HT1612†	MARCKSL1	65108	1p35.1	ALL
Index_1928	M31303†§	STMN1	3925	1p36.1-p35	ALL
Index_5445	X04526	GNB1	2782	1p36.33	ALL
Index_4095	X06948	FCER1A	2205	1q23	AML
Index_6388	S54005	TMSB10	9168	2p11.2	ALL
Index_2408	M96803	SPTBN1	6711	2p21	ALL
Index_4291	X56468	YWHAQ	10971	2p25.1	ALL
Index_6184	M26708	PTMA	5757	2q35-q36	ALL
Index_5501	Z15115†§	TOP2B	7155	3p24	ALL
Index_412	D42043†	RFTN1	23180	3p24.3	ALL
Index_668	D86967†	EDEM1	9695	3p26.2	ALL
Index_760	D88422†	CSTA	1475	3q21	AML
Index_6200	M28130†§	IL8	3576	4q13-q21	AML
Index_6201	Y00787†§	IL8	3576	4q13-q21	AML
Index_7128	M71243	GYPA	2993	4q28.2-q31.1	AML
Index_5952	U05255	GYPB	2994	4q28-q31	AML
Index_6796	J02982	GYPB	2994	4q28-q31	AML
Index_5950	M29610†	GYPE	2996	4q31.1	AML

Continued on Next Page...

Table 5.6 – *Continued*

Index	Gene Accession	Symbol	GeneID	Cytoband	Group
Index_1630	L47738§	CYFIP2	26999	5q33.3	ALL
Index_3258	U46751§	SQSTM1	8878	5q35	AML
Index_3320	U50136§	LTC4S	4056	5q35	AML
Index_2354	M92287†§	CCND3	896	6p21	ALL
Index_4409	X64594	RHAG	6005	6p21.1-p11	AML
Index_4438	X66401	TAP1/TAP2	6890/6891	6p21.3	ALL
Index_6049	U89922†	LTB	4050	6p21.3	ALL
Index_5772	U22376†§	MYB	4602	6q22-q23	ALL
Index_4211	X51521†	EZR	7430	6q25.2-q26	ALL
Index_4847	X95735†§	ZYX	7791	7q32	AML
Index_1745	M16038†§	LYN	4067	8q13	AML
Index_6539	X85116§	STOM	2040	9p34.1	AML
Index_1941	M31994	ALDH1	216	9q21.13	AML
Index_1144	J05243	SPTAN1	6709	9q33-q34	ALL
Index_4196	X17042†§	SRGN	5552	10q22.1	AML
Index_1685	M11722†	DNTT	1791	10q23-q24	ALL
Index_4229	X52056	SPI1	6688	11p11.2	AML
Index_3984	U94855	EIF3F	8665	11p15.4	ALL
Index_1962	M33680†	CD81	975	11p15.5	ALL
Index_2121	M63138†§	CTSD	1509	11p15.5	AML
Index_6702	X97267	PTPRCAP	5790	11q13.3	ALL
Index_4050	X03934	CD3D	915	11q23	ALL
Index_6041	L09209†	APLP2	334	11q23-q25—11q24	AML
Index_3252	U46499†	MGST1	4257	12p12.3-p12.1	AML
Index_4377	X62654†	CD63	967	12q12-q13	AML
Index_1239	L07633†	PSME1	5720	14q11.2	ALL
Index_1809	M21624	TRD@	6964	14q11.2	ALL
Index_4328	X59417†§	PSMA6	5687	14q13	ALL
Index_4680	X82240	TCL1A	8115	14q32.1	ALL
Index_2020	M55150§	FAH	2184	15q23-q25	AML
Index_6225	M84371	CD19	930	16p11.2	ALL
Index_2111	M62762§	ATP6V0C	527	16p13.3	AML
Index_4951	Y07604†	NME4	4833	16p13.3	AML
Index_6271	M33493	TPSAB1	7177	16p13.3	AML
Index_1975	M34344†	ITCA2B	3674	17q21.32	AML
Index_4373	X62320†	GRN	2896	17q21.32	AML
Index_6079	U59632†	SEPT4	5414	17q22-q23	AML
Index_2335	M89957	CD79B	974	17q23	ALL

Continued on Next Page...

Table 5.6 – *Continued*

Index	Gene Accession	Symbol	GeneID	Cytoband	Group
Index_1779	M19507†	MPO	4353	17q23.1	AML
Index_6215	M19508	MPO	4353	17q23.1	AML
Index_5542	M37271	CD7	924	17q25.2-q25.3	ALL
Index_5543	D00749	CD7	924	17q25.2-q25.3	ALL
Index_2642	U05259†§	CD79A	973	19p13.2	ALL
Index_7119	U29175§	SMARCA4	6597	19p13.2	ALL
Index_2288	M84526†§	CFD	1675	19p13.3	AML
Index_2402	M96326†§	AZU1	566	19p13.3	AML
Index_6855	M31523†§	TCF3	6929	19p13.3	ALL
Index_1796	M20902†	APOC1	341	19q13.2	AML
Index_1834	M23197†§	CD33	945	19q13.3	AML
Index_1674	M11147	FTL	2512	19q13.3	AML
Index_1882	M27891†§	CST3	1471	20p11.21	AML
Index_1704	M13792§	ADA	100	20q12-q13.11	ALL
Index_1829	M22960†	CTSA	5476	20q13.1	AML
Index_758	D88270†	IGL@	3535	22q11.1-q11.2	AML
Index_6376	M83652†§	CFP	5199	Xp11.3-p11.23	AML

Amongst the 74 identified genes, 35 were expressed in ALL class and the remaining 39 genes were expressed in AML class.

The already proven efficient biomarkers from the existing medical studies that were involved in the translocations were found by the GANN prototype. These biomarkers are DNTT (Index_1685, previously known as TdT), SMARCA4 (Index_7119), TCF3 (Index_6855), CD3D (Index_4050), CD79 (Index_2642 and Index_2335), MPO (Index_1779 and Index_6215) and APLP2 (Index_6041). The genes DNTT, SMARCA4 and TCF3 were reported as being under-expressed in an MLL translocation (Armstrong et al., 2002). Amongst these biomarkers, 5 were found by all systems (i.e. Index_1685, Index_6855, Index_2642, Index_1779 and Index_6041) and the remaining 2 were identified by multiple systems (i.e. Index_7119 in linear/tanh based and Index_2335 in sigmoid/linear/tanh based). Please refer to Table 5.3 on page 138 for the complete list of the extracted genes by each system.

DNTT (Index_1685) plays a role in DNA repair and is normally expressed in the normal and the early stage of malignant B-cell development. It was expressed in some B-cell ALL samples in Figure 5.7 which suggests that these samples were taken at the early stage of B-ALL development.

SMARCA4 (Index_7119) is a SWI/SNF family member, is a tumour suppression gene that interacts with the tumour suppression gene p53, for p53-driven transcriptional activation and it plays an important role in

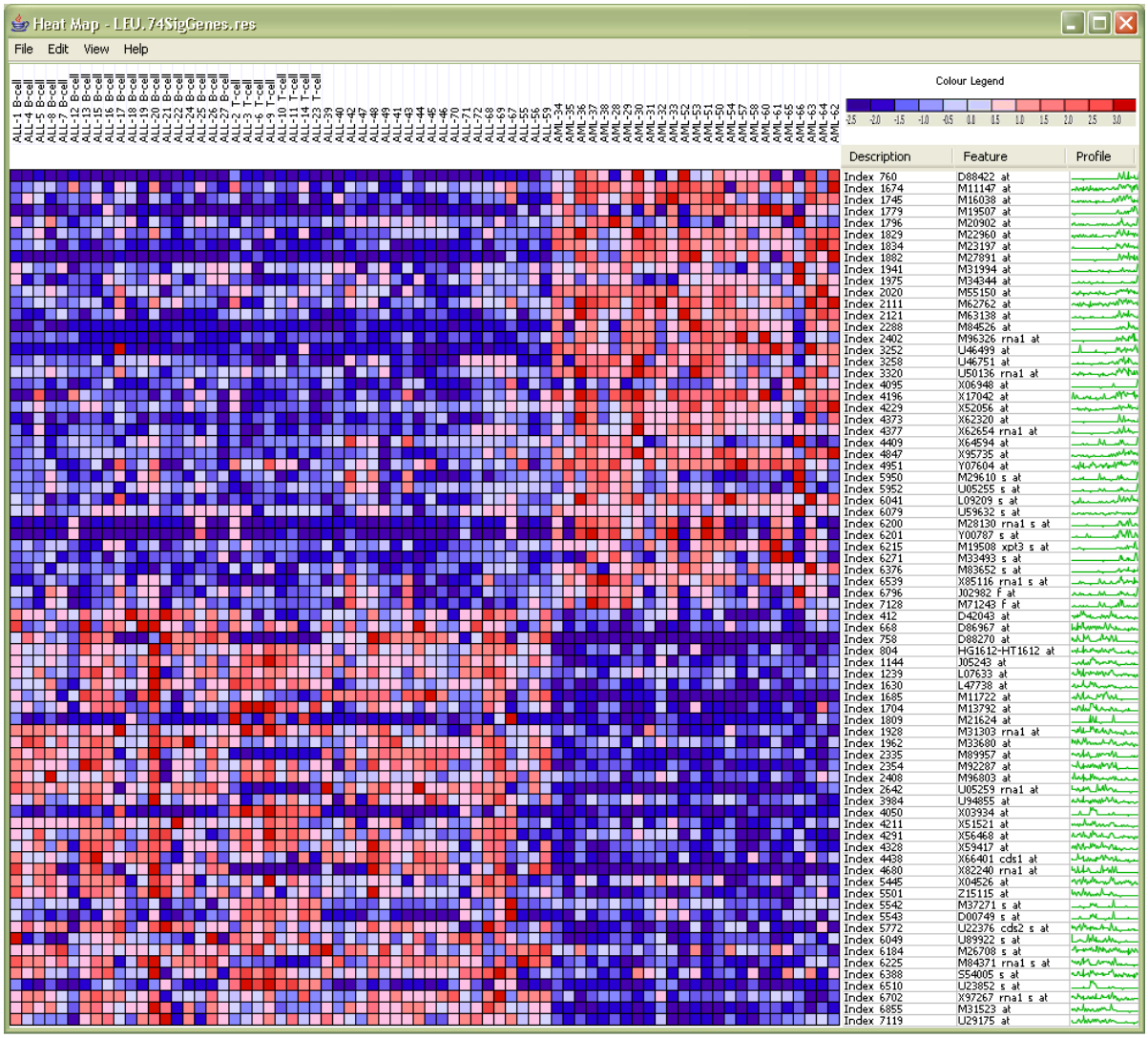


Figure 5.7: The heatmap of ALL/AML genes. The gene with high intensity (i.e. excited by the fluorescence dyes) denotes in red and the gene with low intensity denotes in blue.

p53-mediated cell cycle control (Lee et al., 2002). The mutation of gene SMARCA4 in the absence of MYC amplification is associated with the lung tumourigenesis (Medina et al., 2008). The presence of the gene SMARCA4 in almost all the ALL samples, but not in the AML samples in Figure 5.7, indicates that it is associated in ALL tumourigenesis.

TCF3 (Index_6855) is a transcriptional regulator that coordinates the regulation of the expression of genes involved in cell survival, cell cycle progression, lipid metabolism, stress response and lymphoid maturation (Schwartz et al., 2006) and is important for appropriate B-cell development. A similar observation to SMARCA4, TCF3 is associated with ALL tumourigenesis, as is indicated in Figure 5.7

CD3D (Index_4050) is a T-cell receptor involved in signal transduction and the development of T-cell. The deficiency in CD3D is characterised by the absence of T-cells but it does not affect the development of normal

B-cells (Dadi et al., 2003). This observation is also indicated in Figure 5.7 where CD3D is highly expressed in all the T-ALL samples and in a B-ALL sample. This may suggest that possibly combined immunodeficiency genes in the B-ALL induce the activation of the CD3D receptor.

CD79A (Index_2642) and CD79B (Index_2335) are B lymphocyte antigen receptors required for the development of normal B-cells. The down-regulation of both CD79 genes are the hallmark of tumoural B lymphocytes in B-cell chronic lymphocytic leukaemia (Vuillier et al., 2005) and the coexpression of CD79A with AML1/ETO fusion in t(8;21) represents biphenotypic acute leukaemia (He et al., 2007). Both CD79A and CD79B have expressed a moderate level in all ALL samples in Figure 5.7 but not in T-ALL samples.

APLP2 (Index_6041) belongs to the APP family that may assist in the regulation of haemostasis. This is indicated by the presence of APLP2 in most AML samples in Figure 5.7.

MPO (Index_1779 and Index_6215) is a key enzyme in the cellular response to hypoxia and consequent development of tissue fibrosis (Saed et al., 2009). The gene MPO is presented during abnormal myeloid cell differentiation. The high level of MPO in some AML samples in Figure 5.7 suggests the possibilities of the AML subgroups in the AML class.

In addition to the biomarkers, there are also genes which are only expressed in one of the subgroups of ALL. These genes includes genes that are only expressed in T-ALL, i.e. ADA (Index_1704) and CD7 (Index_5542 and Index_5543) and genes that are only expressed in B-ALL, i.e. IGL@ (Index_758), CD19 (Index_6225) and TCL1A (Index_4680). These genes could be potential signature markers in differentiating T-ALL and B-ALL. Amongst these genes, excepting Index_758 that was found by all systems, 3 were overlapped in sigmoid/tanh based systems (i.e. Index_1704, Index_5543 and Index_4680), Index_6225 in the linear based system and Index_5542 was solely identified by the tanh based system with the fitness evaluation size 25000 (see Table 5.3 on page 138).

ADA (Index_1704) catalyses the hydrolysis of adenosine to inosine. The deficiency in this gene could causes a dysfunction of both B and T lymphocytes with the impairment of cell immunity and decreased production of immunologlobulins (EntrezGene: GeneID 100). The elevation of this gene may activate the Th1 response to the disease (Kisacik et al., 2009) and have been associated with congenital haemolytic anaemia. The gene CD7 (Index_5542 and Index_5543) is the immunoglobulin superfamily member that is normally found on thymocytes and mature T-cells. It is required in T-cell interactions and also in T-cell/B-cell interaction during early lymphoid development (EntrezGene: GeneID 924).

IGL@ (Index_758) plays an important role in B-cell development. It contains lambda light chain in the germline organisation which allows it to recognise foreign antigens and to initiate immune responses (EntrezGene: GeneID 3535). CD19 (Index_6225) encodes the cell surface molecule which binds with the antigen

receptor of B lymphocytes to decrease the threshold for antigen receptor-dependent stimulation (EntrezGene: GeneID 930). It is constantly expressed in all stages of B lineage differentiation and is a reliable marker for diagnosing B-lineage ALL (Chen et al., 2004). TCL1A (Index_4680) is a powerful oncogene that is involved in the pathogenesis of mature T-cell leukaemia (Pekarsky et al., 2004). It plays an important role in controlling the growth and the effector T-cell functions (Hoyer et al., 2005). In Figure 5.7, TCL1A is not expressed in T-ALL samples, but in most B-ALL samples, which could suggests that the T-ALL samples in the ALL/AML data set are premature, i.e. in the early stages of T-cell development and TCL1A is predominantly associated with mature precursor lymphocytes but not differentiated B- or T-cells.

Amongst the identified genes, some genes may be involved in some known leukaemia chromosomal aberrations. These genes include E2A/PBX1 fusion related genes, i.e. FCER1A (Index_4095), SMARCA4 (Index_7119), CFD (Index_2288), AZU1 (Index_2402), TCF3 (Index_6855), APOC1 (Index_1796), CD33 (Index_1834) and FTL (Index_1674); TCR/NOTCH1 fusion related gene, i.e. SPTAN1 (Index_1144); BCR/ABL1 fusion related genes, i.e. SPTAN1 (Index_1144) and IGL@; MLL rearrangement genes, i.e. CD3D (Index_4050) and APLP2 (Index_6041).

The findings in this study shows that the GANN prototype is a robust feature extraction method that is able to extract highly informative genes from the raw (unprocessed) data set. This is indicated by the majority of the identified genes involved in leukaemogenesis and also some signature genes in differential B- and T-ALL. The tanh based system outperformed the other three systems to extract the biological significant genes from the data set.

5.4.2 The SRBCTs microarray data

The term *"Small round blue cell tumours"* (SRBCTs) is a generic name (category) used to describe a large number of malignant tumours that occur in childhood in the medical studies. They are characterised by small, round and relatively undifferentiated cells, such as Ewing's sarcoma, acute leukaemia, small cell mesothelioma, neuroblastoma, rhabdomyosarcoma, synovial sarcoma, Non-Hodgkin's lymphoma and many more small round cells. In this thesis, we examine only four types of SRBCTs tumours, which are *Burkitt's Lymphoma (NB)*, a subtype of Non-Hodgkin's Lymphoma, *Ewing's Sarcoma (EWS), Neuroblastoma (NB)* and *Rhabdomyosarcoma (RMS).*

Burkitt's lymphoma (BL) is a cancer yielded from the dysfunction of B lymphocytes, a type of white blood cell. This cancer is associated with the translocation of the c-myc gene with other genes, such as the translocation between c-myc gene with genes located in chromosome 14, i.e. t(8;14)(q24;32). The c-myc gene plays an important role in cell cycle progression and its dysfunctional, i.e. mutation, over-expression, rearrangement and translocation, are associated with a variety of haematopoietic malignancy (EntrezGene:

GeneID 4609).

Ewings sarcoma (EWS) is a bone malignant cancer that predominantly occurs in the second decade of life and commonly affects areas including the pelvis, the femur, the humerus and the ribs (Rajwanshi et al., 2009). The cell origin of this tumour is uncertain. EWS has a shared cytogenetic abnormality with the primitive neuroectodermal tumour (PNET) which is a small round cell malignancy that arises from the soft tissue or bone. The shared cytogenetic abnormality involves a translocation between chromosome 11 and chromosome 22, i.e. the EWS/FLI1 fusion in t(11;22)(q24;q12), a signature marker for EWS/PNET from other small round tumours (Owen et al., 2008). The EWS (EWSR1) gene acts as a strong transcriptional activator in various cellular processes and chromosomal translocations between this gene and other transcriptional factor genes could result in the production of chimeric proteins that are involved in tumourigenesis. The FLI1 gene is a member of ETS transcription factor that is involved in a wide variety of bodily regulation functions, including cell differentiation, cell cycle control, cell migration, cell proliferation, apoptosis and angiogenesis. The EWS/FLI1 fusion can be detected by the positive expression of CD99 (Rajwanshi et al., 2009) and the negative expression of CD45 (Bernsteina et al., 2006). Other less frequent translocations involved in this tumour are the EWS/ERG fusion in t(21;22) (Sorensen et al., 1994) and the EWS/ETV1 fusion in t(7;22)(p22;q12) (Jeon et al., 1995).

Neuroblastoma (NB) is the third most common, solid malignant tumour of infancy and childhood that predominantly occur in the male group (Rajwanshi et al., 2009). This tumour arises from the neuroblasts which are the undifferentiated precursor cells of the sympathetic nervous system. The etiology of NB is not well understood by medical scientists, however, there are cases where this tumour could be inherited from a parent and is associated with the mutation of the germline in the ALK gene (Mossé et al., 2008) at chromosome band 2p23. The ALK gene plays an important role in the development of the brain and exerts its effects on specific neurons in the nervous system. The presence of del(1p), i.e. deletion of chromosome 1 short arm, or MYCN amplification are also useful in detecting the early stage of NB development in the young age group (Sheer and Shipley, 2005).

Rhabdomyosarcoma (RMS) is a connective tissue related cancer that commonly occurs in children (Rajwanshi et al., 2009). This tumour can be classified into three main subtypes: embryonal rhabdomyosarcoma (ERMS), alveolar rhabdomyosarcoma (ARMS) and pleomorphic rhabdomyosarcoma (PRMS). The former two subtypes of RMS are almost undifferentiated based on the morphological result, however, at a molecular level, ARMS shows predominantly dissociated cells or chance formations and ERMS shows large tissue fragments with abundant eosinophilic material and disassociated cells (Rajwanshi et al., 2009). The Desmin protein has been reported as a useful marker for detecting large rhabdomyoblasts during the early differentiation of skeletal and smooth muscle cells, however, it is also positively expressed in smaller, less well

differentiated tumour cells (Rajwanshi et al., 2009). The other two markers: MyoD1 and myogenin, have a higher sensitivity (positive) than desmin in RMS detection and they can differentiate ARMS from the embryonal type (Rajwanshi et al., 2009). The MyoD1 gene involves in muscle regeneration and regulates muscle cell differentiation. The myogenin (MYOG) gene is a muscle-specific transcription factor that is essential in the development of functional skeletal muscle. Table 5.7 shows some cytogenetic differentiation in four types of SRBCTs tumours.

Table 5.7: Some cytogenetic differentiation in four types of SRBCTs.

Cytogenetic Difference	Genes involved	Tumour Type
t(8;14)(q24;q32)	c-myc (MYC), IGH@	BL
c-myc rearrangement	c-myc (MYC)	BL
t(11;22)(q24;q12)	EWS (EWSR1), FLI1	EWS
t(21;22)(q22;q12)	EWS (EWSR1), ERG	EWS
t(7;22)(p22;q12)	EWS (EWSR1), ETV1	EWS
$^{+}$CD99, $^{-}$CD45	CD99, CD45	EWS
ALK mutation	ALK	NB
del(1p)	Various	NB
$^{+}$DES, $^{+}$MYOD1, $^{+}$MYOG	DES, MYOD1, MYOG	RMS

A total of 90 genes were extracted from the SRBCTs data set. Among these 90 genes, 49 genes were found by all systems, 11 genes were found by 3 systems (i.e. 6 in sigmoid/linear/tanh based; 4 in sigmoid/tanh/threshold based; 1 in linear/tanh/threshold based), 18 genes were identified by 2 systems (i.e. 7 in linear/threshold based; 8 in sigmoid-/tanh-based and 1 in linear/tanh based) and 12 were detected by individual system (i.e. 3 by sigmoid based system, 3 by linear based system, 5 by tanh based system and 1 by threshold based system). Table 5.8 presents all 90 extracted genes ordered by the cytoband of the gene. The complete list of these 90 genes is presented in Table 5.4 in page 144. The correlation between genes within the sample and the gene interaction to the sample is presented in Figure 5.8.

Table 5.8: The summary list of SRBCTs. The genes are ordered according to the cytoband of the selected gene. The *Index* denotes the row number of the selected gene. The *Img Id* is the clone number of the selected gene. The *Symbol* represents the official abbreviation of the selected gene based on the NCBI Genbank (as of October 2009). The *GeneID* is the NCBI genbank number of the selected gene. The *Cytoband* is the location of the selected gene. The *Group* is the cancer group to which the selected gene belongs. Genes marked with "§" are genes that were match with the genes reported by Khan et al. (2001). Genes marked with "†" were genes identified by all systems in the GANN prototype.

Index	Img Id	Symbol	GeneID	Cytoband	Group
Index_251	486787§	CNN3	1266	1p22-p21	RMS/NB (<> BL)

Continued on Next Page...

Table 5.8 – *Continued*

Index	Img Id	Symbol	GeneID	Cytoband	Group
Index_1700	796475	FHL3	2275	1p34	EWS/RMS (<> BL/NB)
Index_1738	771323	PLOD1	5351	1p36.22	RMS
Index_742	812105†§	MLLT11	10962	1q21	NB (<> BL)
Index_1613	80338†§	SELENBP1	8991	1q21-q22	EWS (<> BL/NB)
Index_129	298062§	TNNT2	7139	1q32	RMS
Index_1067	489489	LBR	3930	1q42.1	BL
Index_1434	784257†§	KIF3C	3797	2p23	NB (<> BL)
Index_1662	377048§	MYO1B	4430	2q12-q34	NB (<> BL)
Index_2144	308231§	MYO1B	4430	2q12-q34	NB (<> BL)
Index_1105	788107§	BIN1	274	2q14	RMS (<> BL)
Index_566	357031§	TNFAIP6	7130	2q23.3	EWS
Index_783	767183†§	HCLS1	3059	3q13	BL
Index_1764	44563§	GAP43	2596	3q13.1-q13.2	NB
Index_2199	135688§	GATA2	2624	3q21.3	NB (<> BL)
Index_1066	486110§	PFN2	5217	3q25.1-q25.2	NB
Index_589	769657	PPP1R2	5504	3q29	BL
Index_236	878280†§	CRMP1	1400	4p16.1-p15	NB (<> BL)
Index_1319	866702†§	PTPN13	5783	4q21.3	EWS
Index_1601	629896†§	MAP1B	4131	5q13	NB (<> BL)
Index_1	21652†§	CTNNA1	1495	5q31	EWS/RMS (<> BL)
Index_1721	40643	PDGFRB	5159	5q31-q32	EWS/RMS (<> BL)
Index_1955	784224†§	FGFR4	2264	5q35.1-qter	RMS
Index_188	435953†	ITPR3	3710	6p21	BL/EWS (<> NB)
Index_1932	782811	HMGA1	3159	6p21	BL
Index_1606	624360†	PSMB8	5696	6p21.3	BL
Index_1634	82903	TAPBP	6892	6p21.3	BL/EWS
Index_1915	840942§	HLA-DPB1	3115	6p21.3	BL
Index_1916	80109†§	HLA-DQA1	3117	6p21.3	BL
Index_2186	208699†	KIAA1949	170954	6p21.3	BL
Index_846	183337†§	HLA-DMA	3108	6p21.3	BL
Index_1536	530185	CD83	9308	6p23	BL (<> RMS)
Index_407	195751	AKAP7	9465	6q23	EWS
Index_1327	491565†	CITED2	10370	6q23.3	EWS (<> BL)
Index_165	283315†	PGAM2	5224	7p13-p12	BL (<> RMS)
Index_1884	609663†§	PRKAR2B	5577	7q22	BL (<> RMS)
Index_1084	878652†	PMS2L12	392713	7q22.1	NB (<> BL)
Index_246	377461†§	CAV1	857	7q31.1	EWS
Index_1911	898219	MEST	4232	7q32	RMS (<> BL)

Continued on Next Page...

Table 5.8 – *Continued*

Index	Img Id	Symbol	GeneID	Cytoband	Group
Index_951	841620†§	DPYSL2	1808	8p22-p21	EWS/NB (<> BL)
Index_74	193913	LYN	4067	8q13	BL
Index_107	365826†§	GAS1	2619	9q21.3-q22	EWS/RMS (<> BL/NB)
Index_1645	52076†§	NOE1	10439	9q34.3	EWS (<> BL/RMS)
Index_276	868304§	ACTA2	59	10q23.3	BL
Index_166	897177	PGAM1	5223	10q25.3	BL (<> RMS)
Index_417	395708†§	DPYSL4	10570	10q26	NB (<> BL)
Index_139	729964	SMPD1	6609	11p15.4-p15.1	EWS/RMS
Index_1207	143306†	LSP1	4046	11p15.5	RMS
Index_187	296448†§	IGF2	3481	11p15.5	RMS
Index_509	207274†§	IGF2	3481	11p15.5	RMS
Index_1386	745019†	EHD1	10938	11q13	BL
Index_1980	841641§	CCND1	595	11q13	EWS/NB (<> BL)
Index_1263	324494§	HSPB2	3316	11q22-q23	RMS (<> BL/NB)
Index_1991	740554	RDX	5962	11q23	EWS (<> BL)
Index_842	810057†	CSDA	8531	12p13.1	<>NB
Index_1301	346696	TEAD4	7004	12p13.3-p13.2	RMS (<> BL)
Index_836	241412†§	ELF1	1997	13q13	BL
Index_1387	740604†	ISG20	3669	15q26	BL
Index_1497	203003	NME4	4833	16p13.3	EWS/NB (<> BL)
Index_85	297392†§	MT1L	4500	16q13	BL
Index_585	68977	PSMB10	5699	16q22.1	BL
Index_153	383188†§	RCVRN	5957	17p13.1	NB (<> BL)
Index_430	379708	CHD3	1107	17p13.1	EWS (<> BL)
Index_1003	796258†§	SGCA	6442	17q21	RMS
Index_365	1434905†	HOXB7	3217	17q21.3	EWS (<> BL/NB)
Index_976	786084†	CBX1	10951	17q21.32	NB
Index_554	461425†§	MYL4	4635	17q21-qter	RMS
Index_1626	811000§	LGALS3BP	3959	17q25	EWS/NB (<> BL)
Index_255	325182†§	CDH2	1000	18q11.2	NB
Index_1954	814260†§	KDSR	2531	18q21.3	EWS
Index_368	1473131§	TLE2	7089	19p13.3	EWS (<> BL/NB)
Index_847	265874	NFIC	4782	19p13.3	EWS
Index_380	289645†§	APLP1	333	19q13.1	EWS/NB (<> BL)
Index_257	740801	BCKDHA	593	19q13.1-q13.2	EWS (<> BL)
Index_1389	770394†§	FCGRT	2217	19q13.3	EWS
Index_1055	1409509†§	TNNT1	7138	19q13.4	RMS (<> BL)
Index_1158	814526†§	RBM38	55544	20q13.31	RMS/BL (<> NB)

Continued on Next Page...

Table 5.8 – *Continued*

Index	Img Id	Symbol	GeneID	Cytoband	Group
Index_1295	344134	IGLL3	91353	22q11.23	BL
Index_335	1469292†§	PIM2	11040	Xp11.23	BL
Index_123	236282†	WAS	7454	Xp11.4-p11.21	BL (<> NB)
Index_1708	43733§	GYG2	8908	Xp22.3	EWS
Index_545	1435862†§	CD99	4267	Xp22.32	EWS
Index_1776	768246	G6PD	2539	Xq28	NB
Index_94	809603	EST			BL
Index_758	47475				BL
Index_937	789204				BL/NB
Index_1116	626502†	ARPC1B	10095	7q22.1	BL
Index_2046	244618†§	EST			RMS (<> EWS/BL)
Index_2050	295985†§	CDK6	1021	7q21-q22	RMS/NB (<> EWS)
Index_2157	244637				NB

Unlike the ALL/AML data set which is oligonucleotide-based platform, the SRBCTs data set is cDNA-based. Due to the lack of a standard protocol in the cDNA microarray production, the integrity of the cDNA microarray is compromised by poor gene annotation. This has made the validation of the identified genes in this study difficult as an identical clone number (image id of the cDNA gene) could be used to label multiple greatly differential genes.

Amongst the 90 identified genes by the GANN prototype, 14 were highly expressed in RMS class, 17 in EWS class, 27 in BL class, 16 in NB class and 16 genes were moderately expressed in more than one class, including 2 expressed in EWS/BL, 5 in EWS/NB, 1 in RMS/BL, 1 in BL/NB, 2 in RMS/NB, 5 in RMS/EWS and 1 common gene that appeared in RMS/EWS/BL class. Amongst these genes, 5 were undefined in our findings due to the lack of sufficient gene information in the data set. These undefined genes are image id 244637 (Index_2157), 47475 (Index_758), 789204 (Index_937), 809603 (Index_94) and 244618 (Index_2046), as indicated in Table 5.8.

Some already proven efficient biomarkers in the existing medical studies have been found by the GANN prototype. These biomarkers are CD99 (Index_545) and GYG2 (Index_1708) for EWS tumourigenesis; and genes located in the chromosome band 11p15, i.e. SMPD1 (Index_139), LSP1 (Index_1207) and IGF2 (Index_187 and Index_509), which have been associated to RMS pathology. Four of these biomarkers were found by all systems (i.e. Index_545, Index_1207, Index_187 and Index_509), Index_1708 was identified by both the sigmoid and the tanh based systems; and Index_139 was solely identified by tanh based system with the fitness evaluation size 35000 (see Table 5.4 on page 144).

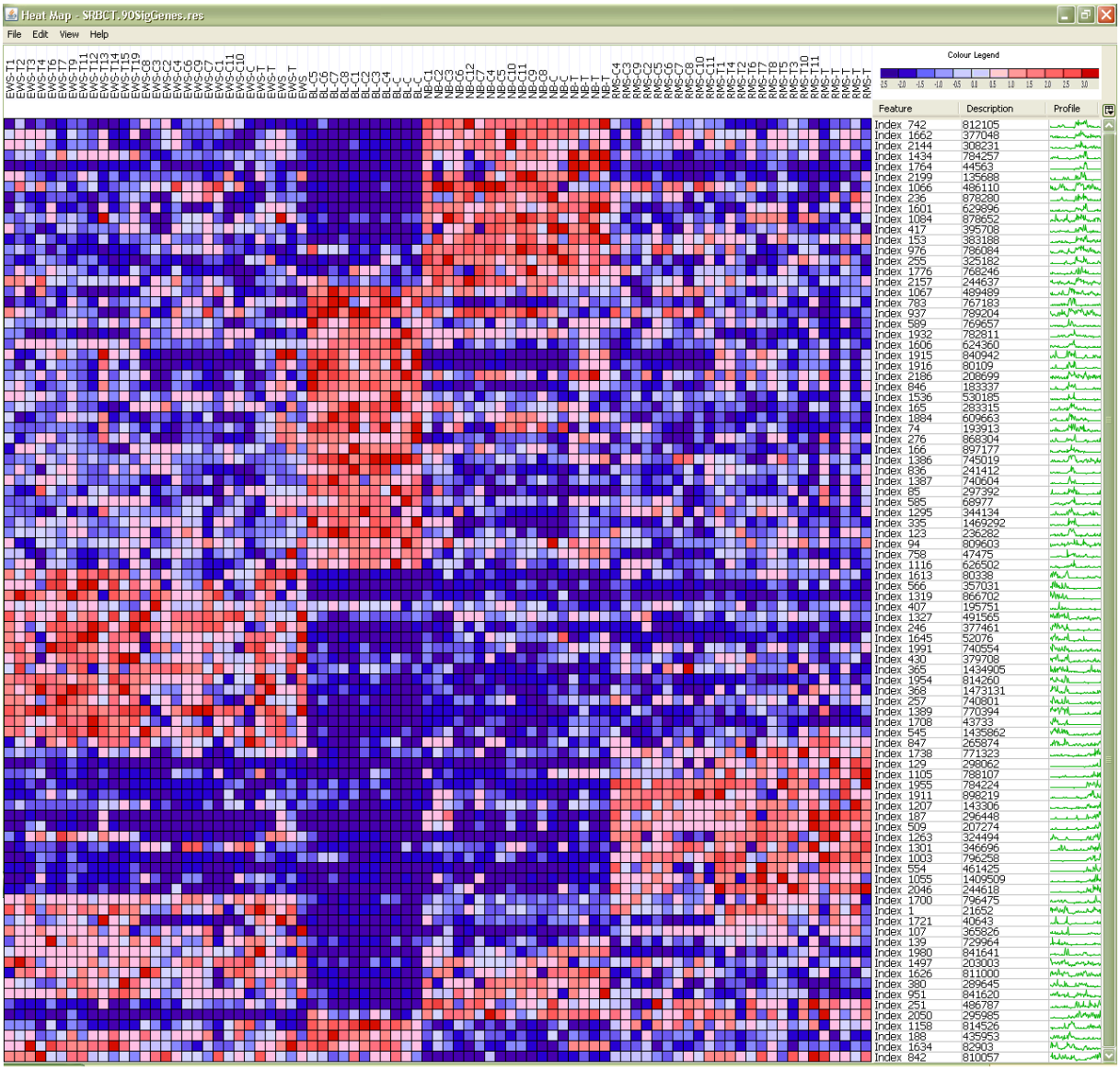


Figure 5.8: The heatmap of SRBCTs genes. The gene with high intensity (i.e. excited by the fluorescence dyes) denotes in red and the gene with low intensity denotes in blue.

The CD99 (MIC, Index_545) molecule is a very effective positive marker used to identify almost all EWS-related patients and it can help to rule out any differential diagnostic considerations when used as part of immunostaining techniques, i.e. antibody-based staining methods (Khoury, 2005). It was over-expressed in almost all EWS samples in Figure 5.8 and presented in 2 RMS samples. It is also a useful marker to identify lymphoblastic-related cancers including leukaemias and lymphomas (Kang and Dunphy, 2006). The gene GYG2 (Index_1708) was moderately expressed in EWS class in Figure 5.8, but not in any of the tumour classes in the data set which indicates that it is effectively more sensitive than CD99 in differentiating EWS tumour from the other SRBCT-related groups.

Genes in chromosome band 11p15 have generated a particular interest in medical studies. This region accom-

modates various types of growth factor genes which play an important role in controlling the regulation of various cellular processes in the body. Deregulation of the genes in this region is associated with over-growth and tumour predisposition syndrome (Smith et al., 2007). Genes SMPD1 (Index_139), LSP1 (Index_1207) and IGF2 (Index_187 and Index_509) were presented in RMS class in Figure 5.8 indicates that this region may play a role in the pathogenesis of RMS. The gene IGF2 has been reported in the pathogenesis of RMS by over-expression of the IGF2 level detected in ERMS subtype rather than in ARMS subtype (Makawita et al., 2009).

In addition to biomarkers, some identified genes may also be involved in some known cytogenetic differentiation among these four tumour groups. These genes are TNNT2 (Index_129) which may be associated with MYOG and CD45 in EWS/RMS tumours; KIF3C (Index_1434) and may play a role in the ALK mutation in NB tumour and LYN (Index_74) which may be associated with c-myc rearrangement in BL tumour.

The GANN prototype also identified some potential genes to be useful as biomarkers when used as a sole differential diagnostic marker or as part of the immunostaining technique in detecting a specific tumour group. These genes include ISG20 (Index_1387) in BL class; GAP43 (Index_1764) in NB class; SGCA (Index_1003), MYL4 (Index_554) and FGFR4 (Index_1955) in RMS class; FGCRT (Index_1389), CAV1 (Index_246), TNFAIP6 (Index_566), PTPN13 (Index_1319) and KDSR (Index_1954) in EWS class; as is indicated in Figure 5.8. There are also genes which were not expressed in certain tumour groups and these genes may be useful in ruling rule out any differential tumour considerations in the diagnosis process.

SGCA (Index_1003) is the muscle specific protein that associated with muscular dystrophy and insulin-like growth factor II (IGF2). High expression of SGCA in RMS class might be more indicative of the tissue origin rather than related to the molecular background of RMS (Lidén et al., 2002). While IGF2 gene (Index_187 and Index_509) is an already known oncogene associated with ERMS subtype cancer (Makawita et al., 2009).

FGFR4 (Index_1955) is a tyrosine kinase receptor that binds to fibroblast growth factor and to carry out the signal transduction to the intracellular environment in cellular proliferation, differentiation and migration (Pal et al., 2007). The activity of FGFR4 normally is undetectable (i.e. suppressed) in normal tissues, however, it becomes active when a tumour is formed. High expression of FGFR4 in RMS tumour was associated with advanced-stage cancer and poor survival (Taylor et al., 2009). In addition, over-expression of FGFR4 has been shown in various cancers (Pal et al., 2007), i.e. pituitary, prostate and thyroid.

CAV1 (Index_246) is a putative tumour suppressor gene involved in the regulation of signal pathway. Low expression of CAV1 has been associated to the development of colon cancer (Futschik et al., 2003). CAV1 has also been associated with prostate cancer (Lidén et al., 2002). Mutation in CAV1 is usually a sign of metastasis breast cancer (Bonuccelli et al., 2009).

The findings in this section showed that the GANN prototype able to identify genes that are not only related to tumour histogenesis, but also genes that may not normally be expressed in the corresponding tissue. This means that the GANN prototype does not detect genes exclusively associated with a single cancer type, however, it explores genes that are differentially expressed in multiple cancer types. The identified genes may provide new insights into the biology of the cancer. We also observed that the tanh based system extracted more biological significant genes than the other systems.

5.5 The differentially expressed genes in various precision levels

In previous sections, we discussed the extraction performance of each system based on the implications from two vital GA parameters that could affect the success of the system in extracting the most relevant genes from the respective data set. The integrity of the findings have been discussed based on the expected results from the synthetic data sets and the biology perspective.

In this section, the differentially expressed genes in three different fitness precision levels, i.e. 95%, 98% and 100% were examined. A comparison study based on these parameters was conducted using genes that were overlapped in all four systems with a specific GA parameter condition, i.e. population size 300 and fitness evaluation size 40000. Two tables, each representing a microarray data set, were produced. The complete list of genes extracted by each system in different precision levels was listed in Tables B.13 and B.14 in Appendix B. This design supports the objectives of our research theme stated in Section 1.4 on page 12.

Table 5.9: The summary list of overlapped ALL/AML genes with three different fitness precision levels.

Index	Accession Number	Symbol	Precision Level		
			100%	98%	95%
Index_760	D88422	CSTA	*	*	*
Index_804	HG1612-HT1612	MARCKSL1	*	*	*
Index_1685	M11722	DNTT	*	*	*
Index_1779	M19507	MPO	*	*	*
Index_1882	M27891	CST3	*	*	*
Index_2121	M63138	CTSD	*	*	*
Index_2288	M84526	CFD	*	*	*
Index_2354	M92287	CCND3	*	*	*
Index_2402	M96326	AZU1	*	*	*
Index_2642	U05259	CD79A	*	*	*
Index_3252	U46499	MGST1	*	*	*
Index_4328	X59417	PSMA6	*	*	*
Index_4377	X62654	CD63	*	*	*

Continued on Next Page...

Table 5.9 – *Continued*

Index	Accession Number	Symbol	Precision Level		
			100%	98%	95%
Index_4847	X95735	ZYX	*	*	*
Index_6041	L09209	APLP2	*	*	*
Index_1829	M22960	CTSA	*	*	
Index_6855	M31523	TCF3	*	*	
Index_1239	L07633	PSME1	*	*	
Index_1834	M23197	CD33	*	*	
Index_1928	M31303	STMN1	*	*	
Index_1962	M33680	CD81	*	*	
Index_4211	X51521	EZR	*	*	
Index_5501	Z15115	TOP2B	*	*	
Index_5772	U22376	MYB	*	*	
Index_6200	M28130	IL8	*	*	
Index_6201	Y00787	IL8	*	*	
Index_6376	M83652	CFP	*	*	
Index_4373	X62320	GRN		*	*
Index_412	D42043	RFTN1	*		
Index_668	D86967	EDEM1	*		
Index_1796	M20902	APOC1	*		
Index_4196	X17042	SRGN		*	
Index_758	D88270	IGL@		*	
Index_1745	M16038	LYN		*	

Table 5.9 shows the list of overlapped genes identified by all systems in the ALL/AML data set in three different fitness precision levels, i.e. 100%, 98% and 95%. There were 30 common genes obtained in 100% precision level, 31 common genes in 98% precision level and in 95% precision level, only 16 common genes were found. Amongst these genes, 15 genes were expressed in all levels. These genes include some biomarkers, i.e. DNTT, MPO and CD79A, that have been described in the previous section. The remaining genes were only expressed when a specific precision level was applied. Twelve genes were detected in higher precision levels, i.e. 98% and 100%, one gene (Index_4373) was found to be present in lower precision levels, i.e. 95% and 98%. These genes include CTSA (Index_1829), TCF3 (Index_6855), PSME1 (Index_1239), CD33 (Index_1834), STMN1 (Index_1928), CD81 (Index_1962), EZR (Index_4211), TOP2B (Index_5501), MYB (Index_5772), IL8 (Index_6200 and Index_6201) and CFP (Index_6376) that were expressed in the precision level 98% and above.

GRN (Index_4373)is a glycosylated peptide that has been cleaved into a variety of sections which may act differently on cell growth. It is important in normal development of cell and brain, wound healing and

tumourigenesis. Mutation in GRN has been associated to brain disease, such as Alzheimer and dementia, and has a crucial role in breast tumourigenesis (EntrezGene: GeneID 2896).

SRGN (Index_4196) encodes a haematopoietic cell granule proteoglycan protein, which was found to be associated with the macromolecular complex of granzymes and perforin, which may serve as a mediator of granule-mediated apoptosis (EntrezGene: GeneID 5552). LYN (Index_1745) is a tyrosine kinase that participate in the regulation of cell activation. The expression of LYN has been associated to B-cell lymphocytes (EntrezGene: GeneID 4067). SRGN, IGL@ (Index_758) and LYN have been identified with 98% fitness precision level.

The findings showed that a low expression but biologically significant gene could be identified by the GANN system. Although these lowly expressed genes are not directly associated in leukaemogenesis, however, they play an important role in normal development and deficiency of these genes may lead to immunodeficiency disease, which, in most cases, promotes tumourigenesis.

Table 5.10: The summary list of overlapped SRBCTs genes with three different fitness precision levels.

Index	Image Id.	Symbol	Precision Level		
			100%	98%	95%
Index_1	21652	CTNNA1	*	*	*
Index_85	297392	MT1L	*	*	*
Index_123	236282	WAS	*	*	*
Index_187	296448	IGF2	*	*	*
Index_236	878280	CRMP1	*	*	*
Index_246	377461	CAV1	*	*	*
Index_255	325182	CDH2	*	*	*
Index_335	1469292	PIM2	*	*	*
Index_417	395708	DPYSL4	*	*	*
Index_509	207274	IGF2	*	*	*
Index_545	1435862	CD99	*	*	*
Index_554	461425	MYL4	*	*	*
Index_742	812105	MLLT11	*	*	*
Index_783	767183	HCLS1	*	*	*
Index_836	241412	ELF1	*	*	*
Index_846	183337	HLA-DMA	*	*	*
Index_951	841620	DPYSL2	*	*	*
Index_976	786084	CBX1	*	*	*
Index_1003	796258	SGCA	*	*	*
Index_1055	1409509	TNNT1	*	*	*
Index_1084	878652	PMS2L12	*	*	*

Continued on Next Page...

Table 5.10 – *Continued*

Index	Image Id.	Symbol	Precision Level		
			100%	98%	95%
Index_1116	626502		*	*	*
Index_1158	814526	RBM38	*	*	*
Index_1207	143306	LSP1	*	*	*
Index_1319	866702	PTPN13	*	*	*
Index_1386	745019	EHD1	*	*	*
Index_1389	770394	FCGRT	*	*	*
Index_1434	784257	KIF3C	*	*	*
Index_1601	629896	MAP1B	*	*	*
Index_1606	624360	PSMB8	*	*	*
Index_1645	52076	NOE1	*	*	*
Index_1916	80109	HLA-DQA1	*	*	*
Index_1954	814260	KDSR	*	*	*
Index_1955	784224	FGFR4	*	*	*
Index_2046	244618	EST	*	*	*
Index_2050	295985	CDK6	*	*	*
Index_153	383188	RCVRN	*	*	
Index_1613	80338	SELENBP1	*	*	
Index_1327	491565	CITED2	*		*
Index_1884	609663	PRKAR2B	*		*
Index_165	283315	PGAM2	*		
Index_365	1473131	HOXB7	*		
Index_107	365826	GAS1		*	

For SRBCTs data set, as is indicated in Table 5.10, most of the identified genes, i.e. 36 genes, were presented in all levels of fitness precision, including 1 undefined gene and some biomarkers, such as CD99, IGF2 and LSP1. These genes may play a role as general markers for detecting small round undifferentiate cells.

Two genes (Index_153 and Index_1613) were detected in higher precision levels, i.e. 98% and 100%, but did not appear in a lower precision level. Index_1327 and Index_1884 were found in both the highest (100%) and the lowest (95%) precision levels, but not in 98%. This would indicate that these genes were provoked by other abnormal genes. However, further molecular analysis on these genes should be performed to validate their roles in tumourigenesis pathology.

Meanwhile, GAS1 (Index_107) was detected in 98% precision level. GAS1 is a putative tumor suppressor gene involved in cell arrest and can induce apoptosis when it is over-expressed in different cell lines. It contains the expected properties as a melanoma tumor suppressor: suppression of metastasis in a spontaneous metastasis assay, promotion of apoptosis following dissemination of cells to secondary sites and down-regulation in

human melanoma metastasis (EntrezGene: GeneID 2619).

To conclude, this section demonstrated the presence of differentially expressed genes in different precision levels which could contribute to an early prognosis of tumourigenesis. The findings show the potential of some genes which might be provoked by the presence of certain genes. This information could be useful for therapeutic and biological diagnostic to prevent the development of tumour pathology.

5.6 Raw microarray data set Versus Normalised microarray data set

Referring to the problems stated in Section 1.2 on page 7, most of the oligonucleotide-based data sets were generally preprocessed before being analysed by the computer algorithm. Table C.1 in Appendix C shows some preprocessing techniques have been used in the ALL/AML data set. We argued that the quality of the data set could be compromised by the normalisation process. To validate our statement, a comparison study based on the raw and the normalised ALL/AML data sets was performed on each system in the population size 300 and the fitness evaluation size 40000. A tailored C++ program is coded to scale down the gene values to the [0, 1] interval, based on the maximum and the minimum values of a gene within the data set. The equation of max-min normalisation is as follows:

$$\bar{x_i} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{5.1}$$

This max-min normalisation is being used in this study as it has been reported effective for the ensemble ANN classifiers by Cho and Won (2007) in dealing the ALL/AML data set.

Table 5.11: The summary of the genes extracted from the raw and the normalised ALL/AML data sets.

System	Raw data set	Normalised data set	No. of consistent genes	No. of selected genes solely presented in the raw data set
Sigmoid-based	45	86	35	10
Linear-based	55	92	47	8
Tanh-based	46	77	33	13
Threshold-based	47	94	41	6

Tables 5.11 and 5.12 present the summary of our findings and the complete list of the extracted genes is presented in Table B.15 in Appendix B. The results show that by normalising the data set, the number of genes identified by each system was tremendously increased. This is indicated by the increased number of

Table 5.12: The processing time (in seconds) spent in the raw and the normalised ALL/AML data sets.

System	Raw data set	Normalised data set
Sigmoid based	8809	4007
Linear based	1951	1482
Tanh based	1644	6265
Threshold based	3036	1596

genes found in the normalised ALL/AML data set as compared to the genes identified in the raw ALL/AML data set in Table 5.11. The normalisation process scaled down the magnitude of the gene values into a specific interval, as a result, the highly differentially expressed genes were forced to be compressed to meet the normalisation criteria and the significance of these genes was not obvious to the system. This is indicated by a significant low selection frequency when similar genes were being selected in the normalised data set as being selected in the raw data set (see Table B.15 in Appendix B). The chances to select low expression but high biological relevant genes become impossible in the normalised data set as these genes have been suppressed by other genes which have a higher expression values.

We observed a significant improvement in the processing time spent by the sigmoid, the linear and the threshold based systems when the data set was normalised. However, the tanh based system required an intensive processing time when analysing this normalised data set. This is because the tanh based system adopting larger interval range, i.e. [-1, 1], in the selection process, whereas this normalised data set has a smaller scaling range that is not appropriate for standard tanh computation.

The findings in this section has demonstrated the deficiency of using the normalisation process in the ALL/AML data set, due to the quality of the data altered by the normalisation technique and, subsequently, the integrity of the results was compromised.

We would like to draw the attention of researchers to the implication of normalisation techniques in the quality of microarray data. The application of normalisation technique in the microarray data without understanding the data itself could severely compromised the integrity of the results. This is due to microarray data being much more complicated than the ordinary real-world data in the sense that it contains a multivariant of cancer subtypes within a known cancer group and some of these cancer subtypes might associated with other disease. Furthermore, the oligonucleotide microarray data are highly skewed with negative data values and consequently, the biological relevant, however, lowly expressed genes in the data set may be 'buried' by the highly expressed ones in the data set. Therefore, the normalisation is usually expected in the oligonucleotide microarray data to reduce the value skewness within the data. The improper use of normalisation techniques could lead to an overly optimistic result and redundant information.

5.7 The significance of the extracted bioassay attributes

In the previous sections, we discussed the GANN performance to handle data with high feature dimension, sample scarcity and complex feature interaction. Based on these findings, we observed that the tanh based system is, amongst all the systems, the most effective system to extract the most significant features from the biology perspective.

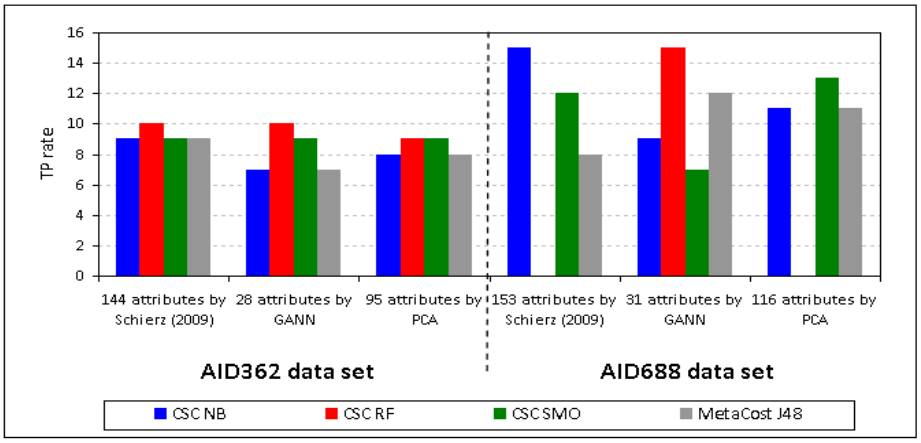
In this section, we examine the selection performance of the tanh based system to handle bioassay data characterised by low feature dimension, feature-independent and highly imbalanced between the positive and the negative samples. An experiment was conducted using the bioassay data sets based on the population size 300 and the fitness evaluation size 30000. The completeness of the findings is evaluated using four cost-sensitive classifiers constructed in the WEKA environment (see Section 4.1.2 on page 95 for parameter settings) and is compared with the original work reported by Schierz (2009). The data sets have been split into an 80% training set and a 20% test set, as recommended by Schierz. For AID362 data set, the training set contains 3423 compounds, i.e. 48 active and 3375 inactive; and the test set contains 856 compounds, i.e. 12 active and 844 inactive. For AID688 data set, the training set has 21751 compounds, i.e. 198 active and 21553 inactive; and 5438 compounds, i.e. 50 active and 5388 inactive.

The top 20% of the attributes from each data set were selected by GANN prototype. To evaluate the generalisation performance of the prototype, these selected attributes were trained by the identical set of cost-sensitive classifiers (CSC) as reported in the original study (Schierz, 2009), i.e. Naive Bayes (CSC NB), Support Vector Machine (CSC SMO), C4.5 tree (MetaCost J48) and Random Forest (CSC RF), with a 10-fold cross-validation procedure. The significance of the attributes were then validated using the independent test set.

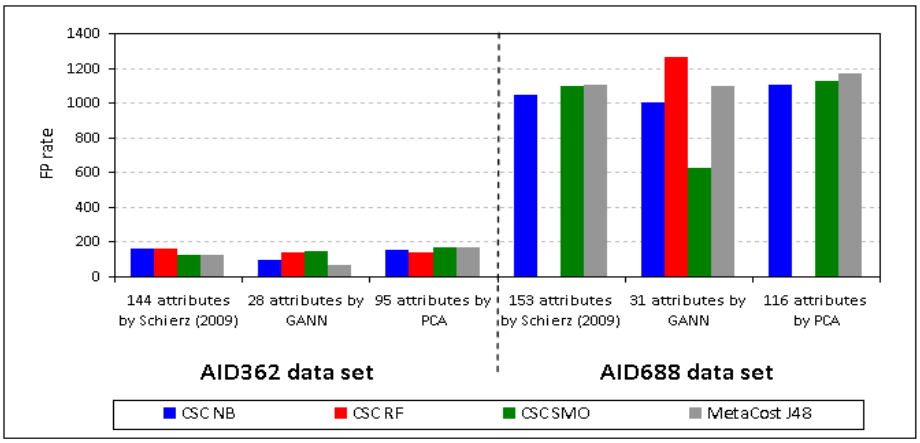
Standard classifiers usually assume equal weighting of the classes in the data set, in the sense that misclassifying class A has similar consequences as misclassifying class B. As a result, most standard classifiers are unable to predict a minority class in the bioassay data sets (Maloof, 2003; Schierz, 2009) due to highly imbalanced data between active and inactive compounds in the data set. This led to the development of cost-sensitive classifiers, which can assign different costing for different weighting classes in imbalanced data sets (Domingos, 1999; Elkan, 2001; Maloof, 2003).

A trial experiment, based on the network size 10-5-2 using the AID362 data set was performed and the results showed a poor performance on the system. This was due to the large data size of bioassay data (4279-27189 samples) when compared to microarray data (72-83 samples), thus, we increased the network size of GANN to 20-10-2, while the remaining parameters remain unchanged. Figure 5.9 shows the summary of the results based on the independent test set of the data sets and the list of the selected attributes by

GANN is presented in Table B.16 in Appendix B. We have also conducted the experiments using principal component analysis (PCA).



(a) The true positive (TP) rate of the bioassay data sets with under or approximately a 20% false positive (FP) rate.



(b) The false positive (FP) rate of the bioassay data sets.

Figure 5.9: The classification results based on the independent test set in the bioassay data sets.

There has been no significant decrease in classification performance of the cost-sensitive classifiers, as showed in Figure 5.9a. Using the 28 attributes selected by GANN in the AID362 data set, a CSC RF has produced better results than using the entire 144 attributes reported by Schierz (2009) and the 95 attributes selected by PCA. There has been a slight decrease in performance when using CSC NB and CSC SMO, but all other results are comparable. Furthermore, using the 28 attributes selected by GANN, a significant decrease on the number of false positive (FP) rate was observed in all classifiers, except a CSC SMO, as is presented in Figure 5.9b.

For AID688 data set, as showed in Figure 5.9a, using the 31 attributes selected by GANN, a MetaCost J48 tree has performed better than using the entire 153 attributes reported by Schierz (2009) and the 116 attributes selected by PCA. Surprisingly, the CSC RF which was unable to run using neither the whole

data set nor the 116 attributes selected by PCA, due to the size of the data set ($\sim$ 27000 compounds), has produced good results using the attributes selected by GANN. However, there has been a decrease in performance using CSC NB and CSC SMO for attributes selected by GANN when compared to using the entire attributes and the 116 attributes selected by PCA. Even so, all cost-sensitive classifiers have the lowest FP using the 31 attributes selected by GANN (see Figure 5.9b).

The findings in this section demonstrate the efficiency of the GANN prototype to handle a large, imbalanced data set. The results show a comparable better performance on the GANN prototype and the entire attributes in the data set and the PCA. The advantage of using GANN to select attributes over using the entire attributes and the PCA is that GANN enables computationally effective algorithms, such as Random Forest and Classification Tree, to be implemented in a larger data set with a high success rate. Considering the GANN prototype only implemented 20% of the attributes from the data set, the classification performance of cost-sensitive classifiers has not been sacrificed. This shows that the GANN prototype has successfully identified the most significant attributes needed to discriminate between the active and the inactive compounds. The only downside of the GANN prototype is that a computationally intensive processing time required for larger data set.

5.8 Summary

In this chapter, we examined the performance of the GANN model as a feature extraction. The prototype was implemented to extract informative features (genes and attributes) from six data sets, comprising two synthetic data, two microarray data and two bioassay data. The results can be summarised as follows:

- The linear based system is able to explore the potential genes more effectively than the other three systems and is very much processing cost effective as compared to the remaining three systems. However, this system also induced a high number of low interest genes in the subject of study.
- The sigmoid based system is also able to efficiently explore the potential genes. However, this system requires a more intensive processing cost than the other three systems.
- The threshold based system is lack of the stability factor in extracting consistent genes and it is sensitive to data distribution.
- The tanh based system is, amongst the four systems, the most effective system to extract the most informative genes from the data sets.
- All systems have a significant improvement on their performance with the population sizes 200 to 300 and the fitness evaluation sizes, ranging from 20000 to 40000.

- The identification of informative genes that were lower expressed in the data set can be achieved with a low fitness precision level. These genes may be useful for therapeutic and biological diagnostic to prevent the development of a tumour.
- The improper use of normalisation technique and a lack of understanding on the microarray data could compromise the integrity of the results.
- The tanh based GANN system has produced better, or at least comparable, results in a large and imbalanced bioassay data set. However, the only downside of the GANN prototype is that a computationally intensive processing time required for larger data set.

In the next chapter, we conclude our research by revising our contributions to the bioinformatics field, the revision on our methodology and suggests the trend of the future research. The chapter will concluded with the overall achievement of this thesis.

Chapter 6

Conclusion and Future Works

This chapter draws conclusions on our work reported in this thesis . A summary of the major contributions of the research is provided and suggestions for possible further research areas.

This chapter contains five sections. Section 6.1 provides conclusions of the thesis. Section 6.2 summarises the major contributions for this research. Section 6.3 presents the areas that have been omitted in this thesis. Section 6.4 indicates the limitations of our research and suggests several interesting avenues for further work to extend our research. Finally, Section 6.5 concludes the thesis with the overall achievements of the research.

6.1 Conclusions of the Thesis

This thesis presents an intelligent extraction approach via the hybridisation of GAs and ANNs to identify informative features in the bioinformatics field. Our research emphasises the Ockham's Razor principle to extract features from high dimensional data and imbalanced data. Our approach is one of the very rare feature extraction facilities that exploits the features of GAs and ANNs as a feature extraction and works efficiently in two distinct types of data in the field. The main conclusions drawn from this thesis to answer the research questions are given below.

1. Using the simplest parameter settings in both GA and ANN, primary features can be extracted from the microarray data and these features show their biological significance on the tumourigenesis pathway.
2. The important genes which were lower expressed in the microarray data can be extracted with a lower fitness precision level.

Some important conclusions derived from this the experimental study are given as below.

1. The use of tanh activation function in ANNs, amongst four ANN functions, is the most effective function to be used to compute the fitness function of a GA.
2. The GA has the best performance when the population size is between 200 to 300 and the fitness evaluation sizes, ranging from 20000 to 40000.
3. The improper use of normalisation technique and a lack of understanding on the microarray data could compromise the integrity of the gene selection results. This is because the normalisation technique homogenised the magnitudes of the primary features with the secondary or the least significant features in the data, and consequently, the chance of discovering primary features become narrowed.

6.2 Summary of Contributions

The main goal of this thesis was to devise a more effective way for extracting informative features from high dimensional data using GAs and ANNs. This goal leads to three major contributions, which are reviewed in related literature, the solution for feature extraction, the prototype implementation and evaluation. The summary of these contributions is as follows:

6.2.1 The review of related literature

Two domains of literature pertaining to the biology and computer fields were reviewed. The existing biology literature shows that there are two dominant types of microarrays: oligonucleotides and cDNA, each of which requires different techniques to be used in the microarray production. The cDNA microarray usually requires a two-phase in normalisation steps, i.e. the pre-normalisation in the fluorescent labelling process and the post-normalisation in the imaging process. The oligonucleotide microarray requires only post-normalisation in the imaging process. Both types of microarray data contain a high dimension of noisy genes, which require the use of computer algorithm to analyse the data.

The existing computer literature shows that there are two dominant ways for analysing microarray data: predicting the accuracy of the samples based on the known cancer group in the data; and discovering new cancer groups from the known cancer group in the data. Both ways expose the immaturity of the gene extraction area. Thus, in this thesis, we devised an efficient solution for extracting informative genes based on the application of hybrid GAs and ANNs, as the previous research restricting the utility of the selection method to retain the effectiveness of the classification method and the presentation of gene extraction based upon the hybrid GA/ANN is rare. The rarity of this solution in feature extraction is mainly due to the ill-conceived hypothesis in the existing works in dealing with microarray data. Another reason is because both GAs and ANNs are not model transparent which lacks the step-by-step logic to explain the interaction

between genes in the model. Thirdly, the ANN is commonly used as a classifier to classify samples and the GA is usually used to optimise the parameter of the ANN. Promising results based on the incorporation of GA into the ANN were reported in many discipline areas. Thus, researchers are not keen to find outcomes when the ANN is incorporated into a GA.

Our work has incorporated the ANN into a GA using as minimal parameter setting as possible to avoid the over-fitting problem which normally arises in ANNs. The ANN in our model is used as a fitness generator to compute the GA fitness function.

The existing literature identifies several problems as follows:

- The lack of understanding of microarray data result in an ambiguity of the objectives of the study.
- The risk of over-fitting and biased underestimates of the error rate due to the misuse of a valid mechanism and resusbstitution estimation in the classification process.
- The lack of supporting evidence in the declaration of new prediction models due to the misuse of a valid mechanism and resusbstitution estimation.
- The researchers being not aware of the influence of the model complexity to the prediction results which result in model over-fitting.
- The researchers being not aware of the influence of the data preprocessing in finding the relevant information for the problems.

Information on these problems can be found in Sections 1.1 and 1.2 in Chapter 1 and Section 2.2 in Chapter 2.

6.2.2 The solution for feature extraction

Identifying a solution for feature extraction is the most important contribution of this research, as the selection method is always needed to reduce the number of noisy information from microarray data. In our solution, we utilised the universal computation power of ANNs and the evolutionary ability of GAs to extract informative genes from a specific microarray data sets. Three main steps in our feature extraction model are: (a) initialising a population of potential members to the problem, i.e. GA chromosomes; (b) computing fitness values for each member in the population using a 3-layer feedforward ANN; and (c) evaluating the fitness of each member in the population using GA crossover and mutation operators.

The capabilities of our solution are as follows:

- The order of genes selected based on the selection frequency of genes, i.e. the number of times that the gene is selected, and the fitness accuracy of the gene subset, i.e. the number of times that the sample is correctly labelled in the class for the selected gene, is calculated.
- The fitness accuracy and the number of fitness evaluations for each GA cycle, i.e. GA generation, are preserved.
- The flexibility to alter parameters, such as types of activation function to be used to compute fitness values, the population size, the fitness evaluation size, the network size, the fitness precision level and the gene list corresponds to a specific fitness accuracy to be displayed.
- The simplicity of the model in which only the fundamental parameter settings are applied which reduce the possible risks arising from the complex model and also provides generalisability in handling multiple types of data structures, i.e. microarray data and bioassay data, in the bioinformatics field.
- Retaining all highly fitted members for the next generation, excepting the least fit member which is replaced by the new member in the next generation.

6.2.3 The prototype implementation and Evaluation

The prototype of the feature extraction solution has been implemented using a C++ programming language in LINUX environment to realise the proposed techniques. This prototype helps to validate our approach and shows the possibility of using ANN as a fitness generator for a GA, as well as extracting informative features from the high dimensional data and in the highly imbalanced data with different data representations. The prototype provided a fundamental basis for conducting our experimental study.

The performance of the prototype has been evaluated via experimental study which serve the following purposes:

- The performance of prototype, each with a different ANN activation function, to extract informative genes from the microarray data.
- The minimal sizes of GA population and fitness evaluation for efficient marker identification.
- The ability of prototype to handle different platform of microarray data.
- The ability of prototype to extract important genes that were lower expressed in microarray data set.
- The ability of prototype to handle a highly imbalanced data with multiple data representations.

6.3 Areas that are not explored in this thesis

There are several areas which were not explored in this thesis. These areas include:

- The mathematical proof on the logic of the activation function to calculate fitness values for each member in the population.
- The implication of various data partition sizes in marker identification.
- The biological validation on the identified genes using RT-PCR assay or FISH analysis.
- The exploration in chemoinformatics literature and the challenges in the field.
- The exploration in pattern recognition literature for handling large and imbalanced data sets.

6.4 Limitations of this research and Further work

Our model has some limitations that should be improved in the future. These limitations are to handle data with multiclasses more efficiently, to reproduce identical sets of extracted features and to reuse the selection results.

- Although we have tested our model with two multiclass data sets in this thesis and the findings from these data sets are promising, however, a lower fitness confidence level of the identified genes was achieved compared to the fitness confidence level of the selected genes from the binary class data sets. With the increased number of classes and different data representation, we may need to re-configure the parameters in the model for better performance.
- Since our model did not preserve information of the network weights after each repetition run, the reproduction of the identical set of genes is impossible, even though identical sets of parameter settings were applied. With no preserved information of the network weights, we are unable to generate similar fitness scores for the identical sets of chromosomes. When we consider the computational cost of our model, with no preservation on the network weights used in the previous run, our model, in fact, has a low computational cost as our model has less tasks to perform. Furthermore, each run represents a new start on the GA and the ANN process in our model which can avoid the model being over-fitted by prior information in the model.
- Our model produced only the list of identified genes ordered by its selection frequency and the total number of correctly labelled samples after the completion of the maximum number of repetition runs. As a result, the gene results cannot be reused in a different number of repetition runs.

For further work, we would like to explore the follows:

1. The implication of data partition on the significance of the extracted genes

   One of the characteristics of microarray data is complex gene interactions, meaning that the number of interacted genes are not fixed. Thus, by partitioning the data into different sizes and/or different number of equal portions, different results may produced. Therefore, we would like to investigate the potentiality of data partitioning in the effect of significant genes identified by our model.

2. Configurating the parameter of GANN

   The current model using the fundamental configuration of the GA and the ANN. Although, promising results are achieved, however, the level of fitness confidence for the selected genes from multiclass data and data with different representation is low. This may be due to single-point crossover and binary tournament selection being applied in our model. Thus, we would like to explore the potentiality of different crossover operators and different variant of tournament selection techniques in improving the fitness performance of the model in multiclass data and in imbalanced data.

3. Program new Functions

   The existing model did not preserve the genes extracted in each repetition run. Thus, the selection results cannot be reused when a different number of repetition runs is needed. In addition, the existing model unable to associate the extracted genes to the respective class in the data set without the assistance of an external tool. To circumvent the problem, we will write new program functions to specifically preserve the list of genes extracted in each repetition run and to calculate the fitness accuracy strength of the gene and, without increase the processing time in the current model.

4. Designing a User Interface

   The current model has no proper user interface. All the changes of the parameter settings have to be made directly in the C++ source code. This could increase the risk of logic errors that may be made by the user which could yield undetected flaws in the results. Thus, we will program the window-based interface in which the user can alter the parameter values and choose the report format to be displayed.

6.5 The overall achievement of the thesis

The overall achievement of this thesis can be summarised as follows:

- The first achievement is the presentation of a solution for extracting highly significant features from high dimensional data sets.

- The second achievement is the way we hybrid GAs and ANNs which can utilised the minimal parameter settings of the techniques to derive highly informative sets of genes.
- The third achievement is the experimental study conducted on four activation functions of ANNs in different population size and GA evaluation, which has never been conducted before.
- The fourth achievement is the interpretation of the extracted genes from the biological perspectives and its association to the disease of interest.
- The fifth achievement is the identification of the important genes appearing in different tumour development stages and the precaution steps on these genes may prevent the spread and the growth of a tumour.
- The sixth achievement is arguably novel, that is, our model is able to extract highly informative features from multiple types of bioinformatics data, including microarray data that was characterised with high gene dimension, sample scarcity and complex gene interaction; and bioassay data featured with enormous compound size and highly imbalanced.
- Lastly, this thesis also demonstrate the practicality and the reliability of the solution through the prototype.

Overall, we believe that our research makes advances in the feature extraction area in the bioinformatics field. Our hybrid GANN method has proved that the selection on both the biological-relevant and statistical-significant genes can be achieved using the raw, unprocessed microarray data. By deliberately not emphasising on the quality of ANNs, true marker genes can be extracted by our model. Furthermore, our approach also demonstrated its efficiency in dealing with bioassay data, which brings a new perspective in finding the association between marker genes and pharmaceutical drug that is used to control the progression of cancer disease. We believe that the findings presented in this thesis will draw more attention to the area and stimulate more research in this field.

List of References

A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E. Marti, T. Moore, J.Jr. Hudson, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, J.O. Armitage, R. Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O. Brown, and L.M. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

C. Ambroise and G.J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *PNAS*, 99(10):6562–6566, 2002.

I.P. Androulakis. Selecting maximally informative genes. *Computers and Chemical Engineering*, 29(3): 535–546, 2005.

D.A. Arber, D.S. Snyder, M. Fine, A. Dagis, J. Niland, and M.L. Slovak. Myeloperoxidase immunoreactivity in adult acute lymphoblastic leukemia. *Am J Clin Pathol*, 116(1):25–33, 2001.

J. Arifovic and R. Gencay. Using genetic algorithms to select architecture of a feedforward artificial neural network. *Physica A*, 289(3):574–594, 2001.

S.A. Armstrong, J.E. Staunton, L.B. Silverman, R. Pieters, M.L. den Boer, M.D. Minden, S.E. Sallan, E.S. Lander, T.R. Golub, and S.J. Korsmeyer. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*, 30(1):41–47, 2002.

M.H. Asyali, D. Colak, O. Demirkaya, and M.S. Inan. Gene expression profile classification: A review. *Current Bioinformatics*, 1:55–73, 2006.

R.P. Auburn, D.P. Kreil, L.A. Meadows, B. Fischer, S.S. Matilla, and S. Russell. Robotic spotting of cDNA and oligonucleotide microarrays. *Trends Biotechnology*, 23(7):374–379, 2005.

M. Barletta, A. Gisario, and S. Guarino. Modelling of electrostatic fluidized bed (EFB) coating process using artificial neural networks. *Eng. Appl. Artif. Intell.*, 20(6):721–733, 2007.

D. Beasley, D.R. Bull, and R.R. Martin. An overview of genetic algorithms: Part I, fundamentals. *University Computing*, 15(2):58–69, 1993.

R.G. Beiko and R.L. Charlebois. GANN: Genetic algorithm neural networks for the detection of conserved combinations of features in DNA. *BMC Bioinformatics*, 6(36), 2005.

M. Bernsteina, H. Kovarb, M. Paulussenc, R. Lor Randalld, A. Schucke, L.A. Teotf, and H. Juergensgg. Ewings sarcoma family of tumors: Current management. *The Oncologist*, 11(5):503–519, 2006.

V. Bevilacqua, G. Mastronardi, and F. Menolascina. Genetic algorithm and neural network based classification in microarray data analysis with biological validity assessment. In D-S Huang, K.L, and G.W. Irwin, editors, *ICIC'06: Computational Intelligence and Bioinformatics, International Conference on Intelligent Computing, proceedings, Part III*, volume 4115 of *Lecture Notes in Computer Science*, pages 475–484. Springer, 2006a.

V. Bevilacqua, G. Mastronardi, F. Menolascina, A. Paradiso, and S. Tommasi. Genetic algorithms and artificial neural networks in microarray data analysis: a distributed approach. *Engineering Letters - Special Issue on Bioinformatics*, 13(3):335–343, 2006b.

G. Bloom, I.V. Yang, D. Boulware, K.Y. Kwong, D. Coppola, S. Eschrich, J. Quackenbush, and T.J. Yeatman. Multi-platform, multi-site, microarray-based human tumor classification. *American Journal of Pathology*, 164(1):9–16, 2004.

T.H. Bø and I. Jonassen. New feature subset selection procedures for classification of expression profiles. *Genome Biology*, 3(4):research00, 2002.

G. Bonuccelli, M.C. Casimiro, F. Sotgia, C. Wang, M. Liu, S. Katiyar, J. Zhou, E. Dew, F. Capozza, K.M. Daumer, C. Minetti, J.N. Milliman, F. Alpy, M.C. Rio, C. Tomasetto, I. Mercier, N. Flomenberg, P.G. Frank, R.G. Pestell, and M.P. Lisanti. Caveolin-1 (p132l), a common breast cancer mutation, confers mammary cell invasiveness and defines a novel stem cell/metastasis-associated gene signature. *Am J Pathol.*, 174(5):1650–1662, 2009.

A.L. Boulesteix, C. Strobl, T. Augustin, and M. Daumer. Evaluating microarray-based classifiers: An overview. *Cancer Informatics*, 6:77–97, 2008.

S. Brenner, M. Johnson, J. Bridgham, G. Golda, D.H. Lloyd, D. Johnson, S. Luo, S. McCurdy, M. Foy, M. Ewan, R. Roth, D. George, S. Eletr, G. Albrecht, E. Vermaas, S.R. Williams, K. Moon, T. Burcham, M. Pallas, R.B. DuBridge, J. Kirchner, K. Fearon, J i. Mao, and K. Corcoran. Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nature Biotechnology*, 18(6): 630–634, 2000.

N. Brown. Chemoinformatics—an introduction for computer scientists. *ACM Computing Surveys*, 41(2): 1–38, 2009.

H. Bruchova, M. Kalinova, and R. Brdicka. Array-based analysis of gene expression in childhood acute lymphoblastic leukemia. *Leukemia Research*, 28:1–7, 2004.

F. Busetti. Genetic algorithms overview. Available from: http://citeseer.nj.nec.com/464346.html [Accessed 31 October 2007], 2001.

H. Cartwright. *Using artificial intelligence in chemistry and biology: A practical guide*, chapter Artificial Neural Networks, pages 9–49. CRC Press, Taylor & Francis Group, 2008a.

H. Cartwright. *Using artificial intelligence in chemistry and biology: A practical guide*, chapter Evolutionary Algorithms, pages 113–172. CRC Press, Taylor & Francis Group, 2008b.

Q-R. Chen, G. Vansant, K. Oades, M. Pickering, J.S. Wei, Y.K. Song, J. Monforte, and J. Khan. Diagnosis of the small round blue cell tumors using multiplex polymerase chain reaction. *Journal of Molecular Diagnostics*, 9(1):80–88, 2007.

Y.H. Chen, Y.M. Tang, H.Q. Shen, H. Song, S.L. Yang, S.W. Shi, B.Q. Qian, W.Q. Xu, and B.T. Ning. The expression of CD19 in 210 cases of childhood acute leukemia and its significance. *Zhonghua Er Ke Za Zhi*, 42(3):188–191, 2004.

J. Cheng and Q.S. Li. Reliability analysis of structures using artificial neural network based genetic algorithms. *Comput. Methods Appl. Mech. Engrg.*, 197(45-48):3742–3750, 2008.

M-Y. Cheng and C-H. Ko. A genetic-fuzzy-neuro model encodes FNNs using SWRM and BRM. *Engineering Applications of Artificial Intelligence*, 19(8):891–903, 2006.

G. Chetty and M. Chetty. Multiclass microarray gene expression analysis based on mutual dependency models. In V. Kadirkamanathan et al., editor, *PRIB'09: 4th IAPR International Conference on Pattern Recognition in Bioinformatics, proceedings*, volume 5780 of *Lecture Notes in Bioinformatics*, pages 46–55, 2009.

H.S. Cho, T.S. Kim, J.W. Wee, S.M. Jeon, and C.H Lee. cDNA microarray data based classification of cancers using neural networks and genetic algorithms. In *Nanotech'03: Nanotechnology Conference and Trade Show, proceedings*, volume 1, 2003a.

J-H. Cho, D. Lee, J.H. Park, and I-B. Lee. New gene selection method for classification of cancer subtypes considering within-class variation. *FEBS letters*, 551(1-3):3–7, 2003b.

S-B. Cho and H-H. Won. Machine learning in dna microarray analysis for cancer classification. In Y-P.P. Chen, editor, *APBC'03: P1st Asia-Pacific bioinformatics conference on Bioinformatics, proceedings*, volume 19, pages 189–198. Australian Computer Society, 2003.

S.B. Cho and H-H. Won. Cancer classification using ensemble of neural networks with multiple significant gene subsets. *Applied Intelligence*, 26(3):243–250, 2007.

W. Chu, Z. Ghahramani, F. Falciani, and D.L. Wild. Biomarker discovery in microarray gene expression data with gaussian processes. *Bioinformatics*, 21(16):3385–3393, 2005.

J.A. Cruz and D.S. Wishart. Applications of machine learning in cancer prediction and prognosis. *Cancer Informatics*, 2:59–78, 2006.

A.C. Culhane, G. Perrière, E.C. Considine, T.G. Cotter, and D.G. Higgins. Between-group analysis for microarray data. *Bioinformatics*, 18(12):1600–1608, 2002.

A.R. Dabney. Classification of microarrays to nearest centroids. *Bioinformatics*, 21(22):4148–4154, 2005.

H.K. Dadi, A.J. Simon, and C.M. Roifman. Effect of CD3delta deficiency on maturation of alpha/beta and gamma/delta t-cell lineages in severe combined immunodeficiency. *N Engl J Med*, 349(19):1821–1828, 2003.

K.A. DeJong. Learning with genetic algorithms: An overview. *Machine Learning*, 3(2-3):121–138, 1988.

K.A. DeJong and W.M. Spears. An analysis of the interacting roles of population size and crossover in genetic algorithms. In H-P. Schwefel and R. Männer, editors, *PPSN'91: 1st Workshop on Parallel Problem Solving from Nature, proceedings*, volume 496 of *Lecture Notes in Computer Science*, pages 38–47. Springer, 1991.

J.M. Deutsch. Algorithm for finding optimal gene sets in microarray prediction. Available from: http://stravinsky.ucsc.edu/josh/gesses, August 2001. [Accessed 31 October 2007].

J.M. Deutsch. Evolutionary algorithms for finding optimal gene sets in microarray prediction. *Bioinformatics*, 19(1):45–52, 2003.

R. Díaz-Uriarte and S.A. de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3, 2006.

B. Djavan, M. Remzi, A. Zlotta, C. Seitz, P. Snow, and M. Marberger. Novel artificial neural network for early detection of prostate cancer. *J Clin Oncol.*, 20(4):921–929, 2002.

P. Domingos. Metacost: a general method for making classifiers cost-sensitive. In *KDD'99: Fifth ACM SIGKDD international conference on Knowledge discovery and data mining, proceedings*, pages 155–164, 1999.

S. Dudoit, J. Fridlyand, and T.P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. Technical report 576, Department of Statistics, University of California, Berkeley, June 2000.

S. Dudoit, J. Fridlyand, and T.P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.

A. Dupuy and R.M. Simon. Critical review of published microarray studies for cancer outcome and guidelines on statistical analysis and reporting. *JNCI Journal of the National Cancer Institute*, 99(2):147–157, 2007.

R. Dybowski, P. Weller, R. Chang, and V. Gant. Prediction of outcome in critically ill patients using artificial neural network synthesised by genetic algorithm. *THE LANCET*, 347(9009):1146–1150, 1996.

B.L. Ebert and T.R. Golub. Genomic approaches to hematologic malignancies. *Blood*, 104(4):923–932, 2004.

C. Elkan. The foundations of cost-sensitive learning. In *IJCAI'01: Seventeenth International Joint Conference on Artificial Intelligence, proceedings*, pages 973–978, 2001.

M.H. Fatemi. Prediction of ozone tropospheric degradation rate constant of organic compounds by using artificial neural networks. *Analytica Chimica Acta*, 556(2):355–363, 2006.

T. Froese, S. Hadjiloucas, R.K.H. Galv ao, V.M. Becerra, and C.J. Coelho. Comparison of extrasystolic ecg signal classifiers using discrete wavelet transforms. *Pattern Recogn. Lett.*, 27(5):393–407, 2006.

M.E. Futschik, A. Reeve, and N. Kasabov. Evolving connectionist systems for knowledge discovery from gene expression data of cancer tissue. *Artificial Intelligence in Medicine*, 28:165–189, 2003.

J. Gasteiger. Chemoinformatics: a new field with a long tradition. *Analytical and Bioanalytical Chemistry*, 384(1):57–64, 2006.

D.E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*, chapter Some Applications of Genetic Algorithms, pages 59–88. Addison Wesley Longman, 1989.

D.E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G.J.E. Rawlins, editor, *FOGA'90: 1st Workshop on Foundations of Genetic Algorithms, proceedings*, pages 69–93. Morgan Kaufmann, 1990.

T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–536, 1999.

S.M. Gonia, S. Oddone, J.A. Segura, R.H. Mascheroni, and V.O. Salvadori. Prediction of foods freezing and thawing times: Artificial neural networks and genetic algorithm approach. *Journal of Food Engineering*, 84(1):164–178, 2008.

Z. Guan and H. Zhao. A semiparametric approach for marker gene selection based on gene expression data. *Bioinformatics*, 21(4):529–536, 2005.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3: 1157–1182, 2003.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: An update. volume 11, 2009.

C. Hayter. Cancer: "the worst scourge of civilized mankind". *Can Bull Med Hist.*, 20(2):251–264, 2003.

G. He, D. Wu, A. Sun, Y. Xue, Z. Jin, H. Qiu, M. Miao, X. Tang, Z. Fu, and Z. Chen. Cyt CD79a expression in acute leukemia with t(8;21): biphenotypic or myeloid leukemia? *Cancer Genet Cytogenet*, 174(1):76–77, 2007.

P.S. Heckerling, G.J. Canaris, S.D. Flach, T.G. Tape, R.S. Wigton, and B.S. Gerber. Predictors of urinary tract infection based on artificial neural networks and genetic algorithms. *International Journal of Medical Informatics*, 76(4):289–296, 2007.

J.H. Holland. *Adaptation in Natural and Artificial Systems.* The MIT Press, 1992. ISBN 0-262-08213-6.

K.K. Hoyer, M. Herling, K. Bagrintseva, D.W. Dawson, S.W. French, M. Renard, J.G. Weinger, D. Jones, and M.A. Teitell. T cell leukemia-1 modulates TCR signal strength and IFN-gamma levels through phosphatidylinositol 3-kinase and protein kinase c pathway activation. *J Immunol*, 175(2):864–873, 2005.

HQ.P. Hu, M. Xie, S.H. Ng, and G. Levitin. Robust recurrent neural network modeling for software fault detection and correction prediction. *Reliability Engineering & System Safety*, 92(3):332–340, 2007.

K.B. Hwang, D.Y. Cho, S.W. Park, S.D. Kim, and B.Y. Zhang. Applying machine learning techniques to analysis of gene expression data: Cancer diagnosis. In *CAMDA'02: The Conference on Critical Assessment of Microarray Data Analysis, proceedings*, pages 167–182, 2002.

I. Inza, P. Larrañaga, R. Blanco, and A.J. Cerrolaza. Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence in Medicine*, 31(2):91–103, 2004.

I.B. Jeffery, D.G. Higgins, and A.C. Culhane. Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. *BMC Bioinformatics*, 7:359, 2006.

I.S. Jeon, J.N. Davis, B.S. Braun, J.E. Sublett, M.F. Roussel, C.T. Denny, and D.N. Shapiro. A variant ewing's sarcoma translocation (7;22) fuses the EWS gene to the ETS gene ETV1. *Oncogene*, 10(6): 1229–1234, 1995.

D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: A survey. *IEEE Transaction on Knowledge and Data Engineering*, 16(11):1370–1386, 2004.

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

L.C. Kang and C.H. Dunphy. Immunoreactivity of MIC2 (CD99) and terminal deoxynucleotidyl transferase in bone marrow clot and core specimens of acute myeloid leukemias and myelodysplastic syndromes. *Arch Pathol Lab Med*, 130(2):153–157, 2006.

M. Karzynski, Á. Mateos, J. Herrero, and J. Dopazo. Using a genetic algorithm and a perceptron for feature selection and supervised class learningin DNA microarray data. *Artif. Intell. Rev.*, 20(1-2):39–51, 2003.

E.C. Keedwell and A. Narayanan. Genetic algorithms for gene expression analysis. In Raidl et al., editor, *EvoWorkshops'03: Applications in Evolutionary Computing, 1st European Workshop on Evolutionary Bioinformatics, proceedings*, volume 2611 of *Lecture Notes in Computer Science*, pages 76–86. Springer, 2003.

T.W. Kelley, D. Huntsman, K.M. McNagny, C.D. Roskelley, and E.D. Hsi. Podocalyxin: a marker of blasts in acute leukemia. *Am J Clin Pathol*, 124(1):134–142, 2005.

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

H. Khoury, B.I. Dalal, S.H. Nantel, D.E. Horsman, J.C. Lavoie, J.D. Shepherd, D.E. Hogge, C.L. Toze, K.W. Song, D.L. Forrest, H.J. Sutherland, and T.J. Nevill. Correlation between karyotype and quantitative immunophenotype in acute myelogenous leukemia with t(8;21). *Mod Pathol*, 17(10):1211–1216, 2004.

J.D. Khoury. Ewing sarcoma family of tumors. *Adv Anat Pathol*, 12(4):212–220, 2005.

H. Kim, G.H. Golub, and H. Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, 21(2):187–198, 2005.

K-J. Kim and I. Han. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, 19(2):125–132, 2000.

B. Kisacik, A. Akdogan, G. Yilmaz, O. Karadag, F.M. Yilmaz, S. Koklu, O. Yuksel, A.I. Ertenli, and S. Kiraz. Serum adenosine deaminase activities during acute attacks and attack-free periods of familial mediterranean fever. *Eur J Intern Med*, 20(1):44–47, 2009.

N. Kiyokawa, T. Sekino, T. Matsui, H. Takenouchi, K. Mimori, W.R. Tang, J. Matsui, T. Taguchi, Y.U. Katagiri, H. Okita, Y. Matsuo, H. Karasuyama, and J. Fujimoto. Diagnostic importance of CD179a/b as markers of precursor b-cell lymphoblastic lymphoma. *Mod Pathol*, 17(4):423–429, 2004.

D. Ko, W. Xu, and B. Windle. Gene function classification using NCI-60 cell line gene expression profiles. *Computational Biology and Chemistry*, 29(6):412–419, 2005.

R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2):273–324, 1997.

S.B. Kotsiantis. Supervised machine learning: A review of classification. *Informatica*, 31:249–268, 2007.

S.B. Kotsiantis, D. Kanellopoulos, and P.E. Pintelas. Data preprocessing for supervised leaning. *International Journal of Computer Science*, 1(2):111–117, 2006.

W.P. Kuo, E-Y. Kim, J. Trimarchi, T-K. Jenssen, S.A. Vinterbo, and L. Ohno-Machado. A primer on gene expression and microarrays for machine learning researchers. *J. of Biomedical Informatics*, 37(4):293–303, 2004.

Y-K. Kwon and B.R. Moon. Nonlinear feature extraction using a neuro genetic hybrid. In H-G. Beyer and U-M. O'Reilly, editors, *GECCO'05: Genetic and Evolutionary Computation Conference, proceedings*, pages 2089–2096, New York, NY, USA, 2005. ACM.

D. Lee, J.W. Kim, T. Seoand S.G. Hwang, E.J. Choi, and J. Choe. SWI/SNF complex interacts with tumor suppressor p53 and is necessary for the activation of p53-mediated transcription. *J Biol Chem*, 277(25): 22330–22337, 2002.

J.W. Lee, J.B. Lee, M. Park, and S.H. Song. An extensive comparison of recent classification tools applied to microarray data. *Computational statistics & data analysis*, 48(4):869–885, 2005.

Y. Lee and C-K. Lee. Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*, 19(9):1132–1139, 2003.

Y.F. Leung and D. Cavalieri. Fundamentals of cDNA microarray data analysis. *Trends in Genetics*, 19(11): 649–659, 2003.

L. Li, T.A. Darden, C.R. Weinberg, A.J. Levine, and L.G. Pedersen. Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method. *Combinatorial Chemistry & High Throughput Screen*, 4:727–739, 2001a.

L. Li, C.R. Weinberg, T.A. Darden, and L.G. Pedersen. Gene selection for sample classification based on gene expression data: Study of sensitivity to choice of parameters of the GA/kNN method. *Bioinformatics*, 17 (12):1131–1142, 2001b.

T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437, 2004.

W. Li and Y. Yang. How many genes are needed for a discriminant microarray data analysis? In S.M. Lin and K.F. Johnson, editors, *Methods of Microarray Data Analysis*, CAMDA'00: Critical Assessment of Techniques for Microarray Data Analysis, pages 137–150. Kluwer Academic, 2002.

P. Lidén, L. Asker, and H. Boström. Rule induction for classification of gene expression array data. In *PKDD'02: 6th European Conference on Principles of Data Mining and Knowledge Discovery, proceedings*, pages 338–347. Springer, 2002.

T-C. Lin, R-S. Liu, Y-T. Chao, and S-Y. Chen. Multiclass microarray data classification using GA/ANN method. In Q. Yang and G.I. Webb, editors, *PRICAI'06: Trends in Artificial Intelligence, 9th Pacific Rim International Conference on Artificial Intelligence, proceedings*, volume 4099 of *Lecture Notes in Computer Science*, pages 1037–1041. Springer, 2006.

B. Liu, Q. Cui, T. Jiang, and S. Ma. A combinational feature selection and ensemble neural network method for classification of gene expression data. *BMC Bioinformatics*, 5:136, 2004a.

J.J. Liu, G. Cutler, W. Li, Z. Pan, S. Peng, T. Hoey, L. Chen, and X.B. Ling. Multiclass cancer classification and biomarker discovery using GA-based algorithms. *Bioinformatics*, 21(11):2691–2697, 2005a.

L. Liu, L. McGavran, M.A. Lovell, Q. Wei, B.A. Jamieson, S.A. Williams, N.N. Dirks, M.S. Danielson, L.M. Dubie, and X. Liang. Nonpositive terminal deoxynucleotidyl transferase in pediatric precursor b-lymphoblastic leukemia. *Am J Clin Pathol*, 121(6):810–815, 2004b.

X. Liu, A. Krishnan, and A. Mondry. An entropy-based gene selection method for cancer classification using microarray data. *BMC Bioinformatics*, 6:76, 2005b.

Z. Liu, C. Wang, A. Liu, and Z. Niu. Evolving neural network using real coded genetic algorithm (GA) for multispectral image classification. *Future Generation Computer Systems*, 20(7):1119–1129, 2004c.

Y. Lu and J. Han. Cancer classification using gene expression data. *Information Systems*, 28(4):243–268, 2003.

N. Mah, A. Thelin, T. Lu, S. Nikolaus, T. Kühbacher, Y. Gurbuz, H. Eickhoff, G. Klöppel, H. Lehrach, B. Mellgård, C.M. Costello1, and S. Schreiber. A comparison of oligonucleotide and cDNA-based microarray systems. *Physiol. Genomics*, 16:361–370, 2004.

S. Makawita, M. Ho, A.D. Durbin, P.S. Thorner, D. Malkin, and G.R. Somers. Expression of insulin-like growth factor pathway proteins in rhabdomyosarcoma: IGF-2 expression is associated with translocation-negative tumors. *Pediatr Dev Pathol*, 12(2):127–135, 2009.

M. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML'03: Workshop on Learning from Imbalanced Data Sets, proceedings*, 2003.

Y. Mao, X. Zhou, D. Pi, Y. Sun, and S.T.C. Wong. Multiclass cancer classification by using fuzzy support vector machine and binary decision tree with gene selection. *Journal of Biomedicine and Biotechnology*, 2005(2):160–171, 2005.

F. Markowetz and R. Spang. Molecular diagnosis: Classification, model selection and performance evaluation. *Methods Inf Med.*, 44(3):438–443, 2005.

P.P. Medina, O.A. Romero, T. Kohno, L.M. Montuenga, R. Pio, J. Yokota, and M. Sanchez-Cespedes. Frequent BRG1/SMARCA4-inactivating mutations in human lung cancer cell lines. *Hum Mutat*, 29(5): 617–622, 2008.

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

T.M. Mitchell. Does machine learning really work? *AI Magazine*, 18(3):11–20, 1997b.

S. Mocellin and C.R. Rossi. Principles of gene microarray data analysis. *Advances in experimental medicine and biology*, 593:19–30, 2007.

D.J. Montana and L. Davis. Training feedforward neural networks using genetic algorithms. In *11th International Joint Conference on Artificial Intelligence, proceedings*, pages 762–767. Morgan Kaufmann, 1989.

Y.P. Mossé, M. Laudenslager, L. Longo, K.A. Cole, A. Wood, E.F. Attiyeh, M.J. Laquaglia, R. Sennett, J.E. Lynch, P. Perri, G. Laureys, F. Speleman, C. Kim, C. Hou, H. Hakonarson, A. Torkamani, N.J. Schork, G.M. Brodeur, G.P. Tonini, E. Rappaport, M. Devoto, and J.M. Maris. Identification of ALK as a major familial neuroblastoma predisposition gene. *Nature*, 455(7215):930–935, 2008.

M. Mramor, G. Leban, J. Demsar, and B. Zupan. Conquering the curse of dimensionality in gene expression cancer diagnosis: Tough problem, simple models. In S. Miksch, J. Hunter, and E.T. Keravnou, editors, *AIME'05: 10th Conference on Artificial Intelligence in Medicine, proceedings*, volume 3581 of *Lecture Notes in Computer Science*, pages 514–523. Springer, 2005.

C.G. Mullighan, S. Goorha, I. Radtke, C.B. Miller, E. Coustan-Smith, J.D. Dalton, K. Girtman, S. Mathew, J. Ma, S.B. Pounds, X. Su, C-H. Pui, M.V. Relling, W.E. Evans, S.A. Shurtleff, and J.R. Downing. Genome-wide analysis of genetic alterations in acute lymphoblastic leukaemia. *Nature*, 446:758–764, 2007.

L. Nanni and A. Lumini. Ensemblator: An ensemble of classifiers for reliable classification of biological data. *Pattern Recognition Letters*, 28(5):622–630, 2007.

A. Narayanan, A. Cheung, J. Gamalielsson, E. Keedwell, and C. Vercellone. *Bioinformatics using Computational Intelligence Paradigms*, chapter Artificial Neural Networks for Reducing the Dimensionality of Gene Expression Data. Studies in Fuzziness and Soft Computing. Springer Berlin / Heidelberg, 2005.

M. Nasseri, K. Asghari, and M.J. Abedini. Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network. *Expert Syst. Appl.*, 35(3):1415–1421, 2008.

G. Nunnari. Modelling air pollution time-series by using wavelet functions and genetic algorithms. *Soft Computing*, 8(3):173–178, 2004.

S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii. A bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096, 2003.

C.H. Ooi and P. Tan. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. *Bioinformatics*, 19(1):37–44, 2003.

A. Osareh and B. Shadgar. Classification and diagnostic prediction of cancers using gene microarray data analysis. *Journal of Applied Sciences*, 9(3):459–468, 2008.

L.A. Owen, A.A. Kowalewski, and S.L. Lessnick. EWS/FLI mediates transcriptional repression via NKX2.2 during oncogenic transformation in ewing's sarcoma. *PLoS ONE*, 3(4):e1965, 2008.

N.R. Pal, K. Aguan, A. Sharma, and S i. Amari. Discovering biomarkers from gene expression data for predicting cancer subgroups using neural networks and relational fuzzy clustering. *BMC Bioinformatics*, 8:5, 2007.

A. Pandey and M. Mann. Proteomics to study genes and genomes. *Nature*, 405(6788):837–846, 2000.

C.H. Park, M. Jeon, P. Pardalos, and H. Park. Quality assessment of gene selection in microarray data. *Optimization Methods and Software*, 22(1):145–154, 2007.

G. Parmigiani, E.S. Garrett, R.A. Irizarry, and S.L. Zeger. *The Analysis of Gene Expression Data*, chapter The Analysis of Gene Expression Data: An Overview of Methods and Software, pages 1–45. Statistics for Biology and Health. Springer, 2003.

Y. Pekarsky, N. Zanesi, R. Aqeilan, and C.M. Croce. TCL1 as a model for lymphomagenesis. *Hematol Oncol Clin North Am*, 18(4):863–879, 2004.

S. Peng, Q. Xu, X.B. Ling, X. Peng, W. Du, and L. Chen. Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. *FEBS Letters*, 555(2):358–362, 2003.

A. Rajwanshi, R. Srinivas, and G. Upasana. Malignant small round cell tumors. *Journal of Cytology*, 26(1): 1–10, 2009.

S. Ramaswamy and T.R. Golub. DNA microarrays in clinical oncology. *Journal of Clinical Oncology*, 20(7): 1932–1941, 2002.

M. Reich, T. Liefeld, J. Gould, J. Lerner, P. Tamayo, and J.P. Mesirov. Genepattern 2.0. *Natural Genetics*, 38(5):500–501, 2006.

R. Robinson. *Genetics*, volume 3 of *Macmillan Reference USA Science Library*. Thomson Gale, 2003.

M. Rocha, R. Mendes, P. Maia, D. Glez-Pe na, and F. Fdez-Riverola. A platform for the selection of genes in DNA microarray data using evolutionary algorithms. In *GECCO'07: 9th Annual Conference on Genetic and Evolutionary Computation, proceedings*, pages 415–423. ACM, 2007.

D.T. Ross, U. Scherf, M.B. Eisen, C.M. Perou, C. Rees, P. Spellman, V. Iyer, S.S. Jeffrey, M.V. de Rijn, M. Waltham, A. Pergamenschikov, J.C. Lee, D. Lashkari, D. Shalon, T.G. Myers, J.N. Weinstein, D. Botstein, and P.O. Brown. Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, 24:227–235, 2000.

M.E. Ross, R. Mahfouz, M. Onciu, H-C. Liu, X. Zhou, G. Song, S.A. Shurtleff, S. Pounds, C. Cheng, J. Ma, R.C. Ribeiro, J.E. Rubnitz, K. Girtman, W.K. Williams, S.C. Raimondi, D-C. Liang, L-Y. Shih, C-H. Pui, and J.R. Downing. Gene expression profiling of pediatric acute myelogenous leukemia. *Blood*, 104 (12):3679–3687, 2004.

G.M. Saed, Z. Jiang, M.P. Diamond, and H.M. Abu-Soud. The role of myeloperoxidase in the pathogenesis of postoperative adhesions. *Wound Repair Regen*, 17(4):531–539, 2009.

Y. Saeys, I. Inza, and P. Larra naga. A review of feature selection techqniues in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

E. Sakhinia, M. Farahangpour, E. Tholouli, J.A. Liu Yin, J.A. Hoyland, and R.J. Byers. Comparison of gene-expression profiles in parallel bone marrow and peripheral blood samples in acute myeloid leukaemia by real-time polymerase chain reaction. *Journal of Clinical Pathology*, 59:1059–1065, 2006.

M.A. Sanz, L. Larrea, G. Sanz, G. Martín, A. Sempere, F. Gomis, J. Martínez, A. Regadera, S. Saavedra, I. Jarque, C. Jimínez, J. Cervera, and J. de La Rubia. Cutaneous promyelocytic sarcoma at sites of vascular access and marrow aspiration. A characteristic localization of chloromas in acute promyelocytic leukemia? *Haematologica*, 85(7):758–762, 2000.

A.C. Schierz. Virtual screening of bioassay data. *Journal of Chemoinformatics*, 1(2), 2009.

R. Schwartz, I. Engel, M. Fallahi-Sichani, H.T. Petrie, and C. Murre. Gene expression patterns define novel roles for E47 in cell cycle progression, cytokine-mediated signaling, and T lineage development. *Proc Natl Acad Sci U.S.A.*, 103(26):9976–9981, 2006.

G. Schwarzer, W. Vach, and M. Schumacher. On the misuses of artificial neural network for prognostic and diagnostic classification in oncology. *Statistics in Medicine*, 19(4):541–561, 2000.

P. Sethi, A.E. Alex, and S. Alagiriswamy. Feature ranking and scoring of gene expression data using associative pattern mining. In V. Kadirkamanathan et al., editor, *PRIB'09: 4th IAPR International Conference on Pattern Recognition in Bioinformatics, suppl. proceedings*, page Paper Id: 10, 2009.

R.S. Sexton and R.E. Dorsey. Reliable classification using neural networks: A genetic algorithm and backpropagation comparison. *Decis. Support Syst.*, 30(1):11–22, 2000.

R.S. Sexton and J.N.D. Gupta. Comparative evaluation of genetic algorithm and backpropagation for training neural networks. *Inf. Sci. Inf. Comput. Sci.*, 129(1-4):45–59, 2000.

D. Sheer and J.M. Shipley. *Introduction to the cellular and molecular biology of cancer*, chapter Molecular cytogenetics of cancer, pages 95–116. Oxford University Press, 2005.

L. Shen and E.C. Tan. Reducing multiclass cancer classification to binary by output coding and SVM. *Computational Biology and Chemistry*, 30(1):63–71, 2006.

E. Shenouda. A quantitative comparison of different MLP activation functions in classification. In *ISNN'06: Advances in neural networks, 3rd International Symposium on Neural Networks, proceedings, Part I-III*, volume 2971 of *Lecture Notes in Computer Science*, pages 849–857. Springer, 2006.

R. Simon. Diagnostic and prognostic prediction using gene expression profiles in high-dimensional microarray data. *British Journal of Cancer*, 89:1599–1604, 2003.

R. Simon, M.D. Radmacher, K. Dobbin, and L.M. Mcshane. Pitfalls in the use of dna microarray data for diagnostic and prognostic classification. *Journal of the National Cancer Institute*, 95(1):14–18, 2003.

A.C. Smith, S. Choufani, J.C. Ferreira, and R. Weksberg. Growth regulation, imprinted genes, and chromosome 11p15.5. *Pediatr Res*, 61(5 Pt 2):43R–47R, 2007.

R.L. Somorjai, B. Dolenko, and R. Baumgartner. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: Curses, caveats, cautions. *Bioinformatics*, 19(12):1484–1491, 2003.

P.H. Sorensen, S.L. Lessnick, D. Lopez-Terrada, X.F. Liu, T.J. Triche, and C.T. Denny. A second ewing's sarcoma translocation, t(21;22), fuses the EWS gene to another ETS-family transcription factor, erg. *Nat Genet*, 6(2):146–151, 1994.

A. Statnikov, C.F. Aliferis, and I. Tsamardinos. Methods for multi-category cancer diagnosis from gene expression data: A comprehensive evaluation to inform decision support system development. *Medinfo*, 11(Part 2):813–817, 2004.

A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5): 631–643, 2005.

D. Stekel. *Microarray Bioinformatics*, chapter Microarrays: Making then and using them, pages 1–18. Cambridge University Press, 2003a.

D. Stekel. *Microarray Bioinformatics*, chapter Normalisation, pages 73–99. Cambridge University Press, 2003b.

D. Stekel. *Microarray Bioinformatics*, chapter Data standards, storage and sharing, pages 231–252. Cambridge University Press, 2003c.

J.A. Strauchen. Indolent t-lymphoblastic proliferation: Report of a case with an 11-year history and association with myasthenia gravis. *Am J Surg Pathol*, 25(3):411–415, 2001.

G. Syswerda. Uniform crossover in genetic algorithms. In J.D. Schaffer, editor, *ICGA'89: 3rd International Conference on Genetic Algorithms, proceedings*, pages 2–9. Morgan Kaufmann, 1989.

M. Taheri and A. Mohebbi. Design of artificial neural networks using a genetic algorithm to predict collection efficiency in venturi scrubbers. *J Hazard Mater*, 157(1):122–129, 2008.

H. Takahashi, T. Kobayashi, and H. Honda. Construction of robust prognostic predictors by using projective adaptive resonance theory as a gene filtering method. *Bioinformatics*, 21(2):179–186, 2005.

F. Tan, X. Fu, Y. Zhang, and A.G. Bourgeois. A genetic algorithm-based method for feature subset selection. *Soft Comput.*, 12(2):111–120, 2007.

Y. Tan, L. Shi, W. Tong, and C. Wang. Multi-class cancer classification by total principal component regression (TPCR) using microarray gene expression data. *Nucleic Acids Research*, 33(1):56–65, 2005.

A.L. Tarca, V.J. Carey, X.W. Chen, R. Romero, and S. Drăghici. Machine learning and its applications to biology. *PLoS Comput Biol.*, 3(6):0953–0963, 2007.

J.G. Taylor, A.T. Cheuk, P.S Tsang, J.Y. Chung, Y.K. Song, K. Desai, Y. Yu, Q.R. Chen, K. Shah, V. Youngblood, J. Fang, S.Y. Kim, C. Yeung, L.J. Helman, A. Mendoza, V. Ngo, L.M. Staudt, J.S. Wei, C. Khanna, D. Catchpoole, S.J. Qualman, S.M. Hewitt, G. Merlino, S.J. Chanock, and J. Khan. Identification of FGFR4-activating mutations in human rhabdomyosarcomas that promote metastasis in xenotransplanted models. *J Clin Invest.*, 119(11):3395–3407, 2009.

The ALL/AML oligonucleotide microarray data. *The* B*road* I*nstitute*, 2007. Available from: http://www.broad.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43 [Acccessed 31 October 2007].

The GenePattern software suites. *The* B*road* I*nstitute*. Available from: http://www.broadinstitute.org/cancer/software/genepattern/ [Acccessed 6 May 2008].

The NCBI Genbank. *The* NCBI E*ntrez* G*ene system*. Available from: http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi [Acccessed 17 October 2009].

The R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006. Available from: http://www.R-project.org [ISBN 3-900051-07-0].

The SRBCTs cDNA microarray data. *The* NHGRI I*nstitute*, 2007. Available from: http://research.nhgri.nih.gov/microarray/Supplement/ [Acccessed 31 October 2007].

The Stanford SOURCE search and retrieval system. *The* S*tanford* U*niversity*. Available from http://source.stanford.edu/cgibin/source/sourceSearch [Acccessed 17 October 2009].

The WEKA data mining software. *The* U*niversity of* W*aikato*. Available from: http://www.cs.waikato.ac.nz/ml/weka/ [Accessed 1 October 2009].

R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *PNAS: Proceedings of the National Academy of Sciences of the United States of America*, 99(10):6567–6572, 2002.

S. Tomiuk and K. Hofmann. Microarray probe selection strategies. *Briefings in Bioinformatics*, 2(4):329–340, 2001.

D.L. Tong. Hybridising genetic algorithm-neural network (GANN) in marker genes detection. In *ICMLC'09: 8th International Conference on Machine Learning and Cybernetics, proceedings*, volume 2, pages 1082–1087, 2009.

D.L. Tong, K. Phalp, A. Schierz, and Robert Mintram. Innovative hybridisation of genetic algorithms and neural networks in detecting marker genes for leukaemia cancer. In V. Kadirkamanathan et al., editor, *PRIB'09: 4th IAPR International Conference on Pattern Recognition in Bioinformatics, suppl. proceedings*, 2009.

A. Toure and M. Basu. Application of neural network to gene expression data for cancer classification. In *IJCNN'01: International Joint Conference on Neural Networks, proceedings*, volume 1, pages 583–587. IEEE Press, 2001.

O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R.B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

F. Valafar. Pattern recognition techniques in microarray data analysis: A survey. *Ann NY Acad Sci*, 980 (1):41–64, 2002.

J.J. Valdés and A.J. Barton. Gene discovery in leukemia revisited: A computational intelligence perspective. In *IEA/AIE'04: 17th International Conference on Innovations in applied artificial intelligence, proceedings*, pages 118–127. Springer, 2004.

V.E. Velculescu, L. Zhang, B. Vogelstein, and K.W. Kinzler. Serial analysis of gene expression. *Science*, 270: 484–487, 1995.

V.E. Velculescu, L. Zhang, W. Zhou, J. Vogelstein, M.A. Basrai, D.E. Bassett, P. Hieter, B. Vogelstein, and K.W. Kinzler. Characterization of the yeast transcriptome. *Cell*, 88(2):243–251, 1997.

V.E. Velculescu, C. Zhang, W. Zhou, G. Traverso, B.St. Croix, B. Vogelstein, and K.W. Kinzler. Serial analysis of gene expression - detailed protocol. SAGE Protocol 1.0e, Johns Hopkins Oncology Center and Howard Hughes Medical Institute, Baltimore, June 2000. URL `http://www.sagenet.org/`.

F. Vuillier, G. Dumas, C. Magnac, M.C. Prevost, A.I. Lalanne, P. Oppezzo, E. Melanitou, G. Dighiero, and B. Payelle-Brogard. Lower levels of surface b-cell-receptor expression in chronic lymphocytic leukemia are associated with glycosylation and folding defects of the mu and CD79a chains. *Blood*, 105(7):2933–2940, 2005.

H. Wang, S. Huang, J. Shou, E.W. Su, J.E. Onyia, B. Liao, and S. Li. Comparative analysis and integrative classification of NCI60 cell lines and primary tumors using gene expression profiling data. *BMC Genomics*, 7:166–176, 2006.

J. Wang, T.H. Bø, I. Jonassen, O. Myklebost, and E. Hovig. Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data. *BMC Bioinformatics*, 4:60–72, 2003.

L. Wang, F. Chu, and W. Xie. Accurate cancer classification using expressions of very few genes. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 4(1):40–53, 2007.

Y. Wang, D.J. Miller, and R. Clarke. Approaches to working in high-dimensional data spaces: gene expression microarrays. *British journal of cancer*, 98(6):1023–1028, 2008.

G. Weber, S. Vinterbo, and L. Ohno-Machado. Multivariate selection of genetic markers in diagnostic classification. *Artificial Intelligence in Medicine*, 31(2):155–167, 2004.

J.S. Wei, B.T. Greer, F. Westermann, S.M. Steinberg, C-G. Son, Q-R. Chen, C.C. Whiteford, S. Bilke, A.L. Krasnoselsky, N. Cenacchi, D. Catchpoole, F. Berthold, M. Schwab, and J. Khan. Prediction of clinical outcome using gene expression profiling and artificial neural networks for patients with neuroblastoma. *Cancer Research*, 64(19):6883–6891, 2004.

L.D. Whitley. An overview of evolutionary algorithms: practical issues and common pitfalls. *Information & Software Technology*, 43(14):817–831, 2001.

WHO. The World Health Organisation, 2010. Available from: http://www.who.int/mediacentre/factsheets/fs297/en/ [Accessed 20 May 2010].

D.L. Wilson, M.J. Buckley, C.A. Helliwell, and I.W. Wilson. New normalization methods for cDNA microarray data. *Bioinformatics*, 19(11):1325–1332, 2003.

R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3): 645–678, 2005.

R. Xu, G.C. Anagnostopoulos, and D.C. Wunsch. Multiclass cancer classification using semisupervised ellipsoid ARTMAP and particle swarm optimization with gene expression data. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 4(1):65–77, 2007.

C-H. Yeang, S. Ramaswamy, P. Tamayo, S. Mukherjee, R.M. Rifkin, M. Angelo, M. Reich, E. Lander, J. Mesirov, and T.R. Golub. Molecular classification of multiple tumor types. *Bioinformatics*, 17(90001): S316–S322, 2001.

J. Yu, J. Yu, A.A. Almal, S.M. Dhanasekaran, D. Ghosh, W.P. Worzel, and A.M. Chinnaiyan. Feature selection and molecular classification of cancer using genetic programming. *Neoplasia*, 9(4):292303, 2007.

C. Zhang, H-R. Li, J-B. Fan, J. Wang-Rodriguez, T. Downs, X-D. Fu, and M.Q. Zhang. Profiling alternatively spliced mRNA isoforms for prostate cancer classification. *BMC Bioinformatics*, 7:202, 2006.

J. Zhang, T. Jiang, B. Liu, X. Jiang, and H. Zhao. Systematic benchmarking of microarray data feature extraction and classification. *International Journal of Computer Mathematics*, 85(5):803–811, 2008.

P. Zhang, B. Verma, and K. Kumar. Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection. *Pattern Recognition Letters*, 26(7):909–919, 2005.

X. Zhou and K.Z. Mao. LS bound based gene selection for DNA microarray data. *Bioinformatics*, 21(8): 1559–1564, 2005.

X. Zhou, X. Wang, and E.R. Dougherty. Gene selection using logistic regression based on AIC, BIC and MDL criteria. *New Mathematics and Natural Computation*, 1(1):129–145, 2005.

Appendix A

Feature Extraction Model

This appendix contains relevant screen shot figures of the GANN prototype that is written in C++ programming.

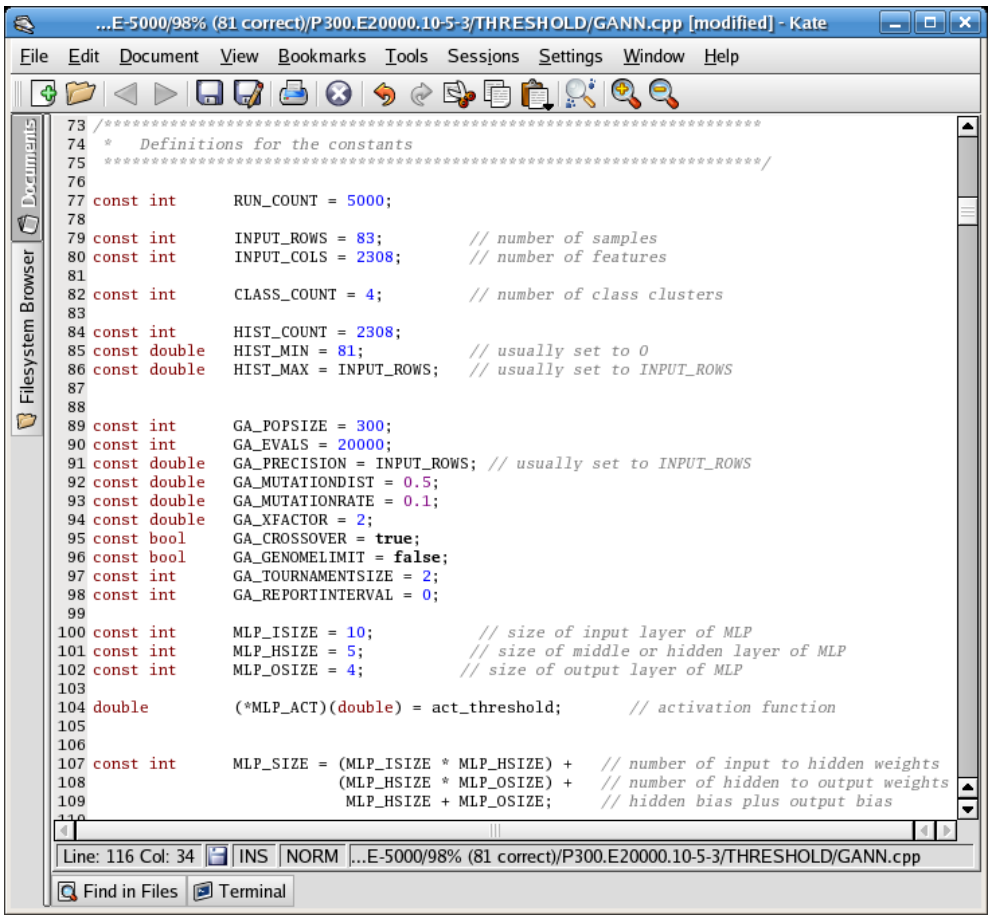


Figure A.1: The parameters in the Prototype.

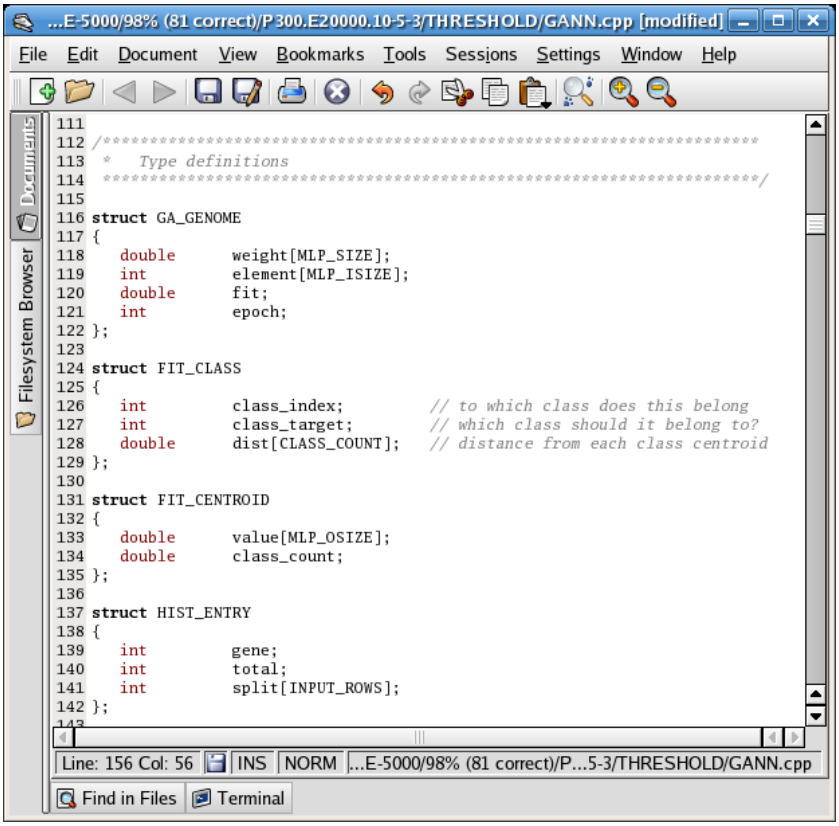
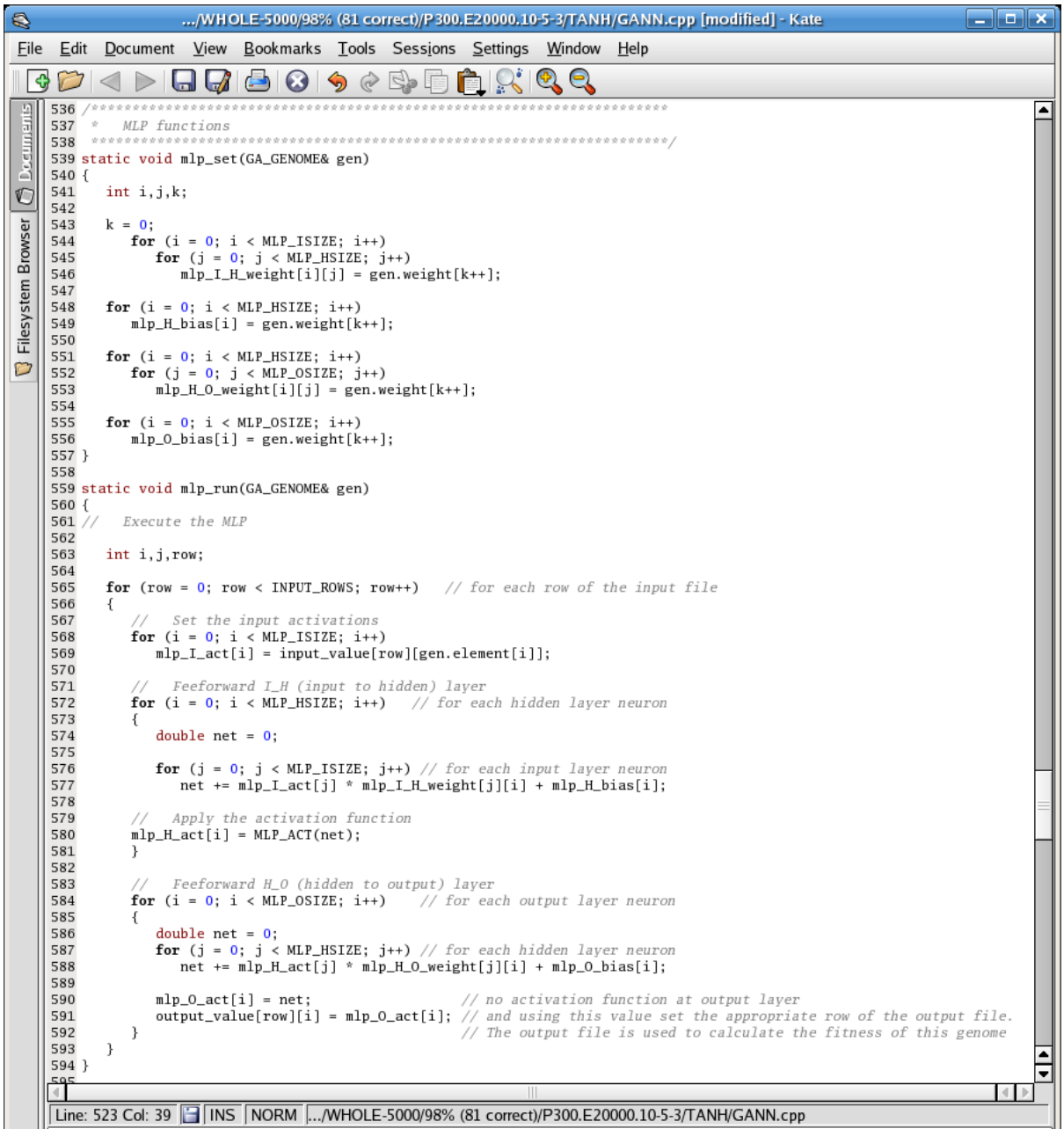


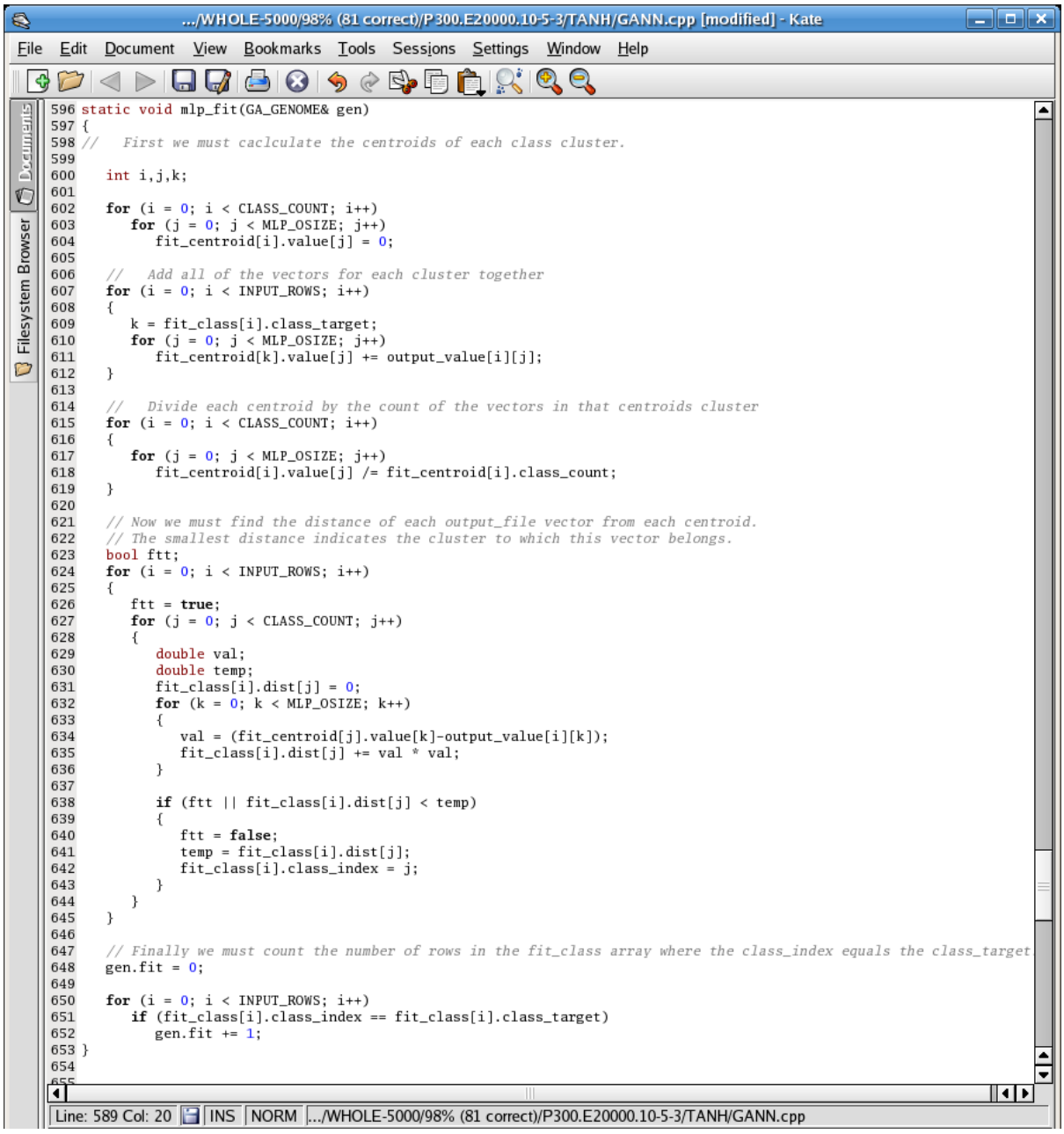
Figure A.2: The storage arrays in the Prototype.




```
536 /*************************************************************************
537  *   MLP functions
538  *************************************************************************/
539 static void mlp_set(GA_GENOME& gen)
540 {
541    int i,j,k;
542
543    k = 0;
544       for (i = 0; i < MLP_ISIZE; i++)
545          for (j = 0; j < MLP_HSIZE; j++)
546             mlp_I_H_weight[i][j] = gen.weight[k++];
547
548    for (i = 0; i < MLP_HSIZE; i++)
549       mlp_H_bias[i] = gen.weight[k++];
550
551    for (i = 0; i < MLP_HSIZE; i++)
552       for (j = 0; j < MLP_OSIZE; j++)
553          mlp_H_O_weight[i][j] = gen.weight[k++];
554
555    for (i = 0; i < MLP_OSIZE; i++)
556       mlp_O_bias[i] = gen.weight[k++];
557 }
558
559 static void mlp_run(GA_GENOME& gen)
560 {
561 //   Execute the MLP
562
563    int i,j,row;
564
565    for (row = 0; row < INPUT_ROWS; row++)   // for each row of the input file
566    {
567       //   Set the input activations
568       for (i = 0; i < MLP_ISIZE; i++)
569          mlp_I_act[i] = input_value[row][gen.element[i]];
570
571       //   Feeforward I_H (input to hidden) layer
572       for (i = 0; i < MLP_HSIZE; i++)   // for each hidden layer neuron
573       {
574          double net = 0;
575
576          for (j = 0; j < MLP_ISIZE; j++) // for each input layer neuron
577             net += mlp_I_act[j] * mlp_I_H_weight[j][i] + mlp_H_bias[i];
578
579       //   Apply the activation function
580       mlp_H_act[i] = MLP_ACT(net);
581       }
582
583       //   Feeforward H_O (hidden to output) layer
584       for (i = 0; i < MLP_OSIZE; i++)    // for each output layer neuron
585       {
586          double net = 0;
587          for (j = 0; j < MLP_HSIZE; j++) // for each hidden layer neuron
588             net += mlp_H_act[j] * mlp_H_O_weight[j][i] + mlp_O_bias[i];
589
590          mlp_O_act[i] = net;                  // no activation function at output layer
591          output_value[row][i] = mlp_O_act[i]; // and using this value set the appropriate row of the output file.
592       }                                       // The output file is used to calculate the fitness of this genome
593    }
594 }
```

(a) mlp_set() and mlp_runl()

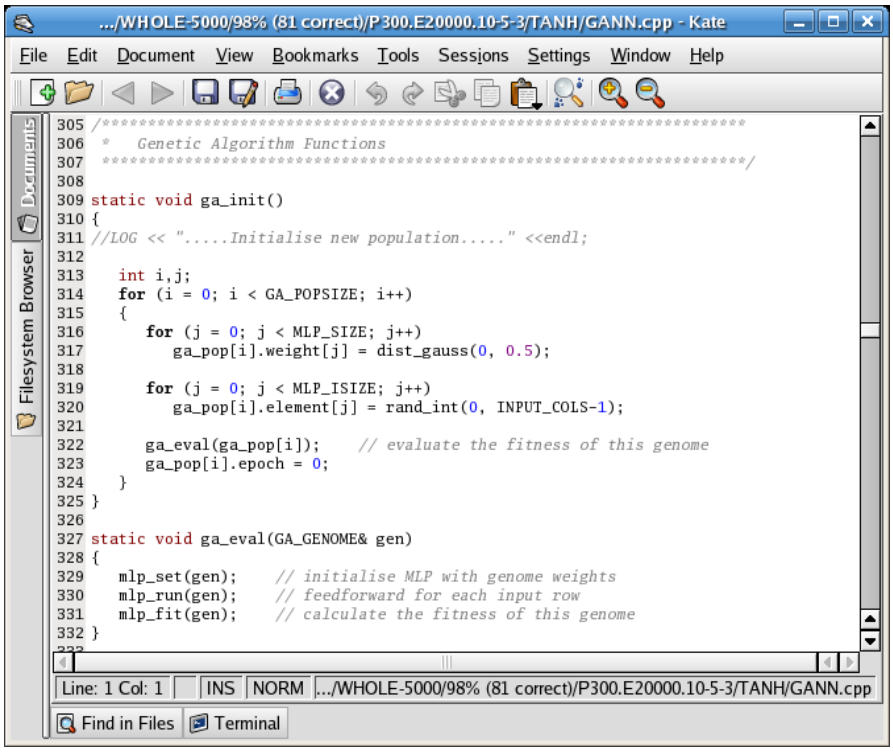
Figure A.3: The ANN functions in the Prototype.



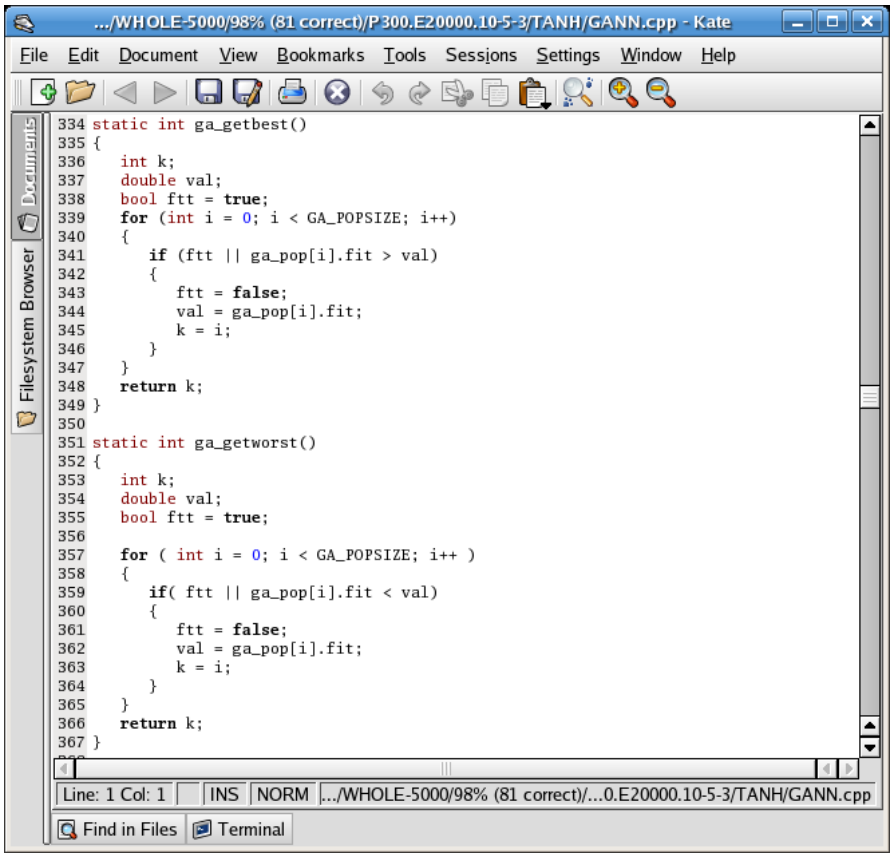

```
596 static void mlp_fit(GA_GENOME& gen)
597 {
598 //   First we must caclculate the centroids of each class cluster.
599
600    int i,j,k;
601
602    for (i = 0; i < CLASS_COUNT; i++)
603       for (j = 0; j < MLP_OSIZE; j++)
604          fit_centroid[i].value[j] = 0;
605
606    //   Add all of the vectors for each cluster together
607    for (i = 0; i < INPUT_ROWS; i++)
608    {
609       k = fit_class[i].class_target;
610       for (j = 0; j < MLP_OSIZE; j++)
611          fit_centroid[k].value[j] += output_value[i][j];
612    }
613
614    //   Divide each centroid by the count of the vectors in that centroids cluster
615    for (i = 0; i < CLASS_COUNT; i++)
616    {
617       for (j = 0; j < MLP_OSIZE; j++)
618          fit_centroid[i].value[j] /= fit_centroid[i].class_count;
619    }
620
621    // Now we must find the distance of each output_file vector from each centroid.
622    // The smallest distance indicates the cluster to which this vector belongs.
623    bool ftt;
624    for (i = 0; i < INPUT_ROWS; i++)
625    {
626       ftt = true;
627       for (j = 0; j < CLASS_COUNT; j++)
628       {
629          double val;
630          double temp;
631          fit_class[i].dist[j] = 0;
632          for (k = 0; k < MLP_OSIZE; k++)
633          {
634             val = (fit_centroid[j].value[k]-output_value[i][k]);
635             fit_class[i].dist[j] += val * val;
636          }
637
638          if (ftt || fit_class[i].dist[j] < temp)
639          {
640             ftt = false;
641             temp = fit_class[i].dist[j];
642             fit_class[i].class_index = j;
643          }
644       }
645    }
646
647    // Finally we must count the number of rows in the fit_class array where the class_index equals the class_target
648    gen.fit = 0;
649
650    for (i = 0; i < INPUT_ROWS; i++)
651       if (fit_class[i].class_index == fit_class[i].class_target)
652          gen.fit += 1;
653 }
654
```

(b) mlp_fit()

Figure A.3: – Continued

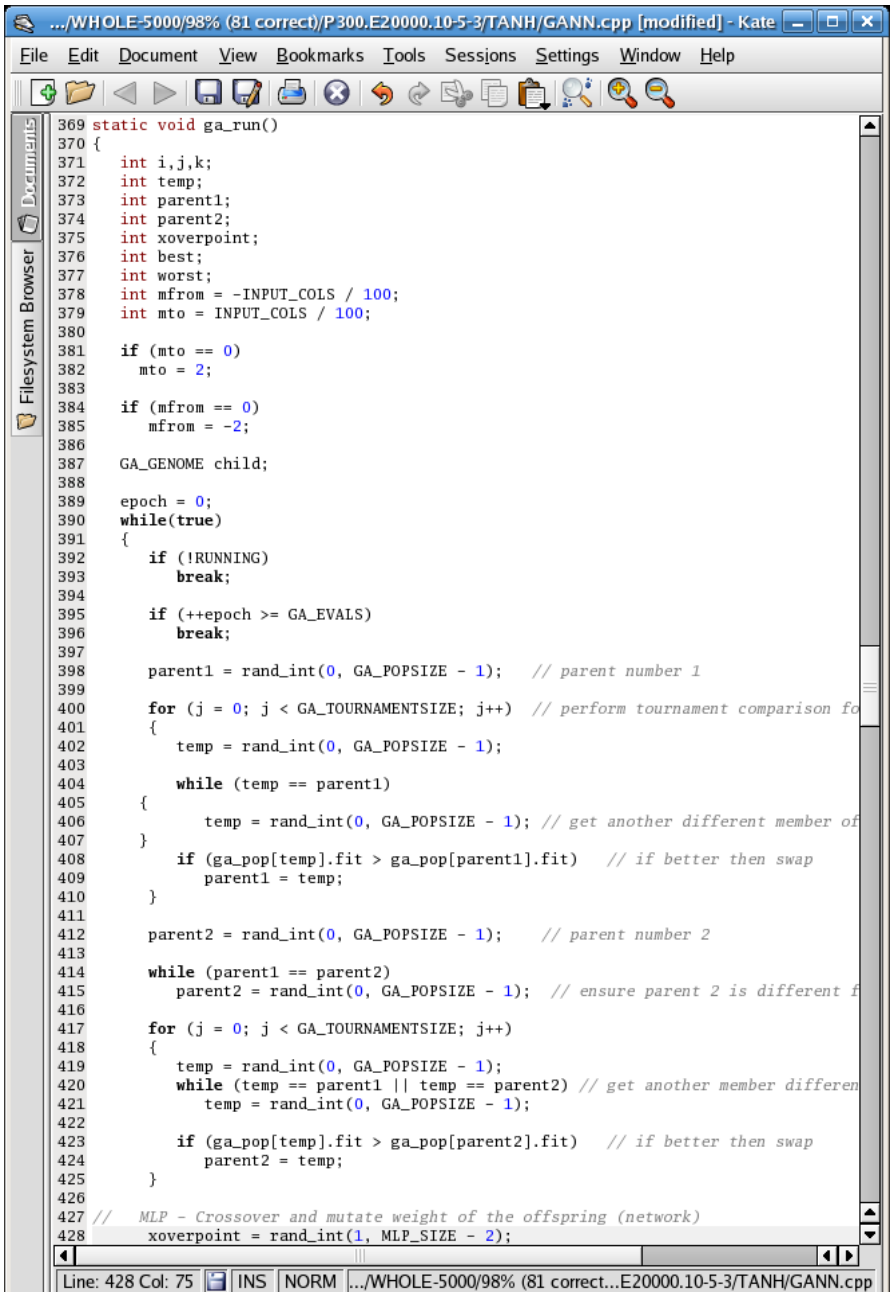


(a) ga_init() and ga_eval()



(b) ga_getbest() and ga_getworst()

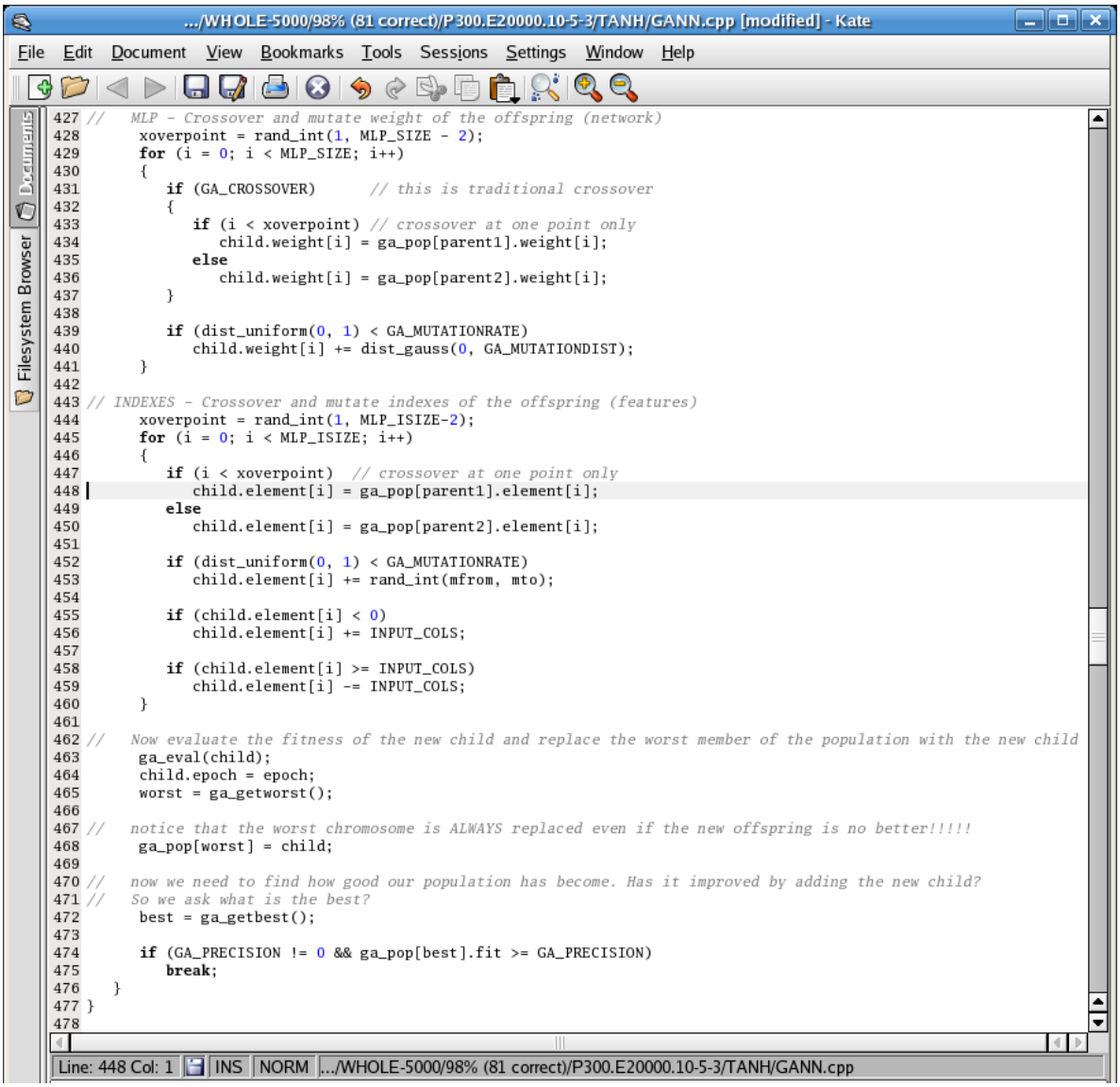
Figure A.4: The GA functions in the Prototype.




```
369 static void ga_run()
370 {
371    int i,j,k;
372    int temp;
373    int parent1;
374    int parent2;
375    int xoverpoint;
376    int best;
377    int worst;
378    int mfrom = -INPUT_COLS / 100;
379    int mto = INPUT_COLS / 100;
380
381    if (mto == 0)
382      mto = 2;
383
384    if (mfrom == 0)
385       mfrom = -2;
386
387    GA_GENOME child;
388
389    epoch = 0;
390    while(true)
391    {
392       if (!RUNNING)
393          break;
394
395       if (++epoch >= GA_EVALS)
396          break;
397
398       parent1 = rand_int(0, GA_POPSIZE - 1);   // parent number 1
399
400       for (j = 0; j < GA_TOURNAMENTSIZE; j++)  // perform tournament comparison fo
401       {
402          temp = rand_int(0, GA_POPSIZE - 1);
403
404          while (temp == parent1)
405       {
406             temp = rand_int(0, GA_POPSIZE - 1); // get another different member of
407       }
408          if (ga_pop[temp].fit > ga_pop[parent1].fit)   // if better then swap
409             parent1 = temp;
410       }
411
412       parent2 = rand_int(0, GA_POPSIZE - 1);    // parent number 2
413
414       while (parent1 == parent2)
415          parent2 = rand_int(0, GA_POPSIZE - 1);  // ensure parent 2 is different f
416
417       for (j = 0; j < GA_TOURNAMENTSIZE; j++)
418       {
419          temp = rand_int(0, GA_POPSIZE - 1);
420          while (temp == parent1 || temp == parent2) // get another member differen
421             temp = rand_int(0, GA_POPSIZE - 1);
422
423          if (ga_pop[temp].fit > ga_pop[parent2].fit)   // if better then swap
424             parent2 = temp;
425       }
426
427 //   MLP - Crossover and mutate weight of the offspring (network)
428       xoverpoint = rand_int(1, MLP_SIZE - 2);
```

(c) ga_run()

Figure A.4: – Continued




```
427 //   MLP - Crossover and mutate weight of the offspring (network)
428       xoverpoint = rand_int(1, MLP_SIZE - 2);
429       for (i = 0; i < MLP_SIZE; i++)
430       {
431          if (GA_CROSSOVER)      // this is traditional crossover
432          {
433             if (i < xoverpoint) // crossover at one point only
434                child.weight[i] = ga_pop[parent1].weight[i];
435             else
436                child.weight[i] = ga_pop[parent2].weight[i];
437          }
438
439          if (dist_uniform(0, 1) < GA_MUTATIONRATE)
440             child.weight[i] += dist_gauss(0, GA_MUTATIONDIST);
441       }
442
443 // INDEXES - Crossover and mutate indexes of the offspring (features)
444       xoverpoint = rand_int(1, MLP_ISIZE-2);
445       for (i = 0; i < MLP_ISIZE; i++)
446       {
447          if (i < xoverpoint)  // crossover at one point only
448             child.element[i] = ga_pop[parent1].element[i];
449          else
450             child.element[i] = ga_pop[parent2].element[i];
451
452          if (dist_uniform(0, 1) < GA_MUTATIONRATE)
453             child.element[i] += rand_int(mfrom, mto);
454
455          if (child.element[i] < 0)
456             child.element[i] += INPUT_COLS;
457
458          if (child.element[i] >= INPUT_COLS)
459             child.element[i] -= INPUT_COLS;
460       }
461
462 //   Now evaluate the fitness of the new child and replace the worst member of the population with the new child
463       ga_eval(child);
464       child.epoch = epoch;
465       worst = ga_getworst();
466
467 //   notice that the worst chromosome is ALWAYS replaced even if the new offspring is no better!!!!!
468       ga_pop[worst] = child;
469
470 //   now we need to find how good our population has become. Has it improved by adding the new child?
471 //   So we ask what is the best?
472       best = ga_getbest();
473
474       if (GA_PRECISION != 0 && ga_pop[best].fit >= GA_PRECISION)
475          break;
476    }
477 }
478
```

(d) ga_run() – Continued

Figure A.4: – Continued

Appendix B

Experimental Results

This appendix contains relevant tables to be used for supporting our findings reported in Chapter 5.

Table B.1: The complete list of synthetic data set 1 genes in the population size 100.

(a) Sigmoid-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**9**	*208*	**10**	*234*	**15**	*217*	**14**	*234*	**10**	*253*	**16**	*228*	**11**	*247*	**13**	*265*	**11**	*252*	**11**	*251*
2	**23**	*90*	**27**	*65*	**24**	*107*	**25**	*93*	**26**	*105*	**25**	*104*	**26**	*85*	**24**	*116*	**24**	*119*	**24**	*106*
3	**18**	*157*	**16**	*192*	**18**	*177*	**17**	*199*	**20**	*177*	**17**	*205*	**17**	*204*	**19**	*193*	**16**	*213*	**17**	*230*
4	**11**	*192*	**11**	*231*	**10**	*255*	**10**	*258*	**9**	*274*	**9**	*287*	**9**	*309*	**9**	*299*	**8**	*295*	**7**	*320*
5	**26**	*62*	**25**	*81*	**26**	*85*	**26**	*82*	**23**	*116*	**26**	*90*	**24**	*108*	**26**	*94*	**25**	*112*	**25**	*104*
6			**28**	*52*	**29**	*53*	**27**	*64*	**28**	*61*	**28**	*59*	**27**	*60*	**28**	*61*	**30**	*50*	**27**	*73*
7	**5**	*269*	**7**	*276*	**6**	*316*	**5**	*341*	**6**	*356*	**7**	*330*	**6**	*349*	**6**	*346*	**6**	*334*	**6**	*345*
8	**10**	*206*	**13**	*215*	**9**	*260*	**15**	*227*	**12**	*244*	**11**	*271*	**10**	*287*	**11**	*279*	**9**	*278*	**8**	*303*
9	**6**	*237*	**6**	*281*	**7**	*300*	**7**	*303*	**7**	*304*	**6**	*334*	**8**	*311*	**7**	*311*	**7**	*320*	**10**	*289*
10	**3**	*325*	**3**	*374*	**3**	*386*	**3**	*424*	**4**	*377*	**3**	*421*	**3**	*452*	**3**	*430*	**3**	*429*	**3**	*460*
11	**7**	*235*	**5**	*317*	**5**	*318*	**6**	*317*	**5**	*374*	**5**	*364*	**5**	*355*	**5**	*353*	**5**	*352*	**5**	*377*
12	**2**	*413*	**2**	*496*	**2**	*508*	**2**	*500*	**2**	*544*	**2**	*544*	**2**	*531*	**2**	*589*	**2**	*574*	**2**	*526*
13																	**29**	*57*	**30**	*52*
14	**20**	*130*	**19**	*159*	**20**	*159*	**21**	*155*	**18**	*190*	**20**	*177*	**21**	*170*	**21**	*152*	**21**	*170*	**20**	*179*
15			**26**	*69*	**27**	*63*	**28**	*60*	**27**	*64*	**29**	*58*	**29**	*52*	**29**	*60*	**27**	*71*	**28**	*69*
5001	**4**	*310*	**4**	*365*	**4**	*371*	**4**	*372*	**3**	*381*	**4**	*417*	**4**	*384*	**4**	*420*	**4**	*376*	**4**	*413*
5002	**25**	*67*	**24**	*83*	**25**	*96*	**24**	*97*	**25**	*106*	**24**	*110*	**25**	*100*	**25**	*103*	**26**	*111*	**26**	*90*
5003	**17**	*164*	**15**	*206*	**14**	*223*	**13**	*234*	**15**	*224*	**14**	*230*	**12**	*238*	**16**	*219*	**14**	*229*	**14**	*239*
5004	**22**	*108*	**22**	*124*	**22**	*113*	**23**	*103*	**24**	*113*	**23**	*133*	**23**	*113*	**23**	*130*	**23**	*122*	**22**	*126*
5005	**8**	*208*	**8**	*275*	**8**	*278*	**8**	*278*	**8**	*300*	**8**	*318*	**7**	*342*	**8**	*300*	**10**	*270*	**9**	*292*
5006	**19**	*130*	**20**	*152*	**17**	*187*	**20**	*169*	**17**	*209*	**19**	*182*	**19**	*182*	**18**	*197*	**19**	*186*	**18**	*196*
5007	**24**	*82*	**23**	*119*	**23**	*107*	**22**	*133*	**22**	*135*	**22**	*142*	**22**	*136*	**22**	*131*	**22**	*137*	**23**	*124*
5008	**16**	*167*	**14**	*210*	**13**	*228*	**12**	*242*	**16**	*220*	**13**	*241*	**15**	*227*	**15**	*221*	**15**	*215*	**12**	*250*
5009	**13**	*184*	**9**	*260*	**12**	*233*	**11**	*245*	**13**	*242*	**12**	*252*	**14**	*227*	**12**	*266*	**13**	*242*	**13**	*247*
5010	**15**	*170*	**18**	*165*	**16**	*187*	**16**	*226*	**14**	*226*	**15**	*228*	**16**	*218*	**17**	*206*	**17**	*209*	**15**	*236*
5011					**28**	*60*					**27**	*65*	**28**	*56*	**27**	*68*	**28**	*60*	**29**	*59*
5012	**14**	*173*	**17**	*179*	**19**	*171*	**18**	*179*	**19**	*184*	**18**	*190*	**18**	*196*	**14**	*223*	**18**	*191*	**19**	*183*
5013	**12**	*190*	**12**	*231*	**11**	*234*	**9**	*261*	**11**	*250*	**10**	*272*	**13**	*231*	**10**	*281*	**12**	*250*	**16**	*234*
5014	**1**	*532*	**1**	*644*	**1**	*664*	**1**	*647*	**1**	*687*	**1**	*727*	**1**	*709*	**1**	*767*	**1**	*690*	**1**	*763*
5015	**21**	*119*	**21**	*139*	**21**	*157*	**19**	*173*	**21**	*163*	**21**	*166*	**20**	*175*	**20**	*172*	**20**	*185*	**21**	*167*

Continued on Next Page...

Table B.1 – *Continued*

(b) Linear-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**10**	*256*	**12**	*280*	**11**	*303*	**12**	*286*	**10**	*317*	**15**	*286*	**13**	*290*	**13**	*297*	**8**	*350*	**9**	*321*
2	**25**	*94*	**24**	*124*	**26**	*101*	**25**	*125*	**25**	*117*	**24**	*123*	**24**	*126*	**23**	*131*	**24**	*128*	**25**	*128*
3	**13**	*223*	**17**	*224*	**17**	*225*	**17**	*211*	**18**	*215*	**17**	*248*	**16**	*239*	**17**	*238*	**16**	*254*	**18**	*251*
4	**7**	*287*	**8**	*336*	**9**	*326*	**8**	*336*	**11**	*315*	**9**	*323*	**9**	*370*	**8**	*337*	**10**	*342*	**10**	*317*
5	**23**	*127*	**26**	*103*	**25**	*105*	**26**	*124*	**26**	*113*	**25**	*118*	**26**	*107*	**24**	*127*	**26**	*123*	**24**	*136*
6	**27**	*78*	**29**	*74*	**28**	*79*	**28**	*77*	**27**	*73*	**28**	*83*	**27**	*76*	**29**	*66*	**27**	*84*	**29**	*66*
7	**6**	*311*	**7**	*357*	**6**	*369*	**6**	*352*	**5**	*405*	**6**	*379*	**6**	*392*	**7**	*369*	**5**	*403*	**6**	*397*
8	**11**	*252*	**10**	*281*	**12**	*296*	**15**	*248*	**13**	*297*	**12**	*307*	**12**	*290*	**10**	*316*	**14**	*299*	**12**	*315*
9	**9**	*271*	**6**	*362*	**7**	*369*	**7**	*339*	**8**	*347*	**7**	*358*	**8**	*377*	**6**	*383*	**9**	*347*	**7**	*374*
10	**3**	*419*	**3**	*465*	**3**	*479*	**3**	*432*	**3**	*472*	**3**	*475*	**3**	*518*	**3**	*512*	**3**	*491*	**3**	*511*
11	**5**	*327*	**5**	*403*	**5**	*390*	**5**	*396*	**6**	*397*	**4**	*432*	**5**	*410*	**4**	*419*	**6**	*403*	**5**	*421*
12	**2**	*498*	**2**	*593*	**2**	*611*	**2**	*595*	**2**	*676*	**2**	*639*	**2**	*641*	**2**	*671*	**2**	*635*	**2**	*674*
13	**30**	*55*	**30**	*74*	**30**	*58*			**30**	*65*	**29**	*80*	**30**	*52*	**30**	*59*	**29**	*60*	**30**	*56*
14	**20**	*169*	**20**	*197*	**20**	*205*	**19**	*198*	**21**	*179*	**21**	*194*	**21**	*187*	**21**	*201*	**20**	*216*	**19**	*216*
15	**29**	*55*	**28**	*75*	**27**	*82*	**27**	*89*	**29**	*65*	**27**	*90*	**28**	*70*	**28**	*70*	**30**	*56*	**27**	*77*
5001	**4**	*381*	**4**	*435*	**4**	*432*	**4**	*420*	**4**	*457*	**5**	*426*	**4**	*455*	**5**	*417*	**4**	*445*	**4**	*474*
5002	**26**	*85*	**25**	*103*	**23**	*116*	**22**	*143*	**24**	*134*	**26**	*115*	**25**	*117*	**26**	*120*	**25**	*126*	**26**	*104*
5003	**15**	*218*	**14**	*256*	**13**	*288*	**13**	*284*	**14**	*278*	**13**	*298*	**15**	*267*	**14**	*288*	**12**	*315*	**13**	*310*
5004	**22**	*129*	**22**	*130*	**22**	*128*	**23**	*132*	**23**	*137*	**22**	*150*	**23**	*145*	**22**	*135*	**23**	*136*	**23**	*149*
5005	**8**	*283*	**9**	*329*	**8**	*353*	**9**	*329*	**7**	*363*	**8**	*335*	**7**	*391*	**9**	*330*	**7**	*372*	**8**	*370*
5006	**19**	*176*	**19**	*209*	**21**	*204*	**16**	*211*	**20**	*205*	**19**	*218*	**18**	*231*	**20**	*218*	**19**	*216*	**20**	*213*
5007	**24**	*114*	**23**	*130*	**24**	*114*	**24**	*130*	**22**	*147*	**23**	*141*	**22**	*151*	**25**	*124*	**22**	*147*	**22**	*175*
5008	**14**	*219*	**15**	*255*	**14**	*287*	**11**	*292*	**15**	*258*	**14**	*293*	**14**	*271*	**15**	*275*	**15**	*288*	**15**	*305*
5009	**16**	*216*	**11**	*281*	**10**	*308*	**14**	*281*	**9**	*329*	**11**	*315*	**10**	*301*	**12**	*306*	**11**	*319*	**11**	*317*
5010	**18**	*188*	**16**	*240*	**16**	*226*	**18**	*210*	**16**	*235*	**16**	*256*	**17**	*232*	**16**	*261*	**18**	*241*	**16**	*260*
5011	**28**	*72*	**27**	*79*	**29**	*58*	**29**	*74*	**28**	*67*	**30**	*71*	**29**	*65*	**27**	*84*	**28**	*67*	**28**	*73*
5012	**17**	*191*	**18**	*212*	**18**	*219*	**20**	*196*	**17**	*215*	**18**	*224*	**19**	*226*	**19**	*222*	**17**	*254*	**17**	*252*
5013	**12**	*224*	**13**	*273*	**15**	*276*	**10**	*302*	**12**	*299*	**10**	*316*	**11**	*293*	**11**	*313*	**13**	*303*	**14**	*310*
5014	**1**	*677*	**1**	*732*	**1**	*767*	**1**	*752*	**1**	*803*	**1**	*837*	**1**	*802*	**1**	*784*	**1**	*811*	**1**	*821*
5015	**21**	*145*	**21**	*187*	**19**	*217*	**21**	*185*	**19**	*208*	**20**	*211*	**20**	*213*	**18**	*223*	**21**	*205*	**21**	*205*

(c) Tanh-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**9**	*179*	**12**	*195*	**15**	*190*	**13**	*210*	**14**	*211*	**15**	*214*	**12**	*232*	**13**	*228*	**14**	*227*	**14**	*238*
2			**26**	*65*	**24**	*80*	**25**	*73*	**25**	*92*	**24**	*92*	**23**	*104*	**23**	*93*	**25**	*89*	**26**	*77*
3	**16**	*133*	**17**	*156*	**19**	*154*	**17**	*183*	**19**	*164*	**17**	*188*	**17**	*184*	**16**	*211*	**16**	*196*	**18**	*193*
4	**12**	*166*	**8**	*233*	**11**	*217*	**9**	*265*	**9**	*253*	**10**	*240*	**9**	*264*	**10**	*258*	**10**	*257*	**10**	*276*
5	**24**	*69*	**25**	*74*	**25**	*75*	**24**	*94*	**26**	*84*	**26**	*79*	**26**	*81*	**24**	*86*	**24**	*100*	**24**	*97*
6									**28**	*50*	**27**	*61*			**27**	*55*	**29**	*56*	**28**	*54*
7	**7**	*205*	**6**	*248*	**8**	*270*	**6**	*315*	**7**	*264*	**6**	*296*	**6**	*313*	**6**	*305*	**6**	*332*	**6**	*345*
8	**14**	*154*	**11**	*212*	**9**	*246*	**11**	*221*	**10**	*245*	**9**	*252*	**8**	*268*	**8**	*281*	**7**	*300*	**8**	*296*
9	**5**	*230*	**9**	*228*	**6**	*284*	**8**	*278*	**6**	*266*	**7**	*267*	**7**	*283*	**9**	*281*	**8**	*280*	**9**	*286*
10	**3**	*310*	**3**	*360*	**3**	*353*	**3**	*366*	**4**	*354*	**4**	*352*	**3**	*365*	**4**	*375*	**3**	*375*	**3**	*399*
11	**6**	*215*	**5**	*270*	**5**	*297*	**5**	*319*	**5**	*287*	**5**	*334*	**5**	*319*	**5**	*314*	**5**	*337*	**5**	*348*
12	**2**	*362*	**2**	*436*	**2**	*443*	**2**	*467*	**2**	*471*	**2**	*496*	**2**	*504*	**2**	*517*	**2**	*476*	**2**	*480*
14	**21**	*104*	**22**	*104*	**21**	*132*	**21**	*127*	**21**	*146*	**20**	*157*	**19**	*156*	**20**	*163*	**20**	*147*	**20**	*164*
15					**27**	*50*	**27**	*55*	**27**	*62*	**28**	*56*					**27**	*67*	**27**	*68*
5001	**4**	*287*	**4**	*339*	**4**	*329*	**4**	*339*	**3**	*354*	**3**	*357*	**4**	*360*	**3**	*384*	**4**	*359*	**4**	*362*
5002	**25**	*54*	**24**	*79*	**26**	*75*	**26**	*64*	**23**	*106*	**25**	*87*	**25**	*92*	**26**	*77*	**26**	*83*	**25**	*89*
5003	**15**	*150*	**15**	*164*	**13**	*194*	**14**	*201*	**13**	*213*	**16**	*202*	**15**	*196*	**12**	*232*	**13**	*232*	**13**	*249*
5004	**23**	*76*	**21**	*108*	**22**	*114*	**22**	*109*	**24**	*97*	**22**	*113*	**22**	*116*	**25**	*83*	**23**	*112*	**23**	*105*
5005	**8**	*196*	**7**	*234*	**7**	*272*	**7**	*280*	**8**	*263*	**8**	*264*	**10**	*261*	**7**	*283*	**9**	*280*	**7**	*302*
5006	**20**	*118*	**16**	*158*	**18**	*157*	**19**	*156*	**18**	*168*	**18**	*177*	**20**	*144*	**18**	*177*	**18**	*194*	**19**	*168*
5007	**22**	*93*	**23**	*100*	**23**	*106*	**23**	*107*	**22**	*114*	**23**	*110*	**24**	*100*	**22**	*108*	**22**	*117*	**22**	*120*
5008	**13**	*162*	**14**	*175*	**14**	*194*	**15**	*184*	**15**	*202*	**13**	*222*	**16**	*185*	**15**	*215*	**15**	*198*	**16**	*225*
5009	**11**	*172*	**13**	*185*	**10**	*232*	**10**	*222*	**11**	*237*	**12**	*235*	**13**	*218*	**17**	*207*	**12**	*243*	**11**	*254*
5010	**19**	*118*	**18**	*152*	**16**	*175*	**18**	*179*	**16**	*190*	**14**	*219*	**14**	*199*	**14**	*217*	**17**	*194*	**15**	*228*
5011					**28**	*50*	**28**	*53*									**28**	*62*		
5012	**18**	*119*	**19**	*143*	**17**	*164*	**16**	*184*	**17**	*172*	**19**	*171*	**18**	*164*	**19**	*170*	**19**	*170*	**17**	*199*
5013	**10**	*174*	**10**	*215*	**12**	*195*	**12**	*217*	**12**	*232*	**11**	*238*	**11**	*241*	**11**	*254*	**11**	*255*	**12**	*249*
5014	**1**	*477*	**1**	*592*	**1**	*594*	**1**	*623*	**1**	*644*	**1**	*672*	**1**	*638*	**1**	*646*	**1**	*610*	**1**	*643*
5015	**17**	*129*	**20**	*125*	**20**	*140*	**20**	*138*	**20**	*150*	**21**	*139*	**21**	*140*	**21**	*155*	**21**	*143*	**21**	*154*

Continued on Next Page...

Table B.1 – *Continued*

(d) Threshold-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**14**	*210*	**16**	*222*	**9**	*280*	**9**	*286*	**12**	*250*	**10**	*293*	**13**	*263*	**10**	*297*	**14**	*269*	**13**	*268*
2	**24**	*88*	**26**	*96*	**26**	*84*	**22**	*113*	**25**	*105*	**26**	*84*	**26**	*99*	**26**	*95*	**24**	*111*	**26**	*110*
3	**17**	*169*	**18**	*175*	**16**	*212*	**16**	*211*	**17**	*204*	**17**	*205*	**16**	*228*	**17**	*209*	**16**	*217*	**17**	*228*
4	**10**	*241*	**8**	*289*	**10**	*277*	**8**	*302*	**9**	*301*	**9**	*301*	**9**	*290*	**9**	*326*	**9**	*300*	**9**	*322*
5	**25**	*85*	**24**	*111*	**25**	*89*	**23**	*112*	**26**	*95*	**24**	*102*	**23**	*118*	**25**	*105*	**25**	*100*	**25**	*121*
6	**28**	*55*	**27**	*65*	**27**	*66*	**27**	*70*	**28**	*63*	**27**	*73*	**29**	*57*	**28**	*63*	**27**	*72*	**27**	*63*
7	**7**	*263*	**6**	*300*	**5**	*330*	**5**	*378*	**6**	*357*	**6**	*369*	**7**	*334*	**5**	*391*	**6**	*371*	**6**	*364*
8	**15**	*202*	**14**	*226*	**11**	*268*	**13**	*244*	**10**	*289*	**11**	*280*	**12**	*266*	**12**	*281*	**12**	*269*	**12**	*270*
9	**5**	*274*	**7**	*300*	**7**	*309*	**7**	*352*	**7**	*343*	**8**	*304*	**6**	*343*	**8**	*334*	**8**	*346*	**7**	*351*
10	**3**	*361*	**3**	*426*	**3**	*438*	**3**	*458*	**3**	*427*	**3**	*455*	**3**	*467*	**3**	*440*	**3**	*434*	**3**	*441*
11	**6**	*268*	**5**	*341*	**6**	*327*	**6**	*372*	**4**	*401*	**5**	*390*	**5**	*343*	**6**	*364*	**4**	*406*	**5**	*399*
12	**2**	*490*	**2**	*543*	**2**	*562*	**2**	*584*	**2**	*543*	**2**	*508*	**2**	*577*	**2**	*615*	**2**	*555*	**2**	*562*
13					**30**	*56*									**30**	*50*	**29**	*60*	**30**	*50*
14	**19**	*150*	**19**	*171*	**20**	*177*	**19**	*190*	**20**	*185*	**21**	*155*	**19**	*196*	**20**	*174*	**21**	*175*	**20**	*168*
15	**27**	*58*	**29**	*57*	**29**	*56*	**29**	*54*	**29**	*61*	**29**	*56*	**28**	*58*	**27**	*73*	**28**	*63*	**29**	*52*
5001	**4**	*309*	**4**	*374*	**4**	*370*	**4**	*404*	**5**	*398*	**4**	*406*	**4**	*449*	**4**	*430*	**5**	*391*	**4**	*415*
5002	**26**	*74*	**25**	*105*	**24**	*100*	**26**	*96*	**24**	*107*	**25**	*95*	**25**	*106*	**23**	*129*	**26**	*99*	**23**	*126*
5003	**13**	*210*	**12**	*236*	**15**	*217*	**14**	*240*	**14**	*235*	**12**	*269*	**11**	*272*	**13**	*262*	**11**	*281*	**14**	*256*
5004	**22**	*109*	**23**	*114*	**22**	*119*	**24**	*111*	**22**	*121*	**23**	*113*	**24**	*117*	**22**	*136*	**23**	*119*	**24**	*121*
5005	**9**	*245*	**9**	*264*	**8**	*306*	**10**	*281*	**8**	*306*	**7**	*343*	**8**	*320*	**7**	*343*	**7**	*346*	**8**	*349*
5006	**20**	*143*	**21**	*159*	**19**	*178*	**21**	*178*	**18**	*194*	**20**	*166*	**21**	*175*	**19**	*174*	**18**	*201*	**19**	*189*
5007	**23**	*104*	**22**	*126*	**23**	*114*	**25**	*110*	**23**	*118*	**22**	*115*	**22**	*135*	**24**	*114*	**22**	*127*	**22**	*126*
5008	**11**	*231*	**13**	*228*	**14**	*243*	**15**	*236*	**15**	*225*	**14**	*247*	**14**	*254*	**14**	*255*	**13**	*269*	**15**	*241*
5009	**8**	*259*	**11**	*251*	**12**	*264*	**12**	*257*	**13**	*236*	**15**	*245*	**10**	*278*	**11**	*288*	**10**	*284*	**10**	*296*
5010	**16**	*195*	**15**	*222*	**18**	*194*	**17**	*205*	**16**	*212*	**16**	*226*	**17**	*227*	**16**	*209*	**17**	*213*	**16**	*237*
5011	**29**	*53*	**28**	*64*	**28**	*63*	**28**	*62*	**27**	*66*	**28**	*63*	**27**	*66*	**29**	*55*	**30**	*57*	**28**	*53*
5012	**18**	*159*	**17**	*190*	**21**	*171*	**18**	*196*	**19**	*192*	**18**	*191*	**18**	*216*	**18**	*208*	**19**	*183*	**18**	*210*
5013	**12**	*229*	**10**	*256*	**13**	*249*	**11**	*263*	**11**	*259*	**13**	*269*	**15**	*253*	**15**	*233*	**15**	*262*	**11**	*295*
5014	**1**	*593*	**1**	*672*	**1**	*671*	**1**	*715*	**1**	*731*	**1**	*752*	**1**	*776*	**1**	*751*	**1**	*706*	**1**	*757*
5015	**21**	*140*	**20**	*161*	**17**	*196*	**20**	*184*	**21**	*169*	**19**	*172*	**20**	*182*	**21**	*162*	**20**	*179*	**21**	*165*

Table B.2: The complete list of synthetic data set 1 genes in the population size 200.

(a) Sigmoid-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**10**	*388*	**10**	*447*	**12**	*463*	**14**	*429*	**11**	*505*	**11**	*510*	**12**	*477*	**13**	*477*	**11**	*472*	**12**	*470*
2	**24**	*180*	**25**	*206*	**24**	*196*	**24**	*218*	**24**	*208*	**24**	*222*	**24**	*220*	**24**	*223*	**25**	*216*	**24**	*230*
3	**17**	*312*	**16**	*377*	**17**	*391*	**18**	*379*	**16**	*423*	**17**	*386*	**17**	*405*	**19**	*360*	**17**	*397*	**19**	*373*
4	**12**	*377*	**9**	*506*	**11**	*495*	**10**	*519*	**10**	*528*	**10**	*520*	**9**	*537*	**10**	*509*	**9**	*556*	**13**	*461*
5	**25**	*171*	**26**	*177*	**25**	*188*	**26**	*191*	**26**	*194*	**26**	*169*	**26**	*179*	**26**	*210*	**24**	*216*	**26**	*195*
6	**29**	*100*	**27**	*122*	**28**	*112*	**30**	*104*	**27**	*130*	**28**	*124*	**28**	*128*	**29**	*124*	**27**	*146*	**29**	*122*
7	**6**	*480*	**6**	*574*	**6**	*646*	**6**	*640*	**6**	*643*	**6**	*656*	**5**	*661*	**5**	*685*	**6**	*630*	**5**	*654*
8	**11**	*378*	**12**	*443*	**9**	*515*	**9**	*538*	**9**	*546*	**9**	*522*	**11**	*520*	**9**	*560*	**10**	*522*	**9**	*564*
9	**7**	*450*	**8**	*553*	**7**	*613*	**7**	*616*	**8**	*564*	**7**	*640*	**7**	*577*	**7**	*582*	**7**	*602*	**8**	*577*
10	**4**	*585*	**4**	*717*	**3**	*752*	**4**	*726*	**3**	*775*	**4**	*741*	**3**	*768*	**4**	*706*	**3**	*767*	**4**	*771*
11	**5**	*539*	**5**	*646*	**5**	*660*	**5**	*646*	**5**	*648*	**5**	*666*	**6**	*619*	**6**	*674*	**5**	*676*	**6**	*617*
12	**2**	*724*	**2**	*883*	**2**	*905*	**2**	*961*	**2**	*976*	**2**	*932*	**2**	*937*	**2**	*932*	**2**	*901*	**2**	*952*
13	**30**	*78*	**30**	*109*	**30**	*99*	**29**	*110*	**30**	*98*	**30**	*87*	**30**	*94*	**30**	*111*	**30**	*81*	**30**	*95*
14	**20**	*251*	**21**	*286*	**19**	*339*	**20**	*316*	**21**	*317*	**20**	*335*	**20**	*334*	**21**	*322*	**21**	*315*	**20**	*340*
15	**27**	*120*	**28**	*119*	**27**	*126*	**27**	*147*	**28**	*128*	**27**	*146*	**27**	*137*	**28**	*142*	**28**	*119*	**28**	*131*
5001	**3**	*640*	**3**	*735*	**4**	*728*	**3**	*764*	**4**	*734*	**3**	*782*	**4**	*752*	**3**	*780*	**4**	*754*	**3**	*784*
5002	**26**	*147*	**24**	*211*	**26**	*188*	**25**	*210*	**25**	*201*	**25**	*218*	**25**	*218*	**25**	*217*	**26**	*192*	**25**	*213*
5003	**14**	*350*	**15**	*402*	**15**	*411*	**12**	*454*	**14**	*468*	**13**	*457*	**14**	*458*	**14**	*450*	**15**	*436*	**14**	*450*
5004	**23**	*205*	**23**	*226*	**22**	*259*	**23**	*226*	**22**	*245*	**22**	*275*	**23**	*227*	**22**	*255*	**23**	*243*	**22**	*267*
5005	**8**	*425*	**7**	*571*	**8**	*533*	**8**	*580*	**7**	*578*	**8**	*571*	**8**	*568*	**8**	*571*	**8**	*596*	**7**	*605*
5006	**19**	*282*	**17**	*354*	**21**	*325*	**19**	*346*	**19**	*341*	**19**	*347*	**19**	*352*	**17**	*370*	**20**	*361*	**18**	*380*
5007	**22**	*210*	**22**	*242*	**23**	*217*	**22**	*226*	**23**	*230*	**23**	*227*	**22**	*248*	**23**	*233*	**22**	*248*	**23**	*257*
5008	**13**	*358*	**14**	*412*	**16**	*393*	**15**	*427*	**15**	*435*	**16**	*406*	**15**	*425*	**16**	*417*	**14**	*444*	**15**	*432*
5009	**15**	*340*	**13**	*429*	**13**	*456*	**13**	*447*	**12**	*482*	**12**	*471*	**10**	*524*	**11**	*508*	**12**	*465*	**10**	*489*
5010	**16**	*314*	**18**	*352*	**14**	*424*	**17**	*410*	**17**	*402*	**15**	*412*	**16**	*419*	**15**	*432*	**16**	*407*	**17**	*393*
5011	**28**	*106*	**29**	*112*	**29**	*109*	**28**	*113*	**29**	*128*	**29**	*117*	**29**	*126*	**27**	*143*	**29**	*115*	**27**	*144*
5012	**18**	*308*	**19**	*338*	**18**	*386*	**16**	*419*	**18**	*350*	**18**	*368*	**18**	*399*	**18**	*363*	**18**	*390*	**16**	*404*
5013	**9**	*401*	**11**	*447*	**10**	*513*	**11**	*487*	**13**	*479*	**14**	*456*	**13**	*468*	**12**	*503*	**13**	*461*	**11**	*471*
5014	**1**	*988*	**1**	*1231*	**1**	*1217*	**1**	*1243*	**1**	*1215*	**1**	*1234*	**1**	*1251*	**1**	*1246*	**1**	*1282*	**1**	*1260*
5015	**21**	*240*	**20**	*301*	**20**	*333*	**21**	*300*	**20**	*330*	**21**	*318*	**21**	*321*	**20**	*345*	**19**	*365*	**21**	*332*

Continued on Next Page...

Table B.2 – *Continued*

(b) Linear-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**9**	*523*	**10**	*539*	**11**	*527*	**10**	*554*	**13**	*492*	**11**	*538*	**10**	*549*	**13**	*527*	**11**	*544*	**13**	*536*
2	**26**	*209*	**25**	*230*	**24**	*260*	**26**	*234*	**24**	*244*	**24**	*230*	**24**	*249*	**25**	*232*	**25**	*223*	**26**	*224*
3	**17**	*386*	**17**	*413*	**17**	*426*	**18**	*406*	**18**	*412*	**18**	*422*	**16**	*450*	**18**	*429*	**17**	*468*	**17**	*429*
4	**10**	*497*	**11**	*532*	**10**	*534*	**9**	*574*	**10**	*567*	**9**	*565*	**9**	*592*	**12**	*531*	**9**	*599*	**11**	*568*
5	**24**	*217*	**26**	*221*	**26**	*218*	**24**	*246*	**26**	*200*	**26**	*216*	**26**	*235*	**26**	*219*	**26**	*218*	**25**	*240*
6	**29**	*131*	**28**	*157*	**28**	*146*	**28**	*142*	**29**	*132*	**27**	*144*	**27**	*153*	**28**	*146*	**30**	*134*	**28**	*152*
7	**5**	*602*	**5**	*708*	**5**	*734*	**6**	*668*	**6**	*698*	**6**	*713*	**6**	*730*	**6**	*714*	**7**	*656*	**6**	*723*
8	**14**	*456*	**9**	*562*	**9**	*545*	**14**	*513*	**9**	*600*	**10**	*544*	**14**	*490*	**9**	*552*	**10**	*547*	**12**	*539*
9	**6**	*594*	**8**	*614*	**7**	*676*	**7**	*635*	**8**	*602*	**7**	*672*	**7**	*652*	**7**	*695*	**8**	*627*	**8**	*629*
10	**3**	*798*	**3**	*805*	**3**	*836*	**3**	*844*	**3**	*842*	**3**	*858*	**3**	*832*	**3**	*843*	**3**	*878*	**3**	*843*
11	**7**	*586*	**6**	*687*	**6**	*713*	**5**	*736*	**5**	*700*	**5**	*721*	**5**	*753*	**5**	*746*	**5**	*689*	**5**	*734*
12	**2**	*874*	**2**	*962*	**2**	*989*	**2**	*1021*	**2**	*974*	**2**	*938*	**2**	*965*	**2**	*950*	**2**	*1048*	**2**	*997*
13	**30**	*114*	**30**	*126*	**30**	*90*	**30**	*119*	**30**	*109*	**30**	*130*	**28**	*149*	**30**	*129*	**29**	*140*	**30**	*126*
14	**19**	*340*	**18**	*392*	**21**	*363*	**21**	*334*	**20**	*371*	**19**	*372*	**21**	*333*	**20**	*362*	**21**	*363*	**20**	*352*
15	**27**	*161*	**27**	*158*	**29**	*146*	**29**	*134*	**27**	*146*	**28**	*138*	**29**	*146*	**29**	*145*	**27**	*154*	**29**	*131*
5001	**4**	*709*	**4**	*765*	**4**	*762*	**4**	*798*	**4**	*784*	**4**	*803*	**4**	*786*	**4**	*773*	**4**	*815*	**4**	*794*
5002	**23**	*246*	**24**	*232*	**25**	*233*	**25**	*245*	**25**	*238*	**25**	*227*	**25**	*243*	**24**	*237*	**23**	*246*	**23**	*265*
5003	**15**	*418*	**14**	*444*	**12**	*505*	**11**	*543*	**12**	*510*	**14**	*520*	**13**	*493*	**11**	*532*	**14**	*485*	**14**	*504*
5004	**22**	*260*	**23**	*264*	**22**	*285*	**23**	*272*	**23**	*275*	**22**	*302*	**22**	*282*	**22**	*289*	**22**	*313*	**22**	*271*
5005	**8**	*537*	**7**	*628*	**8**	*630*	**8**	*615*	**7**	*649*	**8**	*615*	**8**	*642*	**8**	*630*	**6**	*670*	**7**	*637*
5006	**20**	*326*	**21**	*345*	**19**	*374*	**19**	*384*	**19**	*400*	**21**	*325*	**19**	*375*	**19**	*364*	**20**	*371*	**21**	*333*
5007	**25**	*216*	**22**	*266*	**23**	*269*	**22**	*277*	**22**	*279*	**23**	*275*	**23**	*276*	**23**	*255*	**24**	*241*	**24**	*251*
5008	**12**	*468*	**16**	*427*	**13**	*505*	**16**	*454*	**14**	*491*	**15**	*482*	**15**	*481*	**14**	*492*	**15**	*485*	**15**	*474*
5009	**11**	*473*	**12**	*530*	**14**	*503*	**13**	*521*	**11**	*537*	**12**	*535*	**11**	*522*	**10**	*533*	**13**	*497*	**10**	*569*
5010	**16**	*394*	**15**	*427*	**16**	*437*	**15**	*470*	**16**	*431*	**16**	*474*	**17**	*449*	**17**	*438*	**16**	*481*	**16**	*432*
5011	**28**	*138*	**29**	*136*	**27**	*153*	**27**	*155*	**28**	*137*	**29**	*130*	**30**	*132*	**27**	*155*	**28**	*140*	**27**	*162*
5012	**18**	*359*	**19**	*378*	**18**	*416*	**17**	*432*	**17**	*425*	**17**	*449*	**18**	*404*	**16**	*439*	**18**	*417*	**18**	*412*
5013	**13**	*459*	**13**	*507*	**15**	*478*	**12**	*532*	**15**	*465*	**13**	*527*	**12**	*510*	**15**	*483*	**12**	*526*	**9**	*573*
5014	**1**	*1151*	**1**	*1234*	**1**	*1259*	**1**	*1292*	**1**	*1269*	**1**	*1279*	**1**	*1278*	**1**	*1323*	**1**	*1255*	**1**	*1275*
5015	**21**	*295*	**20**	*377*	**20**	*369*	**20**	*360*	**21**	*335*	**20**	*345*	**20**	*368*	**21**	*358*	**19**	*377*	**19**	*366*

(c) Tanh-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**15**	*306*	**12**	*411*	**13**	*410*	**13**	*412*	**12**	*436*	**14**	*423*	**12**	*429*	**13**	*427*	**14**	*431*	**12**	*462*
2	**25**	*144*	**25**	*174*	**24**	*184*	**24**	*194*	**24**	*180*	**25**	*187*	**24**	*198*	**24**	*197*	**24**	*197*	**25**	*190*
3	**17**	*285*	**15**	*347*	**17**	*347*	**18**	*326*	**18**	*330*	**17**	*343*	**17**	*362*	**19**	*356*	**17**	*362*	**16**	*392*
4	**9**	*351*	**9**	*455*	**9**	*468*	**10**	*477*	**9**	*499*	**9**	*491*	**10**	*501*	**9**	*510*	**10**	*481*	**10**	*489*
5	**26**	*142*	**26**	*155*	**26**	*171*	**26**	*168*	**26**	*158*	**26**	*185*	**25**	*178*	**25**	*184*	**26**	*176*	**26**	*170*
6	**28**	*89*	**29**	*103*	**28**	*113*	**28**	*115*	**29**	*105*	**28**	*109*	**29**	*99*	**28**	*104*	**28**	*117*	**28**	*107*
7	**7**	*384*	**5**	*603*	**5**	*560*	**6**	*613*	**6**	*588*	**5**	*610*	**6**	*611*	**5**	*643*	**6**	*594*	**6**	*620*
8	**10**	*342*	**11**	*423*	**10**	*452*	**9**	*506*	**10**	*474*	**10**	*479*	**9**	*501*	**10**	*499*	**9**	*500*	**9**	*499*
9	**5**	*446*	**8**	*503*	**7**	*526*	**7**	*564*	**7**	*548*	**7**	*574*	**7**	*576*	**7**	*561*	**8**	*531*	**8**	*524*
10	**4**	*506*	**3**	*661*	**4**	*648*	**3**	*712*	**4**	*742*	**4**	*685*	**4**	*727*	**4**	*736*	**3**	*721*	**3**	*801*
11	**6**	*420*	**6**	*588*	**6**	*527*	**5**	*618*	**5**	*629*	**6**	*607*	**5**	*678*	**6**	*636*	**5**	*664*	**5**	*626*
12	**2**	*668*	**2**	*837*	**2**	*893*	**2**	*866*	**2**	*913*	**2**	*929*	**2**	*895*	**2**	*874*	**2**	*903*	**2**	*969*
13	**30**	*75*	**30**	*86*	**30**	*72*	**30**	*77*	**30**	*84*	**30**	*93*	**30**	*84*	**30**	*83*	**30**	*90*	**30**	*86*
14	**20**	*235*	**20**	*283*	**18**	*334*	**19**	*307*	**21**	*268*	**19**	*330*	**21**	*299*	**20**	*312*	**20**	*320*	**21**	*286*
15	**27**	*92*	**28**	*115*	**27**	*118*	**29**	*111*	**27**	*130*	**29**	*106*	**27**	*123*	**27**	*117*	**27**	*122*	**29**	*105*
5001	**3**	*514*	**4**	*656*	**3**	*714*	**4**	*699*	**3**	*770*	**3**	*714*	**3**	*732*	**3**	*776*	**4**	*678*	**4**	*729*
5002	**23**	*168*	**24**	*176*	**25**	*172*	**25**	*180*	**25**	*164*	**24**	*194*	**26**	*168*	**26**	*184*	**25**	*178*	**24**	*195*
5003	**11**	*336*	**16**	*335*	**15**	*390*	**16**	*387*	**14**	*414*	**13**	*428*	**15**	*396*	**11**	*435*	**15**	*410*	**13**	*435*
5004	**22**	*196*	**23**	*179*	**22**	*238*	**23**	*213*	**23**	*207*	**23**	*229*	**23**	*201*	**22**	*224*	**23**	*201*	**22**	*222*
5005	**8**	*384*	**7**	*516*	**8**	*515*	**8**	*561*	**8**	*530*	**8**	*529*	**8**	*520*	**8**	*536*	**7**	*547*	**7**	*549*
5006	**19**	*252*	**18**	*318*	**21**	*294*	**20**	*300*	**17**	*330*	**18**	*338*	**19**	*314*	**18**	*356*	**18**	*339*	**19**	*309*
5007	**24**	*151*	**22**	*215*	**23**	*223*	**22**	*214*	**22**	*216*	**22**	*237*	**22**	*212*	**23**	*218*	**22**	*222*	**23**	*218*
5008	**12**	*336*	**10**	*425*	**14**	*395*	**15**	*389*	**13**	*426*	**16**	*381*	**14**	*417*	**16**	*383*	**16**	*392*	**15**	*397*
5009	**14**	*330*	**14**	*391*	**12**	*412*	**12**	*433*	**15**	*410*	**12**	*451*	**13**	*419*	**14**	*401*	**12**	*455*	**14**	*433*
5010	**18**	*269*	**17**	*331*	**16**	*389*	**14**	*400*	**16**	*408*	**15**	*391*	**16**	*378*	**15**	*386*	**13**	*441*	**17**	*391*
5011	**29**	*85*	**27**	*116*	**29**	*113*	**27**	*115*	**28**	*111*	**27**	*128*	**28**	*120*	**29**	*99*	**29**	*100*	**27**	*118*
5012	**16**	*300*	**19**	*310*	**19**	*329*	**17**	*361*	**19**	*317*	**20**	*323*	**18**	*361*	**17**	*367*	**19**	*338*	**18**	*325*
5013	**13**	*332*	**13**	*398*	**11**	*443*	**11**	*438*	**11**	*451*	**11**	*469*	**11**	*457*	**12**	*435*	**11**	*471*	**11**	*479*
5014	**1**	*888*	**1**	*1169*	**1**	*1167*	**1**	*1192*	**1**	*1230*	**1**	*1183*	**1**	*1270*	**1**	*1186*	**1**	*1201*	**1**	*1225*
5015	**21**	*200*	**21**	*259*	**20**	*307*	**21**	*290*	**20**	*282*	**21**	*285*	**20**	*305*	**21**	*285*	**21**	*278*	**20**	*295*

Continued on Next Page...

Table B.2 – *Continued*

(d) Threshold-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**13**	*425*	**12**	*483*	**12**	*505*	**12**	*480*	**12**	*509*	**12**	*504*	**13**	*485*	**14**	*462*	**14**	*479*	**11**	*506*
2	**25**	*202*	**26**	*192*	**25**	*217*	**24**	*200*	**25**	*225*	**24**	*213*	**25**	*210*	**24**	*220*	**25**	*224*	**26**	*205*
3	**16**	*355*	**17**	*391*	**17**	*423*	**17**	*421*	**18**	*385*	**17**	*401*	**18**	*390*	**17**	*392*	**17**	*387*	**17**	*403*
4	**10**	*442*	**10**	*492*	**9**	*524*	**9**	*525*	**9**	*523*	**10**	*529*	**11**	*522*	**11**	*496*	**9**	*547*	**9**	*556*
5	**26**	*173*	**24**	*235*	**26**	*215*	**26**	*196*	**26**	*191*	**26**	*193*	**26**	*196*	**26**	*192*	**26**	*203*	**25**	*210*
6	**28**	*108*	**27**	*132*	**29**	*131*	**29**	*124*	**28**	*133*	**28**	*134*	**29**	*131*	**30**	*117*	**28**	*134*	**28**	*141*
7	**5**	*589*	**7**	*570*	**6**	*659*	**5**	*655*	**6**	*683*	**5**	*687*	**6**	*658*	**6**	*670*	**6**	*690*	**6**	*706*
8	**15**	*374*	**9**	*516*	**11**	*506*	**10**	*509*	**11**	*513*	**9**	*540*	**12**	*496*	**9**	*521*	**10**	*518*	**10**	*520*
9	**7**	*543*	**8**	*567*	**8**	*580*	**7**	*600*	**7**	*655*	**8**	*577*	**7**	*621*	**8**	*594*	**7**	*625*	**8**	*605*
10	**3**	*687*	**3**	*773*	**3**	*800*	**3**	*748*	**3**	*782*	**3**	*812*	**4**	*785*	**3**	*784*	**3**	*793*	**4**	*740*
11	**6**	*545*	**5**	*649*	**5**	*664*	**6**	*650*	**5**	*737*	**6**	*684*	**5**	*694*	**5**	*687*	**5**	*736*	**5**	*717*
12	**2**	*827*	**2**	*916*	**2**	*924*	**2**	*939*	**2**	*953*	**2**	*952*	**2**	*953*	**2**	*1013*	**2**	*948*	**2**	*972*
13	**29**	*101*	**30**	*115*	**30**	*110*	**30**	*117*	**30**	*118*	**30**	*94*	**28**	*132*	**29**	*127*	**29**	*117*	**30**	*96*
14	**19**	*323*	**20**	*335*	**19**	*346*	**21**	*299*	**19**	*364*	**19**	*350*	**20**	*342*	**19**	*353*	**19**	*326*	**20**	*348*
15	**27**	*126*	**29**	*127*	**27**	*137*	**27**	*130*	**29**	*123*	**29**	*133*	**30**	*126*	**28**	*133*	**30**	*117*	**27**	*146*
5001	**4**	*654*	**4**	*750*	**4**	*708*	**4**	*720*	**4**	*761*	**4**	*718*	**3**	*788*	**4**	*746*	**4**	*766*	**3**	*761*
5002	**24**	*208*	**25**	*209*	**24**	*228*	**25**	*200*	**23**	*250*	**25**	*196*	**24**	*224*	**25**	*213*	**24**	*233*	**24**	*233*
5003	**11**	*436*	**15**	*419*	**16**	*440*	**16**	*441*	**14**	*500*	**14**	*449*	**15**	*457*	**15**	*456*	**12**	*501*	**14**	*464*
5004	**23**	*216*	**23**	*263*	**23**	*252*	**22**	*283*	**24**	*248*	**22**	*291*	**22**	*255*	**22**	*275*	**23**	*249*	**23**	*265*
5005	**8**	*536*	**6**	*595*	**7**	*589*	**8**	*542*	**8**	*584*	**7**	*612*	**8**	*580*	**7**	*620*	**8**	*606*	**7**	*627*
5006	**20**	*280*	**19**	*340*	**21**	*313*	**19**	*321*	**20**	*308*	**21**	*325*	**19**	*356*	**20**	*350*	**20**	*316*	**19**	*351*
5007	**22**	*216*	**22**	*266*	**22**	*260*	**23**	*258*	**22**	*274*	**23**	*236*	**23**	*240*	**23**	*239*	**22**	*264*	**22**	*265*
5008	**14**	*382*	**14**	*428*	**14**	*444*	**13**	*460*	**15**	*449*	**15**	*437*	**14**	*472*	**13**	*487*	**15**	*468*	**16**	*422*
5009	**9**	*473*	**13**	*479*	**10**	*511*	**11**	*484*	**13**	*503*	**11**	*510*	**9**	*527*	**10**	*502*	**11**	*518*	**13**	*494*
5010	**17**	*345*	**16**	*402*	**15**	*442*	**15**	*448*	**16**	*418*	**16**	*418*	**16**	*409*	**16**	*403*	**16**	*460*	**15**	*424*
5011	**30**	*99*	**28**	*130*	**28**	*133*	**28**	*126*	**27**	*137*	**27**	*150*	**27**	*144*	**27**	*147*	**27**	*136*	**29**	*135*
5012	**18**	*335*	**18**	*381*	**18**	*383*	**18**	*387*	**17**	*393*	**18**	*396*	**17**	*399*	**18**	*373*	**18**	*372*	**18**	*362*
5013	**12**	*428*	**11**	*491*	**13**	*480*	**14**	*458*	**10**	*514*	**13**	*478*	**10**	*523*	**12**	*495*	**13**	*492*	**12**	*501*
5014	**1**	*1073*	**1**	*1236*	**1**	*1160*	**1**	*1247*	**1**	*1170*	**1**	*1262*	**1**	*1305*	**1**	*1336*	**1**	*1245*	**1**	*1222*
5015	**21**	*279*	**21**	*310*	**20**	*326*	**20**	*318*	**21**	*301*	**20**	*328*	**21**	*316*	**21**	*331*	**21**	*303*	**21**	*335*

Table B.3: The complete list of synthetic data set 1 genes in the population size 300.

(a) Sigmoid-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**10**	*455*	**12**	*576*	**11**	*608*	**13**	*588*	**10**	*614*	**15**	*560*	**14**	*560*	**12**	*587*	**14**	*552*	**13**	*577*
2	**26**	*187*	**24**	*267*	**24**	*287*	**24**	*269*	**25**	*263*	**23**	*296*	**24**	*279*	**24**	*270*	**24**	*260*	**24**	*270*
3	**17**	*371*	**17**	*483*	**16**	*487*	**17**	*479*	**18**	*474*	**17**	*476*	**17**	*469*	**17**	*489*	**17**	*492*	**17**	*491*
4	**8**	*478*	**10**	*604*	**10**	*618*	**12**	*595*	**12**	*580*	**9**	*675*	**9**	*655*	**10**	*634*	**10**	*653*	**10**	*642*
5	**25**	*201*	**26**	*238*	**26**	*251*	**25**	*254*	**26**	*253*	**26**	*224*	**26**	*242*	**25**	*264*	**25**	*253*	**26**	*230*
6	**29**	*112*	**28**	*158*	**29**	*147*	**28**	*167*	**29**	*172*	**27**	*167*	**28**	*147*	**28**	*155*	**29**	*159*	**30**	*155*
7	**7**	*522*	**5**	*824*	**6**	*810*	**6**	*807*	**6**	*816*	**5**	*866*	**6**	*796*	**6**	*778*	**6**	*787*	**6**	*825*
8	**11**	*454*	**9**	*627*	**9**	*642*	**9**	*653*	**9**	*646*	**10**	*653*	**10**	*638*	**9**	*666*	**9**	*667*	**9**	*647*
9	**5**	*591*	**7**	*706*	**8**	*731*	**8**	*701*	**7**	*745*	**7**	*732*	**8**	*751*	**7**	*717*	**7**	*738*	**8**	*728*
10	**4**	*684*	**3**	*892*	**3**	*908*	**3**	*947*	**3**	*922*	**4**	*931*	**5**	*897*	**3**	*992*	**4**	*895*	**3**	*907*
11	**6**	*577*	**6**	*795*	**5**	*839*	**5**	*820*	**5**	*841*	**6**	*844*	**4**	*900*	**5**	*885*	**5**	*874*	**5**	*851*
12	**2**	*799*	**2**	*1081*	**2**	*1112*	**2**	*1068*	**2**	*1130*	**2**	*1066*	**2**	*1140*	**2**	*1102*	**2**	*1132*	**2**	*1113*
13	**30**	*89*	**30**	*127*	**30**	*128*	**30**	*139*	**30**	*122*	**29**	*143*	**29**	*142*	**29**	*153*	**30**	*134*	**29**	*158*
14	**19**	*339*	**19**	*414*	**20**	*392*	**20**	*420*	**19**	*440*	**19**	*411*	**19**	*438*	**20**	*412*	**20**	*402*	**20**	*395*
15	**28**	*118*	**27**	*165*	**27**	*171*	**27**	*177*	**27**	*183*	**28**	*164*	**27**	*169*	**27**	*162*	**27**	*175*	**27**	*170*
5001	**3**	*713*	**4**	*852*	**4**	*897*	**4**	*903*	**4**	*901*	**3**	*991*	**3**	*919*	**4**	*942*	**3**	*942*	**4**	*891*
5002	**22**	*221*	**25**	*251*	**25**	*283*	**26**	*238*	**24**	*270*	**25**	*242*	**25**	*256*	**26**	*256*	**26**	*250*	**25**	*267*
5003	**13**	*394*	**13**	*567*	**15**	*533*	**14**	*545*	**14**	*563*	**13**	*588*	**15**	*551*	**16**	*541*	**12**	*585*	**12**	*594*
5004	**24**	*202*	**22**	*322*	**22**	*330*	**22**	*318*	**23**	*321*	**22**	*301*	**22**	*330*	**22**	*324*	**22**	*339*	**22**	*321*
5005	**9**	*478*	**8**	*705*	**7**	*742*	**7**	*703*	**8**	*705*	**8**	*700*	**7**	*762*	**8**	*701*	**8**	*705*	**7**	*754*
5006	**20**	*284*	**20**	*410*	**19**	*410*	**21**	*416*	**20**	*428*	**20**	*394*	**20**	*430*	**19**	*442*	**18**	*451*	**19**	*422*
5007	**23**	*220*	**23**	*290*	**23**	*298*	**23**	*305*	**22**	*345*	**24**	*282*	**23**	*286*	**23**	*284*	**23**	*313*	**23**	*304*
5008	**12**	*398*	**14**	*533*	**14**	*539*	**16**	*515*	**15**	*557*	**16**	*533*	**13**	*584*	**11**	*597*	**16**	*518*	**14**	*575*
5009	**14**	*388*	**16**	*523*	**12**	*591*	**11**	*601*	**13**	*573*	**12**	*590*	**12**	*613*	**13**	*570*	**11**	*635*	**11**	*634*
5010	**16**	*378*	**15**	*530*	**17**	*485*	**17**	*527*	**16**	*534*	**14**	*562*	**16**	*551*	**14**	*558*	**15**	*522*	**16**	*536*
5011	**27**	*133*	**29**	*157*	**28**	*165*	**29**	*144*	**28**	*172*	**30**	*133*	**30**	*141*	**30**	*141*	**28**	*160*	**28**	*163*
5012	**18**	*343*	**18**	*479*	**18**	*435*	**18**	*463*	**17**	*493*	**18**	*449*	**18**	*457*	**18**	*476*	**19**	*448*	**18**	*484*
5013	**15**	*384*	**11**	*598*	**13**	*563*	**10**	*606*	**11**	*602*	**11**	*609*	**11**	*624*	**15**	*558*	**13**	*584*	**15**	*563*
5014	**1**	*1148*	**1**	*1445*	**1**	*1452*	**1**	*1463*	**1**	*1495*	**1**	*1454*	**1**	*1453*	**1**	*1484*	**1**	*1488*	**1**	*1529*
5015	**21**	*284*	**21**	*359*	**21**	*371*	**19**	*430*	**21**	*405*	**21**	*394*	**21**	*420*	**21**	*407*	**21**	*393*	**21**	*364*

Continued on Next Page...

Table B.3 – *Continued*

(b) Linear-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**10**	*587*	**10**	*647*	**11**	*653*	**12**	*623*	**11**	*644*	**14**	*613*	**14**	*600*	**12**	*644*	**10**	*650*	**11**	*648*
2	**25**	*246*	**24**	*300*	**25**	*289*	**24**	*297*	**24**	*279*	**24**	*332*	**25**	*277*	**24**	*313*	**25**	*307*	**24**	*309*
3	**17**	*465*	**17**	*545*	**18**	*505*	**16**	*544*	**17**	*540*	**16**	*534*	**17**	*518*	**17**	*500*	**18**	*513*	**17**	*528*
4	**13**	*556*	**13**	*622*	**10**	*660*	**9**	*684*	**10**	*652*	**9**	*661*	**10**	*677*	**11**	*652*	**14**	*628*	**12**	*638*
5	**26**	*245*	**26**	*245*	**26**	*240*	**26**	*255*	**25**	*268*	**26**	*263*	**26**	*265*	**26**	*267*	**26**	*258*	**25**	*263*
6	**27**	*166*	**29**	*174*	**29**	*170*	**27**	*195*	**27**	*195*	**28**	*186*	**28**	*177*	**28**	*183*	**28**	*185*	**28**	*164*
7	**7**	*735*	**5**	*898*	**5**	*889*	**6**	*867*	**5**	*889*	**6**	*825*	**6**	*875*	**6**	*841*	**5**	*862*	**6**	*858*
8	**9**	*608*	**11**	*631*	**13**	*641*	**10**	*651*	**9**	*661*	**12**	*618*	**9**	*689*	**10**	*659*	**9**	*669*	**10**	*661*
9	**6**	*741*	**7**	*771*	**7**	*782*	**7**	*770*	**7**	*799*	**7**	*767*	**8**	*731*	**7**	*807*	**8**	*784*	**7**	*830*
10	**3**	*879*	**3**	*961*	**3**	*935*	**3**	*1023*	**3**	*996*	**3**	*983*	**3**	*990*	**4**	*918*	**4**	*965*	**3**	*953*
11	**5**	*746*	**6**	*892*	**6**	*856*	**5**	*904*	**6**	*818*	**5**	*875*	**5**	*916*	**5**	*902*	**6**	*810*	**4**	*934*
12	**2**	*1004*	**2**	*1103*	**2**	*1176*	**2**	*1069*	**2**	*1101*	**2**	*1145*	**2**	*1093*	**2**	*1140*	**2**	*1081*	**2**	*1121*
13	**29**	*150*	**30**	*143*	**28**	*180*	**30**	*143*	**30**	*163*	**30**	*143*	**30**	*123*	**30**	*140*	**29**	*174*	**29**	*156*
14	**20**	*400*	**19**	*439*	**21**	*442*	**21**	*440*	**20**	*461*	**19**	*439*	**19**	*462*	**21**	*421*	**19**	*457*	**19**	*476*
15	**28**	*165*	**27**	*188*	**27**	*213*	**28**	*181*	**29**	*171*	**27**	*189*	**27**	*190*	**27**	*187*	**27**	*198*	**27**	*172*
5001	**4**	*819*	**4**	*923*	**4**	*914*	**4**	*926*	**4**	*955*	**4**	*935*	**4**	*926*	**3**	*961*	**3**	*975*	**5**	*926*
5002	**24**	*280*	**25**	*276*	**24**	*301*	**25**	*292*	**26**	*259*	**25**	*280*	**24**	*292*	**25**	*295*	**24**	*309*	**26**	*250*
5003	**12**	*561*	**15**	*574*	**12**	*642*	**13**	*608*	**14**	*588*	**15**	*597*	**15**	*593*	**13**	*620*	**13**	*634*	**13**	*621*
5004	**22**	*319*	**22**	*355*	**22**	*333*	**22**	*337*	**22**	*360*	**22**	*366*	**22**	*366*	**22**	*392*	**22**	*338*	**22**	*378*
5005	**8**	*672*	**8**	*762*	**8**	*750*	**8**	*766*	**8**	*756*	**8**	*762*	**7**	*776*	**8**	*777*	**7**	*786*	**8**	*730*
5006	**21**	*385*	**20**	*421*	**19**	*452*	**19**	*446*	**21**	*430*	**20**	*436*	**20**	*428*	**20**	*423*	**21**	*411*	**20**	*445*
5007	**23**	*292*	**23**	*330*	**23**	*313*	**23**	*330*	**23**	*324*	**23**	*356*	**23**	*346*	**23**	*331*	**23**	*331*	**23**	*320*
5008	**15**	*533*	**14**	*589*	**15**	*584*	**15**	*578*	**16**	*561*	**11**	*619*	**11**	*626*	**14**	*616*	**15**	*567*	**14**	*610*
5009	**11**	*567*	**12**	*623*	**14**	*629*	**11**	*628*	**13**	*596*	**10**	*646*	**12**	*621*	**9**	*666*	**11**	*649*	**9**	*673*
5010	**16**	*496*	**16**	*567*	**16**	*554*	**17**	*508*	**15**	*586*	**18**	*526*	**16**	*593*	**16**	*555*	**16**	*530*	**16**	*570*
5011	**30**	*145*	**28**	*186*	**30**	*164*	**29**	*178*	**28**	*194*	**29**	*172*	**29**	*148*	**29**	*166*	**30**	*160*	**30**	*153*
5012	**18**	*449*	**18**	*500*	**17**	*539*	**18**	*494*	**18**	*507*	**17**	*534*	**18**	*514*	**18**	*486*	**17**	*521*	**18**	*511*
5013	**14**	*549*	**9**	*650*	**9**	*661*	**14**	*585*	**12**	*624*	**13**	*614*	**13**	*601*	**15**	*596*	**12**	*636*	**15**	*584*
5014	**1**	*1262*	**1**	*1460*	**1**	*1429*	**1**	*1500*	**1**	*1436*	**1**	*1481*	**1**	*1476*	**1**	*1455*	**1**	*1479*	**1**	*1479*
5015	**19**	*407*	**21**	*404*	**20**	*450*	**20**	*442*	**19**	*473*	**21**	*407*	**21**	*402*	**19**	*440*	**20**	*452*	**21**	*430*

(c) Tanh-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**14**	*353*	**13**	*509*	**12**	*550*	**13**	*543*	**14**	*544*	**12**	*564*	**13**	*553*	**11**	*566*	**12**	*552*	**12**	*566*
2	**25**	*161*	**24**	*242*	**24**	*252*	**25**	*237*	**25**	*254*	**26**	*235*	**25**	*248*	**24**	*273*	**25**	*249*	**24**	*246*
3	**18**	*303*	**18**	*454*	**18**	*458*	**17**	*474*	**18**	*467*	**17**	*474*	**17**	*484*	**17**	*450*	**18**	*434*	**17**	*453*
4	**10**	*394*	**9**	*545*	**10**	*584*	**9**	*618*	**10**	*579*	**11**	*569*	**9**	*649*	**10**	*587*	**11**	*561*	**10**	*593*
5	**24**	*170*	**26**	*221*	**26**	*218*	**26**	*231*	**26**	*235*	**25**	*239*	**26**	*227*	**26**	*241*	**26**	*223*	**26**	*211*
6	**29**	*92*	**28**	*151*	**28**	*147*	**27**	*156*	**28**	*135*	**27**	*168*	**28**	*135*	**28**	*154*	**27**	*168*	**27**	*158*
7	**6**	*476*	**6**	*742*	**6**	*760*	**5**	*794*	**6**	*766*	**6**	*762*	**6**	*798*	**5**	*815*	**4**	*830*	**6**	*775*
8	**12**	*364*	**11**	*534*	**9**	*640*	**10**	*602*	**9**	*644*	**9**	*612*	**11**	*593*	**9**	*630*	**9**	*606*	**9**	*620*
9	**8**	*438*	**7**	*663*	**8**	*679*	**8**	*686*	**8**	*702*	**7**	*683*	**7**	*710*	**8**	*720*	**7**	*728*	**8**	*701*
10	**4**	*575*	**5**	*821*	**4**	*872*	**4**	*904*	**3**	*912*	**3**	*925*	**5**	*831*	**4**	*885*	**5**	*828*	**4**	*929*
11	**5**	*498*	**4**	*833*	**5**	*808*	**6**	*787*	**5**	*789*	**5**	*837*	**4**	*839*	**6**	*809*	**6**	*821*	**5**	*827*
12	**2**	*724*	**2**	*1025*	**2**	*1064*	**2**	*1075*	**2**	*1069*	**2**	*1111*	**2**	*1108*	**2**	*1017*	**2**	*1115*	**2**	*1108*
13	**28**	*92*	**30**	*99*	**30**	*99*	**30**	*99*	**30**	*98*	**30**	*126*	**30**	*117*	**30**	*113*	**30**	*115*	**30**	*105*
14	**19**	*296*	**20**	*392*	**21**	*372*	**19**	*400*	**21**	*359*	**18**	*454*	**20**	*421*	**19**	*422*	**20**	*402*	**21**	*395*
15	**27**	*109*	**27**	*154*	**27**	*164*	**28**	*153*	**27**	*141*	**28**	*166*	**27**	*142*	**27**	*167*	**28**	*154*	**28**	*158*
5001	**3**	*609*	**3**	*879*	**3**	*915*	**3**	*930*	**4**	*911*	**4**	*860*	**3**	*933*	**3**	*909*	**3**	*932*	**3**	*951*
5002	**26**	*161*	**25**	*227*	**25**	*220*	**24**	*267*	**24**	*263*	**24**	*253*	**24**	*263*	**25**	*250*	**24**	*257*	**25**	*245*
5003	**15**	*330*	**14**	*497*	**14**	*531*	**16**	*496*	**13**	*545*	**16**	*513*	**14**	*550*	**15**	*530*	**16**	*497*	**15**	*510*
5004	**22**	*203*	**22**	*279*	**22**	*315*	**22**	*346*	**22**	*289*	**22**	*321*	**23**	*282*	**22**	*291*	**22**	*323*	**23**	*294*
5005	**7**	*441*	**8**	*654*	**7**	*703*	**7**	*695*	**7**	*712*	**8**	*669*	**8**	*663*	**7**	*729*	**8**	*706*	**7**	*741*
5006	**20**	*270*	**19**	*417*	**17**	*458*	**20**	*390*	**20**	*405*	**20**	*420*	**19**	*424*	**20**	*409*	**19**	*421*	**19**	*406*
5007	**23**	*202*	**23**	*275*	**23**	*254*	**23**	*288*	**23**	*287*	**23**	*298*	**22**	*309*	**23**	*281*	**23**	*288*	**22**	*300*
5008	**13**	*357*	**15**	*495*	**15**	*500*	**14**	*539*	**15**	*537*	**14**	*519*	**16**	*492*	**14**	*531*	**14**	*529*	**16**	*477*
5009	**11**	*371*	**12**	*521*	**13**	*545*	**12**	*570*	**12**	*563*	**13**	*539*	**10**	*598*	**13**	*532*	**13**	*551*	**13**	*546*
5010	**16**	*323*	**16**	*488*	**16**	*483*	**15**	*514*	**16**	*500*	**15**	*517*	**15**	*510*	**16**	*511*	**15**	*511*	**14**	*519*
5011	**30**	*89*	**29**	*145*	**29**	*129*	**29**	*134*	**29**	*134*	**29**	*131*	**29**	*132*	**29**	*129*	**29**	*141*	**29**	*144*
5012	**17**	*310*	**17**	*457*	**19**	*441*	**18**	*471*	**17**	*470*	**19**	*424*	**18**	*430*	**18**	*440*	**17**	*461*	**18**	*446*
5013	**9**	*401*	**10**	*538*	**11**	*571*	**11**	*572*	**11**	*570*	**10**	*600*	**12**	*584*	**12**	*565*	**10**	*588*	**11**	*574*
5014	**1**	*952*	**1**	*1445*	**1**	*1425*	**1**	*1483*	**1**	*1400*	**1**	*1486*	**1**	*1513*	**1**	*1455*	**1**	*1523*	**1**	*1494*
5015	**21**	*245*	**21**	*325*	**20**	*401*	**21**	*383*	**19**	*417*	**21**	*368*	**21**	*341*	**21**	*374*	**21**	*382*	**20**	*395*

Continued on Next Page...

Table B.3 – *Continued*

(d) Threshold-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	**12**	*517*	**15**	*565*	**13**	*593*	**10**	*617*	**11**	*633*	**11**	*611*	**12**	*601*	**11**	*611*	**12**	*606*	**12**	*617*
2	**25**	*235*	**24**	*287*	**24**	*290*	**25**	*258*	**25**	*284*	**24**	*271*	**24**	*267*	**25**	*296*	**24**	*299*	**24**	*314*
3	**18**	*407*	**18**	*488*	**17**	*501*	**18**	*473*	**17**	*498*	**17**	*534*	**17**	*515*	**17**	*496*	**17**	*505*	**18**	*475*
4	**9**	*574*	**10**	*610*	**10**	*655*	**11**	*612*	**12**	*617*	**9**	*648*	**9**	*651*	**9**	*657*	**10**	*640*	**9**	*685*
5	**24**	*238*	**26**	*250*	**26**	*244*	**26**	*243*	**24**	*285*	**26**	*253*	**26**	*245*	**26**	*270*	**26**	*255*	**26**	*262*
6	**28**	*135*	**28**	*182*	**27**	*196*	**27**	*168*	**27**	*189*	**28**	*172*	**27**	*166*	**27**	*187*	**27**	*181*	**27**	*183*
7	**6**	*667*	**6**	*783*	**6**	*786*	**6**	*786*	**6**	*801*	**6**	*822*	**6**	*842*	**6**	*810*	**6**	*788*	**5**	*837*
8	**15**	*504*	**9**	*655*	**9**	*682*	**9**	*639*	**9**	*650*	**12**	*604*	**10**	*638*	**10**	*653*	**9**	*642*	**11**	*634*
9	**8**	*626*	**7**	*713*	**8**	*753*	**8**	*717*	**7**	*774*	**7**	*718*	**7**	*785*	**7**	*777*	**8**	*742*	**8**	*707*
10	**4**	*811*	**4**	*900*	**3**	*956*	**4**	*938*	**3**	*1005*	**3**	*978*	**3**	*976*	**3**	*959*	**4**	*932*	**3**	*951*
11	**5**	*710*	**5**	*810*	**5**	*838*	**5**	*874*	**5**	*814*	**5**	*843*	**5**	*876*	**5**	*856*	**5**	*826*	**6**	*830*
12	**2**	*959*	**2**	*1102*	**2**	*1122*	**2**	*1154*	**2**	*1155*	**2**	*1150*	**2**	*1092*	**2**	*1111*	**2**	*1068*	**2**	*1083*
13	**29**	*132*	**30**	*155*	**30**	*135*	**30**	*139*	**30**	*154*	**30**	*137*	**29**	*141*	**30**	*131*	**29**	*153*	**30**	*134*
14	**20**	*360*	**19**	*467*	**19**	*407*	**20**	*398*	**19**	*445*	**19**	*439*	**20**	*399*	**21**	*418*	**20**	*419*	**21**	*405*
15	**27**	*146*	**29**	*175*	**28**	*168*	**28**	*164*	**28**	*170*	**27**	*174*	**28**	*162*	**28**	*186*	**28**	*167*	**29**	*178*
5001	**3**	*816*	**3**	*915*	**4**	*915*	**3**	*974*	**4**	*889*	**4**	*899*	**4**	*925*	**4**	*889*	**3**	*965*	**4**	*950*
5002	**26**	*211*	**25**	*255*	**25**	*281*	**24**	*277*	**26**	*275*	**25**	*271*	**25**	*247*	**24**	*303*	**25**	*266*	**25**	*288*
5003	**10**	*536*	**13**	*576*	**14**	*588*	**14**	*578*	**14**	*563*	**15**	*574*	**15**	*572*	**12**	*609*	**14**	*576*	**14**	*587*
5004	**22**	*326*	**22**	*315*	**22**	*336*	**22**	*352*	**22**	*327*	**22**	*343*	**22**	*313*	**22**	*330*	**22**	*366*	**22**	*349*
5005	**7**	*650*	**8**	*707*	**7**	*767*	**7**	*732*	**8**	*732*	**8**	*698*	**8**	*742*	**8**	*687*	**7**	*745*	**7**	*769*
5006	**21**	*353*	**20**	*416*	**21**	*401*	**19**	*433*	**20**	*420*	**20**	*434*	**19**	*410*	**20**	*429*	**21**	*393*	**20**	*434*
5007	**23**	*263*	**23**	*315*	**23**	*296*	**23**	*315*	**23**	*318*	**23**	*307*	**23**	*307*	**23**	*324*	**23**	*313*	**23**	*334*
5008	**11**	*521*	**11**	*588*	**11**	*625*	**15**	*567*	**13**	*575*	**14**	*575*	**11**	*601*	**15**	*555*	**15**	*574*	**15**	*556*
5009	**14**	*506*	**14**	*570*	**15**	*566*	**12**	*600*	**10**	*641*	**10**	*618*	**13**	*600*	**14**	*592*	**11**	*628*	**10**	*654*
5010	**16**	*444*	**16**	*549*	**16**	*525*	**16**	*522*	**16**	*529*	**16**	*548*	**16**	*535*	**16**	*527*	**16**	*542*	**16**	*520*
5011	**30**	*126*	**27**	*183*	**29**	*152*	**29**	*151*	**29**	*156*	**29**	*165*	**30**	*129*	**29**	*154*	**30**	*150*	**28**	*180*
5012	**17**	*408*	**17**	*491*	**18**	*473*	**17**	*516*	**18**	*462*	**18**	*443*	**18**	*467*	**18**	*490*	**18**	*467*	**17**	*497*
5013	**13**	*511*	**12**	*582*	**12**	*616*	**13**	*592*	**15**	*560*	**13**	*576*	**14**	*588*	**13**	*594*	**13**	*593*	**13**	*589*
5014	**1**	*1262*	**1**	*1457*	**1**	*1425*	**1**	*1458*	**1**	*1427*	**1**	*1473*	**1**	*1420*	**1**	*1484*	**1**	*1487*	**1**	*1411*
5015	**19**	*370*	**21**	*412*	**20**	*402*	**21**	*373*	**21**	*412*	**21**	*376*	**21**	*378*	**19**	*432*	**19**	*448*	**19**	*440*

Table B.4: The complete list of synthetic data set 2 genes in the population size 100.

(a) Sigmoid-based system

Gene	Fitness Evaluation																	
	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
11			2	56	3	99	2	120	1	165	3	125	2	167	3	172	2	181	2	177
12			3	50	2	122	1	136	3	133	1	161	1	179	2	180	1	183	1	185
16													6	54	8	50	8	55	7	52
17					6	53							7	54	6	61	6	72	6	61
19					4	79	4	79	4	95	4	98	4	110	4	138	4	104	4	125
21			1	76	1	123	3	115	2	142	2	147	3	143	1	188	3	159	3	177
23													8	51	7	51	7	66		
27					5	63	5	76	5	84	5	78	5	103	5	76	5	86	5	68

(b) Linear-based system

Gene	Fitness Evaluation																	
	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
11			3	150	3	209	3	222	3	260	3	255	3	282	3	268	3	299	3	273
12			2	185	1	235	2	264	2	330	1	324	1	328	1	355	1	357	1	371
16			9	55	8	81	7	104	7	128	7	116	7	113	7	109	8	111	7	133
17			6	72	6	99	6	123	6	136	6	132	6	150	6	124	6	136	6	136
18																	11	50		
19	2	58	4	128	4	164	4	206	4	214	4	213	4	226	4	224	4	200	4	210
20																	10	52		
21	1	66	1	191	2	234	1	267	1	341	2	312	2	309	2	340	2	338	2	325
23			7	61	7	88	8	95	9	84	8	113	8	109	8	105	7	128	8	124
24			8	59	9	66	9	78	8	93	9	75	9	70	9	85	9	87	9	73
26									10	56	10	55	10	57					10	51
27			5	76	5	132	5	163	5	170	5	171	5	191	5	168	5	187	5	183

(c) Tanh-based system

Gene	Fitness Evaluation																	
	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
11					1	60	2	72	3	84	1	101	2	123	3	125	3	127	3	123
12							1	81	2	93	2	96	1	130	1	151	1	154	1	144
17															6	51	6	51	5	61
19							4	50	4	51	4	63	4	75	4	83	4	90	4	91
21					2	58	3	63	1	95	3	94	3	104	2	144	2	130	2	130
27											5	55	5	60	5	52	5	78	6	59

(d) Threshold-based system

Gene	Fitness Evaluation																	
	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
11			1	104	2	154	2	173	2	186	3	170	3	196	3	233	3	218	3	236
12			3	94	3	150	1	188	1	191	1	210	1	258	1	240	1	265	2	262
16							7	64	7	70	7	68	7	82	7	87	8	69	7	88
17					5	91	6	75	6	79	6	93	6	93	6	103	6	101	6	105
19			4	84	4	132	4	127	4	144	4	119	4	145	4	152	4	150	4	168
21			2	102	1	165	3	173	3	185	2	206	2	231	2	237	2	251	1	267
23							8	61	8	68	8	64	8	68	8	86	7	84	8	86
24									9	52	9	56	9	56	9	54	9	61	9	55
27			5	59	6	88	5	109	5	85	5	104	5	119	5	120	5	126	5	137

Table B.5: The complete list of synthetic data set 2 genes in the population size 200. Genes marked in red denote noisy genes.

(a) Sigmoid-based system

Gene	Fitness Evaluation																	
	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
11			2	151	2	344	2	472	3	556	3	620	3	677	3	739	3	720	3	768
12			4	109	4	283	3	459	2	596	1	665	2	690	2	747	1	731	1	851
15							12	51	11	65	12	70	12	82	13	77	11	105	12	87
16					6	115	7	157	7	191	6	233	8	216	9	208	7	246	8	242
17			6	66	7	112	6	191	6	223	8	209	6	282	6	282	6	261	6	293
18							11	57	13	56	11	84	11	82	11	92	13	81	11	102
19			3	147	3	287	4	366	4	414	4	457	4	547	4	548	4	544	4	565
20											13	65	14	62	12	83	12	81	13	75
21			1	177	1	354	1	515	1	612	2	656	1	696	1	771	2	722	2	835
22							10	60	12	61	14	58	13	75	14	65	14	55	14	69
23					9	82	9	134	9	149	9	174	9	212	8	213	9	204	9	224
24			7	59	8	110	8	150	8	182	7	219	7	235	7	221	8	228	7	264
26									10	77	10	88	10	89	10	108	10	112	10	107
27			5	76	5	203	5	272	5	298	5	358	5	383	5	381	5	435	5	418
29													15	50			15	52		

Continued on Next Page...

Table B.5 – *Continued*

(b) Linear-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
11			3	454	3	752	3	853	3	950	3	1006	3	1067	3	1090	3	1104	3	1150
12			2	493	2	839	1	1048	1	1130	1	1212	1	1233	1	1315	1	1319	1	1430
13									18	58	19	55	17	70	16	71	18	57	20	54
14							16	56	16	68	16	67	16	83	18	62	16	65	17	82
15			13	61	12	98	11	117	12	127	12	125	12	119	11	157	12	147	11	162
16			6	237	6	358	6	424	7	420	6	503	7	484	6	509	6	517	6	570
17			8	202	7	351	7	398	6	476	7	454	6	500	7	482	7	507	7	531
18			11	63	10	112	10	138	10	140	11	142	11	150	10	167	10	168	12	153
19			4	408	4	628	4	712	4	768	4	867	4	874	4	852	4	846	4	889
20					15	57	14	100	14	103	13	108	14	116	14	105	13	124	13	146
21	1	58	1	553	1	887	2	1022	2	1123	2	1154	2	1214	2	1235	2	1251	2	1320
22			10	65	14	78	13	103	13	122	14	108	13	118	13	109	14	117	14	137
23			9	186	9	295	9	341	9	359	8	391	9	394	8	419	9	400	8	438
24			7	208	8	311	8	362	8	391	9	381	8	399	9	409	8	432	9	436
25									17	63	18	64	18	69	17	65	17	63	16	87
26			12	63	11	111	12	114	11	136	10	168	10	180	12	149	11	162	10	171
27			5	321	5	526	5	591	5	616	5	676	5	724	5	767	5	755	5	790
29					13	79	15	72	15	71	15	88	15	96	15	97	15	104	15	116
667											17	66			19	60	19	51	19	55
2471																			21	53
2828													20	50					22	50
4377													19	51	20	52	20	50	18	69

(c) Tanh-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
11			2	92	2	211	2	334	2	395	3	439	2	484	3	514	3	551	2	561
12			4	62	3	194	3	322	1	400	1	512	1	574	1	563	1	649	1	653
15											10	67			12	58	13	53	11	67
16					8	67	7	112	7	134	7	133	6	188	6	183	9	184	9	157
17					7	71	6	126	6	147	6	135	7	187	7	165	6	214	6	209
18											12	53	11	65	10	68	12	61	12	64
19			3	90	4	177	4	242	4	292	4	319	4	393	4	341	4	396	4	384
20													12	53	13	53	10	83	13	63
21			1	98	1	226	1	348	3	395	2	443	3	462	2	526	2	573	3	530
22																	14	53	14	59
23					9	57	9	84	9	110	9	131	9	151	8	165	7	199	8	188
24					6	77	8	103	8	123	8	131	8	184	9	148	8	192	7	206
26									10	55	11	63	10	83	11	62	11	77	10	71
27					5	108	5	156	5	211	5	250	5	294	5	261	5	286	5	303

(d) Threshold-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
11			3	302	3	594	3	679	3	745	3	737	3	824	3	856	3	903	3	907
12			2	316	2	613	2	736	1	877	1	896	1	940	1	1057	1	1035	1	1091
13																			16	63
14													16	54	17	51	16	55	17	59
15					12	81	11	93	12	96	12	87	12	96	13	114	12	105	13	105
16			7	133	7	252	6	295	7	311	7	315	7	362	7	375	7	375	7	383
17			6	156	6	262	7	291	6	336	6	358	6	369	6	384	6	392	6	416
18					13	66	12	73	10	116	10	116	11	129	11	124	11	117	10	155
19			4	256	4	477	4	525	4	585	4	593	4	611	4	646	4	649	4	673
20					14	60	13	67	14	82	14	73	13	89	14	92	13	98	12	107
21			1	368	1	695	1	758	2	869	2	851	2	936	2	976	2	958	2	1011
22					11	85	14	57	13	84	13	79	14	89	12	117	14	74	14	85
23			9	106	8	209	9	224	9	259	9	266	8	276	8	318	8	356	8	346
24			8	122	9	209	8	264	8	266	8	291	9	268	9	305	9	286	9	291
25													17	53	16	56	15	56	18	59
26					10	87	10	100	11	105	11	114	10	143	10	136	10	129	11	130
27			5	214	5	380	5	459	5	458	5	511	5	514	5	544	5	588	5	569
28																			15	69
29									15	55			15	57	15	57				

Table B.6: The complete list of synthetic data set 2 genes in the population size 300. Genes marked in red denote noisy genes.

(a) Sigmoid-based system

Gene Index	Fitness Evaluation 5000 Rank	5000 Freq.	10000 Rank	10000 Freq.	15000 Rank	15000 Freq.	20000 Rank	20000 Freq.	25000 Rank	25000 Freq.	30000 Rank	30000 Freq.	35000 Rank	35000 Freq.	40000 Rank	40000 Freq.	45000 Rank	45000 Freq.	50000 Rank	50000 Freq.
11			3	79	2	441	2	740	2	971	2	1134	3	1183	3	1295	3	1321	3	1383
12			4	54	4	321	3	645	3	962	3	1114	2	1224	1	1349	1	1399	1	1563
13													18	55	19	52	16	63		
14													19	50	18	52			17	62
15					10	50	11	82	13	97	12	122	11	145	12	148	10	169	12	167
16					8	150	8	226	8	354	8	371	8	386	8	438	8	446	8	501
17					7	154	7	306	7	358	6	433	7	437	7	465	6	475	7	513
18							10	91	10	123	10	129	10	153	10	169	12	164	11	177
19			2	83	3	390	4	617	4	794	4	902	4	929	4	964	4	985	4	1016
20							13	67	14	83	14	91	14	104	13	125	13	139	14	134
21			1	91	1	452	1	793	1	1039	1	1170	1	1242	2	1336	2	1390	2	1481
22							12	75	12	98	13	100	13	113	14	115	14	136	13	142
23					9	112	9	170	9	295	9	359	9	349	9	418	9	402	9	412
24					6	182	6	313	6	399	7	387	6	460	6	493	7	473	6	574
25													17	55	17	53	18	51	16	69
26							14	65	11	119	11	128	12	138	11	148	11	164	10	191
27					5	245	5	421	5	569	5	660	5	674	5	758	5	744	5	790
28																	19	50	20	54
29									15	65	15	59	15	67	15	74	15	83	15	96
667													16	56	16	54	20	50	19	56
2471																	17	52		
4175																			18	60

(b) Linear-based system

Gene Index	Fitness Evaluation 5000 Rank	5000 Freq.	10000 Rank	10000 Freq.	15000 Rank	15000 Freq.	20000 Rank	20000 Freq.	25000 Rank	25000 Freq.	30000 Rank	30000 Freq.	35000 Rank	35000 Freq.	40000 Rank	40000 Freq.	45000 Rank	45000 Freq.	50000 Rank	50000 Freq.
11			3	481	3	1105	3	1460	3	1625	3	1706	3	1740	3	1793	3	1830	3	1904
12			2	508	2	1212	1	1687	1	1879	1	1990	1	2168	1	2134	1	2237	1	2247
13					18	69	16	105	16	109	17	95	18	87	18	87	18	88	17	113
14					16	72	17	98	17	89	16	110	16	122	16	115	16	125	16	118
15			13	53	13	136	11	213	13	213	12	208	11	247	12	241	12	233	11	244
16			8	241	7	540	6	803	6	842	6	878	6	909	6	934	6	886	6	895
17			7	283	8	534	7	678	7	755	8	747	8	752	7	839	7	869	7	846
18			10	80	10	174	10	224	10	227	10	277	10	282	10	250	10	279	10	316
19			4	439	4	923	4	1119	4	1243	4	1286	4	1357	4	1432	4	1375	4	1415
20					14	97	14	141	14	164	14	181	14	183	14	180	14	179	14	178
21			1	573	1	1284	2	1683	2	1853	2	1935	2	1999	2	2051	2	2109	2	2121
22			11	61	12	137	12	190	12	221	11	209	13	220	13	195	13	216	13	195
23			9	194	9	429	9	576	9	622	9	659	9	707	9	698	9	729	9	720
24			6	307	6	558	8	649	8	751	7	748	7	756	8	756	8	800	8	822
25					17	70	18	75	18	80	18	83	17	99	17	92	17	122	18	99
26			12	55	11	142	13	188	11	227	13	192	12	243	11	245	11	239	12	228
27			5	392	5	816	5	1085	5	1147	5	1245	5	1315	5	1338	5	1323	5	1356
28							22	50			22	51	21	62	23	59	22	56	22	59
29					15	90	15	130	15	153	15	122	15	146	15	170	15	159	15	150
667							21	51	19	68	19	76	20	65	19	71	19	77	20	63
2471															26	52				
2816													23	59			25	53	23	55
2828									23	50			22	59	20	63	24	54		
4175									21	58					22	60	23	55		
4377							20	52	22	57	20	61	24	59	24	57	20	64	19	64
4390															25	54				
4883							19	57	20	58	21	55	19	69	21	63	21	57	21	61

(c) Tanh-based system

Gene Index	Fitness Evaluation 5000 Rank	5000 Freq.	10000 Rank	10000 Freq.	15000 Rank	15000 Freq.	20000 Rank	20000 Freq.	25000 Rank	25000 Freq.	30000 Rank	30000 Freq.	35000 Rank	35000 Freq.	40000 Rank	40000 Freq.	45000 Rank	45000 Freq.	50000 Rank	50000 Freq.
11					2	269	2	522	3	698	2	843	3	925	3	1004	2	1047	3	1092
12					4	186	4	439	2	704	3	833	2	950	1	1072	1	1152	1	1210
15									11	71	10	97	11	99	13	93	11	110	12	123
16					8	76	8	162	7	237	8	272	8	285	7	328	8	338	7	370
17					7	96	7	163	8	229	7	281	7	300	8	301	7	358	8	367
18									12	70	12	92	12	91	10	126	10	123	11	131
19					3	249	3	446	4	559	4	651	4	672	4	732	4	783	4	852
20									13	55	14	75	10	99	14	73	14	88	13	104
21			1	50	1	286	1	570	1	763	1	872	1	981	2	1064	3	1034	2	1156
22							10	51	14	54	13	77	14	64	12	97	13	104	14	96
23					9	64	9	117	9	184	9	240	9	227	9	265	9	306	9	283
24					6	105	6	177	6	254	6	294	6	330	6	352	6	388	6	420
26									10	74	11	92	13	89	11	113	12	109	10	152
27					5	140	5	292	5	365	5	476	5	505	5	546	5	592	5	603
29											15	55			15	50	15	62	16	54
667																			15	60

Continued on Next Page...

Table B.6 – *Continued*

(d) Threshold-based system																				
	Fitness Evaluation																			
Gene	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
11			2	297	3	788	3	1158	3	1360	3	1411	3	1470	3	1533	3	1518	3	1605
12			3	269	2	819	2	1224	2	1465	1	1601	1	1750	1	1766	1	1815	1	1828
13							18	57	16	77	17	81	16	92	16	84	18	70	18	65
14							17	58	18	54	16	87	17	87	18	78	17	73	16	91
15					11	91	11	148	12	149	11	205	11	198	12	204	12	207	11	210
16			8	144	7	364	6	514	6	649	6	648	6	733	6	699	6	737	6	731
17			7	150	8	338	7	476	7	574	7	626	7	623	7	624	7	687	7	673
18					10	127	13	136	10	198	10	218	10	217	10	225	10	230	10	223
19			4	256	4	656	4	922	4	999	4	1107	4	1073	4	1106	4	1190	4	1162
20					14	75	14	108	14	114	14	138	14	149	14	139	14	165	14	156
21			1	342	1	918	1	1264	1	1516	2	1575	2	1652	2	1695	2	1736	2	1737
22					13	81	12	139	13	143	13	146	13	173	13	178	13	167	12	186
23			9	103	9	260	9	437	9	480	9	513	9	520	9	554	9	585	9	546
24			6	153	6	365	8	460	8	497	8	563	8	604	8	614	8	621	8	606
25							16	67	17	66	18	79	18	72	17	82	16	93	17	77
26					12	83	10	162	11	156	12	191	12	173	11	204	11	220	13	183
27			5	213	5	553	5	759	5	930	5	985	5	1006	5	1013	5	1042	5	1024
28													22	55			22	52	25	50
29					15	67	15	85	15	99	15	106	15	107	15	135	15	133	15	125
30													20	64						
667													21	58			19	66	19	64
2471																	23	50	20	59
2816															19	54				
2828																			21	56
4175															20	51	21	55	23	51
4377									19	54			19	66					24	51
4883																	20	64	22	52

Table B.7: The complete list of ALL/AML genes in the population size 100.

(a) Sigmoid-based system																					
		Fitness Evaluation																			
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at							42	59	47	51					46	60			51	53
668	D86967_at					40	59	45	56	45	52	42	57	42	67			44	62	50	54
758	D88270_at	14	117	11	162	14	156	11	185	11	193	14	175	11	203	11	216	12	193	13	184
760	D88422_at	12	121	14	154	10	199	14	173	13	183	11	213	12	189	12	207	11	211	11	222
804	HG1612-HT1612_at	7	255	7	272	7	302	7	332	7	322	7	320	7	317	7	314	7	331	7	317
1144	J05243_at	33	50	27	70	39	62	33	74	30	78	41	58	40	68	29	86	35	73	37	75
1239	L07633_at	32	52	41	51			44	57					45	61	48	58	42	64	30	85
1630	L47738_at					38	63	34	73	49	50	46	52	43	64	40	66	43	62	40	67
1669	M10612_at													52	52						
1685	M11722_at	6	255	6	334	5	368	5	366	5	394	5	405	5	396	5	420	5	427	5	436
1704	M13792_at							38	62	36	69	43	56	35	73	33	75	31	81	32	81
1745	M16038_at			34	62	33	65	26	87	41	55	34	70	38	70	30	80	38	68	27	90
1779	M19507_at	8	176	9	228	8	249	9	235	8	275	8	272	8	263	9	261	9	274	8	280
1796	M20902_at													50	53			41	64		
1809	M21624_at									48	51	44	55			45	61			47	59
1829	M22960_at	24	68	26	81	25	88	29	80	26	110	26	103	28	91	26	101	24	114	28	90
1834	M23197_at	23	83	29	70	28	78	28	83	32	77	27	97	30	84	36	73	34	75	29	88
1882	M27891_at	1	562	1	691	1	724	1	767	1	751	1	755	1	772	1	776	1	783	1	785
1909	M29696_at			40	51	37	63	46	54	43	53	38	67	44	63	42	62	45	57	46	59
1928	M31303_rna1_at	28	66	33	63	35	64	27	84	29	83	31	82	34	75	27	94	28	89	34	78
1962	M33680_at	18	110	20	121	20	130	17	146	18	143	17	164	23	122	24	119	17	160	19	146
1975	M34344_at													51	52	52	50	55	51		
2020	M55150_at					44	51									44	61				
2111	M62762_at													49	53						
2121	M63138_at	10	147	10	170	11	189	13	175	12	189	12	183	14	163	13	194	10	215	14	182
2288	M84526_at	4	321	2	439	2	488	2	516	2	536	2	560	2	589	2	574	2	566	2	575
2335	M89957_at	29	60	39	53	41	57	32	77	33	75	33	72	32	77	39	68	39	65	26	94
2354	M92287_at	2	362	3	436	3	441	4	487	4	446	4	436	4	463	4	422	4	444	4	454
2363	M93056_at			43	50									47	56	43	61	47	55	43	64
2402	M96326_rna1_at	16	112	15	147	13	168	12	183	14	181	18	162	15	161	14	190	13	189	12	201
2642	U05259_rna1_at	5	267	5	343	6	318	6	365	6	362	6	347	6	362	6	364	6	364	6	356
3252	U46499_at	22	87	23	98	21	129	21	126	19	140	13	175	16	155	16	177	16	165	17	171
3320	M19508_xpt3_s_at																			52	50
4050	M19508_xpt3_s_at	31	55	31	65	32	68	36	66	34	74	36	69	36	73	31	80			41	66
4196	X17042_at							37	65	38	66			33	76	41	64	36	68	49	55
4211	X51521_at	21	95	17	136	15	152	20	128	22	126	23	122	22	129	18	151	20	148	20	144
4229	X52056_at	26	67	24	92	24	103	24	109	24	112	25	108	26	96	23	120	25	111	24	120
4328	X59417_at	13	117	13	154	12	175	10	206	10	211	10	216	10	214	10	236	14	187	10	236
4342	X59871_at											45	53								
4366	X61587_at																	49	53		
4373	X62320_at	27	66	22	102	26	86	22	122	23	117	24	119	20	133	22	121	27	98	25	109
4377	X62654_rna1_at	11	130	18	126	18	142	16	148	21	126	15	166	21	130	20	142	19	149	18	170
4438	X66401_cds1_at					42	54			46	51					49	57	50	53		
4680	X82240_rna1_at	19	95	19	123	16	151	19	138	17	146	16	166	17	153	19	143	23	131	21	131
4847	X95735_at	3	335	4	372	4	437	3	490	3	484	3	488	3	485	3	474	3	494	3	494
4951	Y07604_at							35	68			40	60	46	58						
5191	Z69881_at																	54	51		
5501	Z15115_at	25	67	25	87	23	108	25	108	25	111	21	127	24	107	25	112	21	142	23	122
5552	L06797_s_at													48	55	51	51	46	56	48	56
5772	U22376_cds2_s_at	17	112	16	142	19	139	18	144	16	148	20	148	18	145	17	173	15	177	16	172
6041	L09209_s_at	9	171	8	232	9	237	8	262	9	271	9	264	9	263	8	301	8	281	9	278
6049	U89922_s_at	30	56	28	70	31	69	30	80	28	83	28	86	29	89	28	89	30	84	35	78
6079	U59632_s_at													53	51						
6167	M12959_s_at			38	54					42	54	47	51					48	54		
6185	X64072_s_at			35	59	30	69	47	50	39	63	37	68	39	69	35	73	29	88	39	68
6200	M28130_rna1_s_at			30	68	27	79	43	58	27	89	29	85	27	93	34	73	37	68	33	80
6201	Y00787_s_at			32	64	29	75	31	79	31	77	32	78	25	100	37	72	26	99	31	83
6215	M19508_xpt3_s_at															50	52				
6225	M84371_rna1_s_at			36	59	34	65	39	62	37	69	39	66	41	68	47	58	40	65	44	64
6271	M33493_s_at											48	50							45	64
6308	M57731_s_at																	53	51		
6376	M83652_s_at	20	95	21	114	22	115	23	115	20	131	22	127	19	134	21	122	22	132	22	125
6378	M83667_rna1_s_at			42	50			41	60	35	69	35	69	31	80	38	72	32	79	36	77
6510	U23852_s_at					43	52			44	53							52	52	42	65
6702	X97267_rna1_s_at			37	58	36	63	40	61	40	62	30	83	37	70	32	79	33	77	38	71
6855	M31523_at	15	115	12	160	17	149	15	169	15	155	19	157	13	183	15	180	18	160	15	182
7119	U29175_at																	51	52		

Continued on Next Page...

Table B.7 – *Continued*

(b) Linear-based system																			
		Fitness Evaluation																	
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at	**1**	*488*	**32**	*72*	**26**	*98*	**32**	*87*	**25**	*104*	**32**	*88*	**27**	*100*	**33**	*89*	**41**	*79*	**38**	*85*
668	D86967_at	**2**	*348*			**44**	*59*	**44**	*61*	**39**	*73*	**55**	*52*	**61**	*51*	**47**	*62*	**40**	*80*	**52**	*59*
758	D88270_at	**3**	*325*	**48**	*50*	**46**	*56*	**52**	*51*	**38**	*74*	**50**	*56*	**43**	*67*	**36**	*87*	**45**	*66*	**43**	*71*
760	D88422_at			**26**	*89*	**22**	*101*	**24**	*101*	**26**	*103*	**28**	*106*	**36**	*85*	**29**	*99*	**29**	*102*	**32**	*100*
804	HG1612-HT1612_at	**4**	*227*	**5**	*275*	**5**	*279*	**5**	*318*	**5**	*337*	**6**	*300*	**7**	*303*	**6**	*320*	**8**	*281*	**7**	*329*
1144	J05243_at			**42**	*57*	**42**	*61*	**48**	*55*	**48**	*62*	**42**	*66*	**60**	*51*	**45**	*66*	**48**	*65*	**51**	*64*
1239	L07633_at	**5**	*214*	**22**	*100*	**25**	*99*	**23**	*107*	**23**	*109*	**27**	*108*	**32**	*92*	**24**	*109*	**22**	*123*	**27**	*108*
1291	L11669_at													**57**	*53*						
1400	L21954_at			**41**	*58*	**49**	*53*	**41**	*66*	**41**	*68*	**53**	*53*	**39**	*76*	**48**	*58*	**43**	*68*	**46**	*67*
1630	L47738_at							**54**	*51*			**54**	*53*			**53**	*54*				
1674	M11147_at															**59**	*50*				
1685	M11722_at	**6**	*204*	**16**	*131*	**19**	*117*	**17**	*136*	**16**	*141*	**19**	*136*	**18**	*145*	**19**	*132*	**18**	*137*	**17**	*164*
1745	M16038_at	**7**	*201*	**37**	*65*	**38**	*77*	**42**	*64*	**44**	*65*	**45**	*62*	**38**	*77*	**42**	*72*	**42**	*75*	**39**	*84*
1779	M19507_at	**8**	*180*	**9**	*221*	**8**	*241*	**7**	*262*	**8**	*266*	**8**	*279*	**6**	*326*	**7**	*313*	**6**	*318*	**5**	*340*
1796	M20902_at			**40**	*63*	**43**	*61*	**35**	*85*	**40**	*69*	**35**	*75*	**34**	*87*	**39**	*74*	**36**	*88*	**36**	*86*
1829	M22960_at	**9**	*179*	**15**	*132*	**12**	*159*	**13**	*146*	**12**	*163*	**14**	*163*	**13**	*170*	**13**	*160*	**17**	*167*	**19**	*143*
1834	M23197_at	**10**	*155*	**20**	*102*	**23**	*101*	**25**	*101*	**29**	*100*	**26**	*108*	**28**	*99*	**26**	*106*	**32**	*98*	**33**	*99*
1882	M27891_at	**11**	*143*	**7**	*226*	**7**	*242*	**8**	*239*	**7**	*274*	**7**	*281*	**8**	*288*	**8**	*270*	**7**	*291*	**8**	*315*
1928	M31303_rna1_at	**12**	*125*	**21**	*101*	**33**	*82*	**33**	*86*	**35**	*83*	**22**	*113*	**37**	*83*	**32**	*95*	**31**	*98*	**28**	*108*
1941	M31994_at									**19**	*138*							**58**	*55*		
1962	M33680_at	**13**	*122*	**17**	*122*	**18**	*132*	**21**	*121*			**20**	*135*	**15**	*158*	**15**	*151*	**12**	*182*	**16**	*166*
1975	M34344_at			**43**	*57*	**36**	*79*	**36**	*83*	**47**	*62*	**46**	*62*	**41**	*73*	**41**	*73*	**34**	*95*	**48**	*66*
2020	M55150_at	**14**	*121*	**25**	*91*	**31**	*85*	**31**	*91*	**21**	*111*	**25**	*111*	**35**	*86*	**22**	*125*	**24**	*117*	**26**	*108*
2111	M62762_at	**15**	*102*	**28**	*79*	**37**	*77*	**29**	*92*	**34**	*83*	**36**	*74*	**29**	*98*	**35**	*88*	**27**	*110*	**34**	*95*
2121	M63138_at	**16**	*98*	**3**	*349*	**3**	*379*	**3**	*396*	**2**	*411*	**2**	*402*	**3**	*426*	**3**	*402*	**3**	*418*	**2**	*458*
2288	M84526_at	**17**	*95*	**4**	*287*	**4**	*311*	**4**	*339*	**4**	*362*	**4**	*376*	**4**	*391*	**4**	*399*	**2**	*432*	**3**	*411*
2335	M89957_at							**53**	*51*			**51**	*55*					**57**	*57*		
2354	M92287_at	**18**	*93*	**2**	*364*	**2**	*410*	**2**	*397*	**3**	*395*	**3**	*387*	**2**	*429*	**2**	*412*	**4**	*397*	**4**	*398*
2363	M93056_at			**44**	*55*	**39**	*74*	**38**	*77*	**46**	*63*	**44**	*63*	**52**	*56*	**44**	*66*	**30**	*98*	**41**	*76*
2394	M95678_at							**51**	*52*	**49**	*61*			**44**	*65*	**55**	*52*	**54**	*58*	**49**	*64*
2402	M96326_rna1_at	**19**	*86*	**29**	*76*	**24**	*100*	**26**	*100*	**32**	*90*	**30**	*96*	**22**	*113*	**27**	*103*	**23**	*118*	**23**	*124*
2408	M96803_at									**56**	*52*			**58**	*53*						
2546	S82470_at															**57**	*51*	**53**	*59*	**55**	*55*
2642	U05259_rna1_at	**20**	*86*	**8**	*226*	**10**	*196*	**9**	*208*	**9**	*212*	**9**	*212*	**9**	*223*	**9**	*252*	**9**	*236*	**9**	*227*
3252	U46499_at	**21**	*78*	**18**	*118*	**16**	*134*	**14**	*144*	**13**	*162*	**12**	*175*	**12**	*178*	**12**	*168*	**13**	*179*	**14**	*183*
3258	U46751_at									**54**	*52*	**52**	*53*			**60**	*50*				
3320	U50136_rna1_at	**22**	*76*	**30**	*76*	**21**	*108*	**27**	*98*	**24**	*105*	**18**	*137*	**21**	*125*	**21**	*128*	**21**	*131*	**18**	*143*
3984	U94855_at									**52**	*54*	**40**	*69*	**53**	*56*					**54**	*56*
4050	X03934_at											**41**	*69*	**47**	*62*			**51**	*61*		
4095	X06948_at																			**58**	*50*
4196	X17042_at	**23**	*73*	**19**	*111*	**17**	*132*	**19**	*125*	**17**	*140*	**17**	*137*	**19**	*143*	**14**	*157*	**16**	*169*	**15**	*173*
4211	X51521_at	**24**	*71*	**13**	*141*	**14**	*145*	**16**	*139*	**14**	*149*	**13**	*167*	**14**	*170*	**16**	*148*	**15**	*175*	**13**	*189*
4229	X52056_at	**25**	*66*	**33**	*71*	**29**	*87*	**37**	*78*	**28**	*102*	**29**	*102*	**26**	*101*	**31**	*95*	**37**	*86*	**31**	*103*
4328	X59417_at	**26**	*66*	**11**	*165*	**11**	*177*	**11**	*203*	**10**	*197*	**11**	*188*	**11**	*182*	**11**	*225*	**10**	*222*	**12**	*204*
4366	X61587_at							**49**	*54*											**50**	*64*
4373	X62320_at	**27**	*66*	**24**	*96*	**20**	*112*	**22**	*111*	**22**	*109*	**21**	*135*	**23**	*110*	**25**	*107*	**20**	*135*	**24**	*123*
4377	X62654_rna1_at	**28**	*63*	**10**	*192*	**9**	*203*	**10**	*206*	**11**	*191*	**10**	*200*	**10**	*210*	**10**	*235*	**11**	*214*	**11**	*206*
4438	X66401_cds1_at			**45**	*53*	**45**	*58*	**50**	*54*	**55**	*52*	**49**	*59*	**49**	*57*	**50**	*57*	**47**	*65*	**44**	*70*
4847	X95735_at	**29**	*61*	**1**	*552*	**1**	*589*	**1**	*597*	**1**	*629*	**1**	*633*	**1**	*636*	**1**	*659*	**1**	*628*	**1**	*653*
4951	Y07604_at	**30**	*57*	**14**	*138*	**15**	*137*	**15**	*141*	**20**	*116*	**15**	*161*	**17**	*148*	**18**	*137*	**19**	*137*	**21**	*135*
5062	Z14982_rna1_at																	**61**	*50*		
5280	J02783_at			**34**	*71*					**50**	*58*			**42**	*68*	**46**	*65*	**49**	*65*		
5501	Z15115_at			**38**	*63*	**35**	*79*	**30**	*91*	**27**	*103*	**31**	*90*	**30**	*96*	**28**	*102*	**25**	*115*	**29**	*106*
5772	U22376_cds2_s_at	**31**	*57*	**35**	*71*	**28**	*91*	**34**	*85*	**36**	*79*	**34**	*84*	**31**	*92*	**37**	*83*	**35**	*92*	**25**	*110*
5950	M29610_s_at							**55**	*51*	**53**	*53*			**50**	*57*	**58**	*51*	**52**	*60*	**53**	*59*
5952	U05255_s_at							**6**	*314*	**57**	*51*			**45**	*64*	**49**	*58*	**55**	*58*	**42**	*74*
6041	L09209_s_at	**32**	*56*	**6**	*274*	**6**	*252*			**6**	*324*	**5**	*338*	**5**	*327*	**5**	*348*	**5**	*326*	**6**	*338*
6049	U89922_s_at									**45**	*65*			**55**	*55*	**61**	*50*				
6079	U59632_s_at					**50**	*52*	**39**	*75*			**37**	*74*	**56**	*55*	**38**	*77*	**46**	*66*	**45**	*69*
6185	X64072_s_at	**33**	*54*	**31**	*72*	**41**	*67*	**28**	*93*	**33**	*86*	**38**	*71*	**25**	*102*	**30**	*95*	**39**	*82*	**35**	*90*
6200	M28130_rna1_s_at			**36**	*66*	**32**	*83*	**40**	*69*	**30**	*95*	**33**	*86*	**33**	*88*	**34**	*88*	**33**	*95*	**30**	*104*
6201	Y00787_s_at									**42**	*65*			**54**	*55*			**59**	*53*	**56**	*52*
6215	M19508_xpt3_s_at	**34**	*54*	**27**	*80*	**30**	*86*	**20**	*122*	**31**	*93*	**23**	*112*	**20**	*136*	**23**	*118*	**28**	*110*	**20**	*135*
6225	M84371_rna1_s_at	**35**	*53*	**39**	*63*	**34**	*80*	**47**	*57*	**37**	*75*	**39**	*71*	**46**	*64*	**40**	*73*	**50**	*62*	**40**	*81*
6271	M33493_s_at					**48**	*54*									**54**	*53*				
6376	M83652_s_at	**36**	*52*	**23**	*98*	**27**	*95*	**18**	*128*	**18**	*140*	**24**	*112*	**24**	*109*	**20**	*129*	**26**	*114*	**22**	*131*
6539	X85116_rna1_s_at	**37**	*52*	**46**	*51*	**40**	*69*	**45**	*58*	**43**	*65*	**43**	*65*	**40**	*75*	**43**	*71*	**38**	*85*	**37**	*85*
6702	X97267_rna1_s_at							**43**	*61*			**47**	*61*	**48**	*58*	**51**	*56*	**56**	*57*	**47**	*67*
6796	J02982_f_at					**47**	*56*			**51**	*58*			**59**	*52*	**56**	*52*	**44**	*68*	**57**	*51*
6806	X14008_rna1_f_at			**47**	*50*			**46**	*57*			**48**	*60*	**51**	*56*	**52**	*54*	**60**	*51*		
6855	M31523_at	**38**	*51*	**12**	*165*	**13**	*159*	**12**	*172*	**15**	*148*	**16**	*151*	**16**	*154*	**17**	*148*	**14**	*176*	**10**	*210*

Continued on Next Page...

Table B.7 – *Continued*

(c) Tanh-based system																			
		Fitness Evaluation																	
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at	1	589	39	56	38	61	45	53	46	56	50	52	41	61	49	51	51	50	46	62
668	D86967_at	2	404							41	59	46	55	42	61	50	50			43	64
758	D88270_at	3	351	12	154	12	178	12	173	12	181	11	199	12	189	13	191	13	195	11	206
760	D88422_at	4	326	11	159	11	181	13	171	11	188	10	208	11	201	11	202	12	209	13	188
804	HG1612-HT1612_at	5	259	7	309	7	304	7	292	7	315	7	320	8	285	7	333	7	358	7	291
1144	J05243_at	6	253	30	72	34	65	32	73	36	67	32	77	39	62	28	85	47	59	40	65
1239	L07633_at	7	250	40	56			38	63	42	59			47	57	36	71	45	60	52	51
1630	L47738_at	32	52									44	56	51	52	51	50	40	65	35	75
1669	M10612_at															48	51				
1685	M11722_at	8	185	5	357	5	327	5	392	5	400	5	361	5	385	5	382	4	434	5	405
1704	M13792_at					32	66	36	64	50	51	39	60	33	68	41	63	30	82	37	69
1745	M16038_at	9	173	34	60	35	63	34	65	37	65	30	85	46	57	30	82	33	77	36	70
1779	M19507_at	10	159	9	233	9	257	8	259	9	251	8	299	9	268	8	312	8	308	9	275
1796	M20902_at							26	94	47	53			45	60						
1809	M21624_at											49	52			46	54	49	53		
1829	M22960_at	14	115	27	84	23	111	31	78	30	83	25	100	29	79	25	104	26	100	26	105
1834	M23197_at	13	115	26	86	27	83	1	724	26	92	29	86	30	77	34	76	27	99	30	84
1882	M27891_at	15	112	1	672	1	676	37	63	1	756	1	767	1	786	1	752	1	804	1	769
1909	M29696_at							29	83	40	60	37	63	35	67	44	56	43	61	29	84
1928	M31303_rna1_at	12	130	28	77	28	75	23	125	27	90	34	74	27	92	32	81	29	87	34	76
1962	M33680_at	11	141	22	105	16	162	44	56	21	133	16	164	17	158	17	164	18	151	16	165
1975	M34344_at					43	50			45	56									51	54
2020	M55150_at	20	96											52	51					49	55
2111	M62762_at	16	108																		
2121	M63138_at	17	105	10	194	13	177	11	183	16	150	13	175	16	164	12	193	14	181	14	174
2288	M84526_at	19	97	2	454	2	496	2	559	2	533	2	577	2	567	2	562	2	549	2	605
2335	M89957_at	23	75	36	58	39	59	40	61	32	71	42	58	36	65	43	57	36	72	45	63
2354	M92287_at	18	104	3	433	3	451	4	431	4	450	4	459	4	439	4	465	5	429	4	464
2363	M93056_at									39	60	47	53			45	54				
2402	M96326_rna1_at	27	68	13	150	19	142	15	164	14	167	19	149	13	180	15	166	11	216	12	199
2642	U05259_rna1_at	22	76	6	333	6	311	6	327	6	317	6	361	6	344	6	348	6	385	6	391
3252	U46499_at	25	70	17	131	18	147	21	127	15	152	14	167	15	164	18	161	17	155	19	145
3320	U50136_rna1_at	24	70			40	58					52	51	48	54						
4050	X03934_at	21	91	37	58			39	63	38	61	41	60	44	60	38	67	31	78	38	68
4196	X17042_at	28	67	38	56			43	57	43	56	38	61	32	70	37	69	38	69	33	80
4211	X51521_at	29	56	19	126	20	139	19	140	22	125	17	162	18	143	21	143	20	147	15	166
4229	X52056_at			25	92	26	85	25	106	24	109	27	88	25	110	24	108	23	113	23	119
4291	X56468_at					42	52							53	51	42	58	48	55		
4328	X59417_at	30	53	14	146	10	183	10	190	10	215	12	188	10	223	10	226	10	235	10	222
4366	X61587_at													49	53			41	62	44	63
4373	X62320_at	34	50	23	97	25	92	20	128	25	96	24	100	24	110	26	102	24	107	24	119
4377	X62654_rna1_at			20	122	14	176	17	151	18	143	20	127	20	136	20	144	16	164	20	143
4438	X66401_cds1_at			32	63							40	60	37	64			44	61	48	57
4680	X82240_rna1_at			21	119	21	118	18	140	17	146	22	110	22	126	19	151	22	126	25	118
4847	X95735_at			4	387	4	440	3	466	3	475	3	510	3	480	3	509	3	465	3	524
4951	Y07604_at							46	51	49	52	45	56	50	52			46	59		
5501	Z15115_at			24	94	22	111	22	126	23	112	23	102	21	133	22	140	25	107	21	133
5552	L06797_s_at							41	60	44	56										
5772	U22376_cds2_s_at			15	141	17	160	16	164	20	140	18	154	19	140	14	172	19	148	17	151
6041	L09209_s_at			8	238	8	276	9	254	8	289	9	287	7	287	9	297	9	250	8	283
6049	U89922_s_at			29	77	31	70	33	71	28	88	33	76	28	87	29	84	35	74	41	64
6079	U59632_s_at																			50	55
6167	M12959_s_at																	50	51		
6185	X64072_s_at			43	50	37	61			35	68	35	65	34	68	39	66	39	69	47	60
6200	M28130_rna1_s_at	26	69	31	68	36	61	28	88	29	86	28	86	26	99	33	76	34	75	27	102
6201	Y00787_s_at			41	55	29	73	27	89	31	78	31	77	38	63	27	92	32	77	28	91
6225	M84371_rna1_s_at	31	52	35	60	33	66	42	60	34	70	36	64	43	60	31	82	42	62	39	65
6271	M33493_s_at											43	56			47	52			42	64
6376	M83652_s_at			18	126	24	105	24	114	19	142	21	125	23	122	23	121	21	142	22	122
6378	M83667_rna1_s_at			42	52	41	55	30	78	48	52			40	61	35	73	37	70	31	83
6510	U23852_s_at											51	52								
6539	X85116_rna1_s_at											48	53					52	50		
6702	X97267_rna1_s_at			33	62	30	73	35	64	33	70	26	90	31	76	40	64	28	91	32	83
6855	M31523_at	33	51	16	136	15	176	14	168	13	171	15	165	14	174	16	165	15	165	18	150

Continued on Next Page...

Table B.7 – *Continued*

(d) Threshold-based system																			
		Fitness Evaluation																	
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at	3	256	35	51			32	63	43	54	40	58	50	53	55	53	50	58	35	69
668	D86967_at	2	259	34	52	43	51					47	52	48	55			53	54		
758	D88270_at	1	365			25	69	37	59	41	57	44	55	29	83	43	64	37	71	53	52
760	D88422_at	6	176	28	64	29	67	21	85	26	88	26	79	27	85	30	82	23	103	29	75
804	HG1612-HT1612_at	8	143	8	197	7	198	8	205	8	222	8	205	8	200	8	227	8	241	8	223
1144	J05243_at	16	79							45	52					56	52	55	52		
1239	L07633_at	7	163	27	67	31	64	26	81	40	60	49	51	31	76	33	72	35	73	33	73
1400	L21954_at							47	50	49	50	45	53	53	51	38	66	44	64		
1674	M11147_at							45	51	46	51	35	64	37	61	52	54	40	68	41	63
1685	M11722_at	9	127	17	100	14	111	15	117	14	133	16	120	17	123	14	147	18	127	15	140
1745	M16038_at	5	181			34	62	40	55	33	70	39	58	32	74	49	57	47	60	40	65
1779	M19507_at	4	207	7	197	8	183	6	245	6	256	6	229	7	240	7	261	7	277	6	286
1796	M20902_at	29	50			30	66	23	83	20	95	32	71	25	90	24	95	29	87	28	81
1829	M22960_at	10	123	18	94	16	102	16	110	17	105	15	123	15	134	15	146	15	137	17	123
1834	M23197_at			31	61	33	63	28	78	28	87	29	78	26	88	29	86	28	89	27	85
1882	M27891_at	12	95	2	334	2	327	2	353	2	397	3	342	2	394	2	387	2	409	2	409
1928	M31303_rna1_at					26	69			37	63	33	69	40	61	40	66	41	67	52	54
1941	M31994_at											42	57	46	57	53	54	56	51	45	58
1962	M33680_at	14	87	30	61	18	93	25	81	27	87	21	92	30	81	21	102	26	93	19	109
1975	M34344_at			22	73	27	68	35	62	24	89	25	84	24	92	28	89	24	97	23	92
2020	M55150_at	24	57	24	69	19	91	30	69	30	76	28	78	20	99	23	95	21	115	39	66
2111	M62762_at			29	63	28	67	24	81	25	88	34	68	22	94	27	89	31	82	26	88
2121	M63138_at	20	67	3	299	4	286	5	288	3	338	4	315	3	348	4	326	3	375	4	337
2288	M84526_at			5	239	3	286	3	348	4	319	2	353	4	334	3	377	4	361	3	388
2335	M89957_at	23	58																		
2354	M92287_at	21	66	4	294	5	269	4	338	5	298	5	284	5	303	5	318	5	333	5	313
2363	M93056_at			26	67			36	60	32	71			45	57	34	71	45	61	50	55
2394	M95678_at									47	51	41	57	49	53	50	56	46	60	49	55
2402	M96326_rna1_at	19	67	10	148	21	83	27	79	21	93	22	87	23	92	26	89	19	121	24	91
2546	S82470_at					37	58							42	58	31	78	48	59	43	61
2642	U05259_rna1_at	17	71	16	100	9	168	10	172	9	193	9	182	10	173	10	165	10	173	9	181
3183	U41635_at																	57	50		
3252	U46499_at	15	80	19	93	17	99	14	120	15	129	14	136	12	165	17	127	16	130	13	154
3320	U50136_rna1_at					22	83	22	83	23	90	19	98	21	96	19	115	17	130	20	108
4050	X03934_at							44	52												
4095	X06948_at																	54	52		
4196	X17042_at	25	57	14	107	12	134	12	146	12	146	10	160	9	185	9	182	9	189	10	164
4211	X51521_at			15	106	23	82	20	89	16	109	18	104	16	124	18	125	22	105	18	112
4229	X52056_at			32	56	42	51	34	62	39	62	43	55			48	58	51	56	51	55
4328	X59417_at			12	121	13	124	11	151	11	162	13	141	14	153	13	157	13	160	11	163
4366	X61587_at					38	55									57	51			46	56
4373	X62320_at			23	70	24	70	29	74	29	81	24	84	33	71	25	90	27	91	25	89
4377	X62654_rna1_at	26	54	9	155	10	168	9	185	10	178	12	155	11	172	11	163	11	171	12	159
4409	X64594_at															45	61				
4847	X95735_at	11	117	1	465	1	472	1	490	1	547	1	497	1	517	1	539	1	552	1	580
4951	Y07604_at	22	62	11	128	11	150	13	134	13	141	11	158	13	155	12	159	12	166	14	154
5280	J02783_at					41	53					46	52	47	55	54	53	43	65		
5501	Z15115_at	27	52					42	54	34	67	30	77	44	57	44	62	39	69	34	69
5772	U22376_cds2_s_at	13	93	25	68	35	61	41	55	42	55	38	59	34	69	32	74	30	86	36	68
5950	M29610_s_at							31	64	51	50	36	63	55	50	39	66	49	58	37	67
5952	U05255_s_at									50	50			54	50	42	64	36	71	44	61
6041	L09209_s_at	18	69	6	232	6	232	7	240	7	233	7	222	6	249	6	276	6	279	7	256
6049	U89922_s_at															59	50			47	56
6079	U59632_s_at					36	59	33	63	35	67	23	86	38	61	36	70	25	95	31	74
6185	X64072_s_at					39	54	48	50	38	62	37	62	36	62	37	69	32	81	38	66
6200	M28130_rna1_s_at							46	50	36	64	48	51	41	60	41	65	38	70	30	74
6201	Y00787_s_at							43	52					51	52	35	70	42	65		
6215	M19508_xpt3_s_at			20	79	20	88	18	92	19	96	20	96	18	111	16	130	20	117	21	107
6225	M84371_rna1_s_at	28	50			44	50			48	51			35	63					48	56
6271	M33493_s_at															51	56	52	55		
6376	M83652_s_at			21	74	32	63	19	89	22	92	27	79	28	85	20	105	33	75	22	107
6539	X85116_rna1_s_at			33	52	40	54	39	55	31	73	31	74	43	58	46	59	34	73	32	73
6796	J02982_f_at									44	53			39	61	58	50			54	51
6806	X14008_rna1_f_at							38	56					52	52	47	58			42	61
6855	M31523_at			13	110	15	111	17	105	18	104	17	112	19	110	22	100	14	149	16	125

Table B.8: The complete list of ALL/AML genes in the population size 200.

(a) Sigmoid-based system																					
		Fitness Evaluation																			
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at					**41**	*53*	**37**	*69*	**38**	*66*	**38**	*66*			**44**	*56*			**43**	*61*
668	D86967_at	**34**	*51*	**35**	*63*	**43**	*53*	**47**	*54*	**48**	*53*	**48**	*53*	**35**	*73*			**36**	*64*	**37**	*69*
758	D88270_at	**18**	*108*	**18**	*168*	**14**	*183*	**17**	*166*	**18**	*177*	**18**	*177*	**18**	*176*	**17**	*180*	**16**	*194*	**19**	*180*
760	D88422_at	**12**	*148*	**17**	*171*	**12**	*206*	**13**	*205*	**11**	*253*	**11**	*253*	**16**	*195*	**12**	*227*	**15**	*197*	**12**	*244*
804	HG1612-HT1612_at	**5**	*418*	**6**	*486*	**6**	*495*	**7**	*453*	**6**	*484*	**6**	*484*	**6**	*521*	**6**	*509*	**6**	*502*	**6**	*516*
1144	J05243_at			**38**	*57*	**39**	*54*	**36**	*72*	**35**	*68*	**35**	*68*	**34**	*76*	**37**	*66*	**38**	*62*	**40**	*65*
1239	L07633_at			**34**	*65*	**26**	*102*	**32**	*79*	**32**	*76*	**32**	*76*	**28**	*88*	**33**	*77*	**30**	*81*	**32**	*83*
1630	L47738_at	**22**	*74*	**33**	*66*	**42**	*53*	**43**	*56*	**39**	*65*	**39**	*65*	**43**	*59*	**41**	*59*	**31**	*78*		
1669	M10612_at									**51**	*51*	**51**	*51*								
1685	M11722_at	**6**	*338*	**5**	*505*	**5**	*558*	**5**	*589*	**5**	*589*	**5**	*589*	**5**	*599*	**5**	*604*	**5**	*628*	**5**	*653*
1704	M13792_at			**39**	*50*			**38**	*68*	**34**	*69*	**34**	*69*	**37**	*71*	**31**	*78*	**33**	*76*	**38**	*67*
1745	M16038_at	**32**	*53*	**36**	*60*	**34**	*66*	**34**	*74*	**33**	*74*	**33**	*74*	**39**	*63*	**36**	*70*	**39**	*60*	**33**	*82*
1779	M19507_at	**8**	*234*	**9**	*306*	**8**	*347*	**8**	*362*	**8**	*388*	**8**	*388*	**8**	*402*	**8**	*381*	**9**	*367*	**8**	*407*
1796	M20902_at					**36**	*62*	**41**	*58*	**37**	*66*	**37**	*66*	**38**	*70*	**39**	*64*	**40**	*58*	**42**	*64*
1809	M21624_at					**44**	*53*			**50**	*52*	**50**	*52*	**41**	*59*			**44**	*52*	**39**	*66*
1829	M22960_at	**31**	*57*	**30**	*81*	**25**	*102*	**29**	*90*	**26**	*105*	**26**	*105*	**27**	*91*	**26**	*97*	**25**	*113*	**25**	*107*
1834	M23197_at	**21**	*79*	**29**	*85*	**28**	*93*	**33**	*77*	**27**	*101*	**27**	*101*	**32**	*79*	**35**	*75*	**34**	*72*	**29**	*86*
1882	M27891_at	**1**	*986*	**1**	*1206*	**1**	*1263*	**1**	*1269*	**1**	*1246*	**1**	*1246*	**1**	*1264*	**1**	*1240*	**1**	*1284*	**1**	*1244*
1909	M29696_at							**46**	*54*	**49**	*52*	**49**	*52*							**49**	*52*
1928	M31303_rna1_at	**25**	*65*	**28**	*85*	**27**	*93*	**30**	*83*	**29**	*86*	**29**	*86*	**26**	*93*	**27**	*89*	**28**	*90*	**31**	*84*
1962	M33680_at	**16**	*125*	**14**	*178*	**18**	*173*	**15**	*195*	**16**	*208*	**16**	*208*	**15**	*200*	**16**	*181*	**18**	*178*	**14**	*208*
1975	M34344_at					**40**	*54*	**44**	*55*	**47**	*54*	**47**	*54*			**50**	*50*	**45**	*52*		
2020	M55150_at													**50**	*51*						
2121	M63138_at	**10**	*180*	**11**	*223*	**11**	*211*	**12**	*217*	**12**	*243*	**12**	*243*	**12**	*219*	**11**	*246*	**12**	*216*	**13**	*237*
2288	M84526_at	**4**	*540*	**2**	*785*	**2**	*897*	**2**	*879*	**2**	*972*	**2**	*972*	**2**	*1032*	**2**	*995*	**2**	*975*	**2**	*998*
2335	M89957_at					**38**	*54*	**45**	*54*	**46**	*54*	**46**	*54*	**47**	*55*	**43**	*58*	**35**	*66*	**46**	*56*
2354	M92287_at	**2**	*695*	**3**	*696*	**3**	*731*	**4**	*662*	**4**	*716*	**4**	*716*	**4**	*714*	**4**	*762*	**4**	*701*	**4**	*705*
2363	M93056_at																			**50**	*52*
2402	M96326_rna1_at	**17**	*118*	**16**	*172*	**15**	*182*	**11**	*230*	**13**	*235*	**13**	*235*	**11**	*223*	**13**	*206*	**11**	*250*	**11**	*252*
2408	M96803_at													**45**	*56*						
2642	U05259_rna1_at	**7**	*324*	**7**	*418*	**7**	*452*	**6**	*500*	**7**	*483*	**7**	*483*	**7**	*477*	**7**	*493*	**7**	*490*	**7**	*471*
3252	U46499_at	**23**	*72*	**22**	*108*	**20**	*159*	**18**	*164*	**19**	*160*	**19**	*160*	**19**	*175*	**19**	*176*	**20**	*158*	**17**	*196*
4050	X03934_at	**29**	*60*	**32**	*75*	**32**	*84*	**25**	*111*	**31**	*80*	**31**	*80*	**29**	*87*	**34**	*77*	**37**	*63*	**35**	*73*
4196	X17042_at					**35**	*62*	**48**	*52*					**36**	*71*	**38**	*64*	**48**	*50*	**48**	*54*
4211	X51521_at	**11**	*160*	**12**	*205*	**17**	*175*	**16**	*172*	**17**	*178*	**17**	*178*	**14**	*201*	**18**	*179*	**13**	*216*	**15**	*205*
4229	X52056_at	**28**	*60*	**27**	*89*	**29**	*91*	**24**	*115*	**24**	*116*	**24**	*116*	**24**	*117*	**25**	*99*	**26**	*109*	**27**	*99*
4291	X56468_at					**45**	*52*									**46**	*53*				
4328	X59417_at	**14**	*134*	**10**	*230*	**10**	*244*	**10**	*270*	**10**	*269*	**10**	*269*	**10**	*272*	**10**	*280*	**10**	*311*	**10**	*293*
4373	X62320_at	**20**	*83*	**24**	*95*	**22**	*117*	**26**	*98*	**25**	*108*	**25**	*108*	**25**	*100*	**24**	*111*	**24**	*117*	**24**	*115*
4377	X62654_rna1_at	**19**	*95*	**19**	*142*	**19**	*168*	**20**	*143*	**20**	*147*	**20**	*147*	**21**	*151*	**20**	*148*	**19**	*162*	**21**	*142*
4438	X66401_cds1_at							**40**	*60*	**41**	*63*	**41**	*63*	**42**	*59*	**42**	*59*	**46**	*52*	**53**	*51*
4680	X82240_rna1_at	**30**	*58*	**23**	*104*	**23**	*108*	**22**	*129*	**23**	*117*	**23**	*117*	**22**	*141*	**23**	*117*	**22**	*139*	**22**	*137*
4847	X95735_at	**3**	*569*	**4**	*693*	**4**	*728*	**3**	*740*	**3**	*820*	**3**	*820*	**3**	*798*	**3**	*771*	**3**	*819*	**3**	*806*
4951	Y07604_at					**47**	*50*	**51**	*50*	**45**	*55*	**45**	*55*	**44**	*57*	**45**	*54*	**42**	*55*	**41**	*65*
5062	Z14982_rna1_at															**21**	*145*			**51**	*52*
5501	Z15115_at	**27**	*64*	**20**	*112*	**21**	*118*	**21**	*136*	**21**	*144*	**21**	*144*	**20**	*172*			**21**	*154*	**20**	*157*
5543	D00749_s_at																			**54**	*51*
5552	L06797_s_at																			**52**	*51*
5772	U22376_cds2_s_at	**15**	*126*	**13**	*188*	**13**	*187*	**14**	*197*	**14**	*216*	**14**	*216*	**13**	*205*	**14**	*197*	**17**	*187*	**18**	*190*
6041	L09209_s_at	**9**	*226*	**8**	*307*	**9**	*323*	**9**	*337*	**9**	*308*	**9**	*308*	**9**	*317*	**9**	*369*	**8**	*374*	**9**	*370*
6049	U89922_s_at	**26**	*65*	**26**	*91*	**33**	*73*	**27**	*93*	**28**	*91*	**28**	*91*	**33**	*77*	**29**	*82*	**29**	*90*	**26**	*104*
6079	U59632_s_at							**50**	*50*							**47**	*52*			**55**	*50*
6169	M13690_s_at									**53**	*50*	**53**	*50*					**49**	*50*		
6185	X64072_s_at													**49**	*53*					**47**	*56*
6200	M28130_rna1_s_at	**33**	*52*	**25**	*91*	**30**	*90*	**28**	*91*	**40**	*63*	**40**	*63*	**30**	*86*	**28**	*87*	**32**	*77*	**30**	*85*
6201	Y00787_s_at			**31**	*75*	**31**	*88*	**31**	*81*	**30**	*84*	**30**	*84*	**31**	*84*	**30**	*81*	**27**	*93*	**28**	*89*
6225	M84371_rna1_s_at					**46**	*50*	**42**	*57*	**43**	*56*	**43**	*56*			**40**	*62*	**47**	*52*	**44**	*58*
6271	M33493_s_at							**39**	*66*	**52**	*50*	**52**	*50*	**40**	*59*	**22**	*135*	**43**	*54*		
6376	M83652_s_at	**24**	*65*	**21**	*111*	**24**	*106*	**23**	*123*	**22**	*131*	**22**	*131*	**23**	*137*	**48**	*51*	**23**	*123*	**23**	*134*
6378	M83667_rna1_s_at									**44**	*55*	**44**	*55*	**48**	*54*						
6510	U23852_s_at							**49**	*51*	**42**	*61*	**42**	*61*							**34**	*75*
6539	X85116_rna1_s_at															**49**	*51*			**45**	*57*
6702	X97267_rna1_s_at			**37**	*57*	**37**	*56*	**35**	*72*	**36**	*66*	**36**	*66*	**46**	*55*	**32**	*78*	**41**	*55*	**36**	*69*
6855	M31523_at	**13**	*139*	**15**	*174*	**16**	*178*	**19**	*160*	**15**	*210*	**15**	*210*	**17**	*180*	**15**	*184*	**14**	*198*	**16**	*199*

Continued on Next Page...

Table B.8 – *Continued*

(b) Linear-based system																					
		Fitness Evaluation																			
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at	**32**	*56*	**32**	*80*	**24**	*106*	**28**	*98*	**34**	*90*	**33**	*90*	**27**	*110*	**29**	*99*	**34**	*94*	**31**	*110*
668	D86967_at	**29**	*63*			**46**	*53*	**36**	*78*	**42**	*65*	**54**	*55*	**41**	*70*	**52**	*55*	**46**	*66*	**50**	*62*
758	D88270_at			**39**	*60*	**36**	*77*	**33**	*84*	**38**	*84*	**38**	*81*	**34**	*92*	**39**	*82*	**37**	*86*	**35**	*95*
760	D88422_at	**35**	*52*	**25**	*90*	**34**	*84*	**32**	*84*	**31**	*93*	**27**	*103*	**30**	*97*	**38**	*89*	**31**	*97*	**29**	*111*
804	HG1612-HT1612_at	**4**	*333*	**5**	*375*	**6**	*419*	**6**	*431*	**6**	*459*	**6**	*420*	**7**	*412*	**6**	*436*	**6**	*429*	**6**	*459*
1144	J05243_at					**49**	*52*	**40**	*66*	**49**	*60*	**60**	*50*	**50**	*61*	**45**	*62*	**43**	*70*	**59**	*54*
1239	L07633_at	**18**	*93*	**20**	*109*	**21**	*131*	**20**	*141*	**20**	*137*	**23**	*127*	**20**	*146*	**19**	*170*	**24**	*127*	**20**	*140*
1381	L20298_at													**61**	*50*	**49**	*57*			**63**	*50*
1400	L21954_at									**60**	*53*	**59**	*50*								
1630	L47738_at													**59**	*51*						
1674	M11147_at															**54**	*53*			**61**	*52*
1685	M11722_at	**19**	*82*	**15**	*168*	**15**	*177*	**12**	*208*	**17**	*168*	**15**	*191*	**13**	*201*	**16**	*184*	**13**	*204*	**17**	*177*
1745	M16038_at			**43**	*53*	**42**	*60*			**53**	*57*	**41**	*66*	**55**	*55*	**44**	*63*	**48**	*65*	**58**	*54*
1779	M19507_at	**9**	*231*	**7**	*351*	**5**	*433*	**5**	*437*	**5**	*461*	**5**	*506*	**5**	*504*	**5**	*514*	**5**	*528*	**5**	*508*
1796	M20902_at			**33**	*77*	**31**	*90*	**29**	*96*	**25**	*111*	**32**	*92*	**29**	*99*	**27**	*105*	**29**	*103*	**23**	*128*
1809	M21624_at							**55**	*50*												
1829	M22960_at	**15**	*111*	**19**	*129*	**19**	*150*	**19**	*147*	**19**	*151*	**16**	*189*	**19**	*146*	**20**	*159*	**18**	*179*	**19**	*163*
1834	M23197_at	**23**	*75*	**30**	*84*	**28**	*97*	**44**	*61*	**35**	*89*	**39**	*75*	**47**	*65*	**33**	*97*	**36**	*89*	**42**	*78*
1882	M27891_at	**7**	*252*	**6**	*358*	**8**	*348*	**7**	*394*	**8**	*371*	**7**	*414*	**6**	*425*	**7**	*424*	**8**	*401*	**8**	*379*
1928	M31303_rna1_at	**20**	*81*	**29**	*84*	**33**	*85*	**31**	*87*	**28**	*103*	**24**	*122*	**33**	*93*	**25**	*113*	**27**	*109*	**33**	*108*
1941	M31994_at									**45**	*62*	**47**	*61*			**48**	*59*	**53**	*59*	**57**	*55*
1962	M33680_at	**12**	*143*	**11**	*195*	**14**	*186*	**11**	*213*	**13**	*184*	**11**	*234*	**11**	*223*	**12**	*218*	**11**	*218*	**14**	*205*
1975	M34344_at			**38**	*61*	**40**	*66*	**38**	*69*	**33**	*92*	**36**	*82*	**36**	*88*	**26**	*109*	**40**	*78*	**34**	*104*
2020	M55150_at	**25**	*72*	**23**	*100*	**32**	*88*	**26**	*102*	**26**	*105*	**29**	*101*	**32**	*94*	**28**	*101*	**30**	*98*	**36**	*87*
2111	M62762_at	**33**	*54*	**36**	*71*	**29**	*93*	**27**	*100*	**30**	*93*	**31**	*94*	**26**	*113*	**31**	*98*	**21**	*130*	**32**	*108*
2121	M63138_at	**3**	*375*	**3**	*538*	**4**	*543*	**4**	*530*	**4**	*522*	**4**	*552*	**4**	*558*	**3**	*612*	**4**	*585*	**4**	*579*
2288	M84526_at	**5**	*324*	**4**	*504*	**2**	*564*	**2**	*654*	**2**	*645*	**2**	*663*	**2**	*689*	**2**	*698*	**2**	*674*	**2**	*702*
2335	M89957_at					**51**	*50*			**57**	*55*	**57**	*51*	**49**	*63*			**59**	*53*	**53**	*60*
2354	M92287_at	**2**	*490*	**2**	*597*	**3**	*550*	**3**	*596*	**3**	*568*	**3**	*588*	**3**	*628*	**4**	*606*	**3**	*589*	**3**	*611*
2363	M93056_at									**52**	*58*	**58**	*51*					**60**	*53*	**52**	*60*
2394	M95678_at							**49**	*54*	**59**	*54*	**52**	*55*	**46**	*67*	**51**	*56*	**44**	*66*	**41**	*78*
2402	M96326_rna1_at			**34**	*73*	**37**	*76*	**35**	*79*	**36**	*88*	**35**	*88*	**25**	*113*	**30**	*98*	**28**	*107*	**26**	*113*
2408	M96803_at					**41**	*62*	**48**	*55*					**58**	*52*	**61**	*50*	**55**	*57*	**54**	*57*
2546	S82470_at																	**62**	*50*		
2642	U05259_rna1_at	**8**	*233*	**9**	*268*	**9**	*284*	**9**	*324*	**9**	*341*	**9**	*329*	**9**	*294*	**9**	*333*	**10**	*328*	**9**	*315*
3252	U46499_at	**17**	*101*	**17**	*158*	**16**	*175*	**14**	*192*	**15**	*175*	**13**	*202*	**17**	*175*	**13**	*196*	**12**	*209*	**12**	*215*
3258	U46751_at							**51**	*53*	**55**	*56*	**48**	*59*	**43**	*69*	**50**	*56*	**52**	*59*	**44**	*66*
3320	U50136_rna1_at	**26**	*71*	**24**	*91*	**20**	*141*	**25**	*106*	**21**	*128*	**21**	*136*	**21**	*136*	**22**	*134*	**23**	*128*	**21**	*134*
3984	U94855_at					**47**	*52*	**50**	*54*	**61**	*51*	**50**	*59*	**40**	*72*			**49**	*63*		
4050	X03934_at							**47**	*55*	**41**	*66*	**44**	*64*	**53**	*56*			**56**	*57*	**47**	*64*
4196	X17042_at	**27**	*68*	**18**	*137*	**17**	*171*	**17**	*162*	**16**	*173*	**17**	*188*	**15**	*197*	**15**	*189*	**16**	*195*	**16**	*198*
4211	X51521_at	**10**	*172*	**12**	*187*	**11**	*233*	**13**	*195*	**11**	*219*	**12**	*216*	**14**	*198*	**11**	*224*	**15**	*200*	**11**	*248*
4229	X52056_at			**41**	*56*	**38**	*74*	**34**	*82*	**46**	*61*	**34**	*89*	**31**	*96*	**41**	*74*	**42**	*70*	**38**	*84*
4291	X56468_at													**56**	*53*			**63**	*50*		
4328	X59417_at	**13**	*135*	**10**	*229*	**10**	*270*	**10**	*286*	**10**	*311*	**10**	*271*	**10**	*289*	**10**	*318*	**9**	*329*	**10**	*300*
4373	X62320_at	**22**	*77*	**28**	*84*	**23**	*112*	**30**	*94*	**32**	*92*	**30**	*97*	**38**	*83*	**34**	*96*	**32**	*96*	**28**	*113*
4377	X62654_rna1_at	**11**	*150*	**14**	*186*	**12**	*209*	**15**	*191*	**14**	*177*	**18**	*186*	**12**	*202*	**17**	*180*	**17**	*182*	**13**	*214*
4409	X64594_at																	**61**	*51*		
4438	X66401_cds1_at			**40**	*57*			**54**	*51*	**51**	*59*	**55**	*53*	**54**	*56*	**42**	*66*	**45**	*66*	**46**	*65*
4847	X95735_at	**1**	*867*	**1**	*1002*	**1**	*1068*	**1**	*1049*	**1**	*1133*	**1**	*1058*	**1**	*1107*	**1**	*1064*	**1**	*1174*	**1**	*1113*
4951	Y07604_at	**14**	*133*	**16**	*162*	**18**	*155*	**18**	*148*	**18**	*157*	**14**	*202*	**18**	*172*	**14**	*190*	**19**	*169*	**18**	*176*
5062	Z14982_rna1_at															**58**	*51*				
5122	Z32765_at											**62**	*50*								
5191	Z69881_at							**52**	*52*												
5445	X04526_at									**56**	*56*							**57**	*57*	**62**	*52*
5501	Z15115_at	**28**	*63*	**22**	*103*	**26**	*101*	**24**	*112*	**22**	*123*	**26**	*113*	**24**	*118*	**23**	*116*	**26**	*117*	**30**	*110*
5593	D26156_s_at																	**57**	*52*	**55**	*57*
5772	U22376_cds2_s_at	**21**	*81*	**27**	*87*	**22**	*116*	**21**	*122*	**27**	*105*	**25**	*114*	**28**	*106*	**32**	*97*	**25**	*119*	**24**	*128*
5950	M29610_s_at			**37**	*63*	**44**	*56*			**54**	*57*	**42**	*66*	**45**	*68*	**46**	*61*	**47**	*65*	**56**	*56*
5952	U05255_s_at			**42**	*54*	**35**	*80*	**42**	*64*	**47**	*60*	**43**	*65*	**39**	*76*	**35**	*96*	**39**	*82*	**40**	*83*
6041	L09209_s_at	**6**	*268*	**8**	*311*	**7**	*404*	**8**	*388*	**7**	*386*	**8**	*392*	**8**	*388*	**8**	*402*	**7**	*424*	**7**	*428*
6049	U89922_s_at			**44**	*50*					**48**	*60*	**49**	*59*	**48**	*64*	**60**	*50*	**50**	*63*		
6079	U59632_s_at					**43**	*56*	**46**	*55*	**39**	*71*	**56**	*53*	**44**	*69*	**43**	*64*	**38**	*85*	**45**	*66*
6184	M26708_s_at							**53**	*52*	**58**	*55*	**51**	*59*			**55**	*53*			**60**	*53*
6185	X64072_s_at							**45**	*55*	**43**	*63*	**53**	*55*	**51**	*60*	**47**	*60*	**54**	*57*	**43**	*69*
6200	M28130_rna1_s_at	**31**	*57*	**31**	*81*	**30**	*91*	**22**	*118*	**23**	*118*	**20**	*137*	**22**	*128*	**24**	*115*	**20**	*134*	**22**	*130*
6201	Y00787_s_at							**39**	*66*	**40**	*66*	**45**	*62*	**42**	*69*	**40**	*74*	**41**	*72*	**39**	*83*
6215	M19508_xpt3_s_at	**24**	*74*	**26**	*88*	**27**	*99*	**23**	*113*	**24**	*117*	**22**	*129*	**23**	*120*	**21**	*155*	**22**	*128*	**27**	*113*
6225	M84371_rna1_s_at			**45**	*50*	**48**	*52*	**43**	*61*	**44**	*63*	**40**	*68*	**52**	*59*	**53**	*54*			**48**	*64*
6271	M33493_s_at															**56**	*52*				
6376	M83652_s_at	**30**	*60*	**21**	*106*	**25**	*105*	**37**	*69*	**29**	*98*	**28**	*102*	**35**	*88*	**36**	*94*	**33**	*96*	**25**	*116*
6539	X85116_rna1_s_at	**34**	*54*	**35**	*72*	**39**	*67*	**41**	*65*	**37**	*87*	**37**	*81*	**37**	*84*	**37**	*92*	**35**	*91*	**37**	*85*
6702	X97267_rna1_s_at					**45**	*54*			**50**	*59*	**46**	*62*	**57**	*52*			**58**	*54*	**49**	*63*
6796	J02982_f_at											**61**	*50*	**60**	*50*	**59**	*51*	**51**	*63*	**51**	*62*
6855	M31523_at	**16**	*111*	**13**	*186*	**13**	*202*	**16**	*191*	**12**	*192*	**19**	*176*	**16**	*186*	**18**	*172*	**14**	*201*	**15**	*199*
7119	U29175_at					**50**	*51*									**62**	*50*				

Continued on Next Page...

Table B.8 – *Continued*

(c) Tanh-based system																					
		Fitness Evaluation																			
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at			43	50	46	55	45	53							45	58	48	57	52	51
668	D86967_at			37	56	37	66	44	54	38	65	38	65	36	71	35	76	35	76	41	60
758	D88270_at	17	117	16	175	15	189	18	174	17	188	17	188	17	188	18	187	16	176	18	176
760	D88422_at	14	125	17	172	12	230	11	228	11	249	11	249	13	200	12	225	12	237	11	258
804	HG1612-HT1612_at	5	409	6	458	6	515	6	478	6	543	6	543	6	513	6	532	7	495	6	521
1144	J05243_at			34	63	32	72	36	70	32	81	32	81	38	61	43	64	37	72	36	68
1207	L05148_at			30	78			51	50	44	55	44	55			52	50			34	72
1239	L07633_at	30	59			30	76	27	88	25	95	25	95	24	103	31	84	34	82		
1630	L47738_at			42	50	45	56	50	50	39	63	39	63	39	60	38	68	42	65	43	59
1685	M11722_at	6	374	5	532	5	558	5	583	5	582	5	582	5	627	5	569	5	570	5	613
1704	M13792_at			39	55	33	71	35	74	34	74	34	74	32	83	40	66	36	75	35	68
1745	M16038_at			36	62	36	68	37	69	37	68	37	68	35	74	34	77	32	83	47	55
1779	M19507_at	8	226	9	314	8	364	8	377	9	352	9	352	8	398	8	418	8	409	8	390
1796	M20902_at			35	63	43	60	39	56	40	63	40	63	40	59	37	70	39	67	39	65
1809	M21624_at					44	58			49	50	49	50	47	51			47	58		
1829	M22960_at	29	64	25	90	28	90	31	83	28	90	28	90	30	91	23	118	27	95	25	105
1834	M23197_at	27	67	32	65	34	71	30	87	33	76	33	76	34	76	28	87	31	87	31	82
1882	M27891_at	1	1015	1	1177	1	1163	1	1264	1	1292	1	1292	1	1301	1	1307	1	1232	1	1275
1909	M29696_at															53	50			51	51
1928	M31303_rna1_at	26	67	29	78	24	107	28	88	31	84	31	84	26	99	29	87	26	103	29	95
1941	M31994_at							47	51					42	55						
1962	M33680_at	15	123	15	177	18	169	19	171	19	168	19	168	18	179	19	183	14	216	15	203
1975	M34344_at							40	56							54	50			44	58
2121	M63138_at	10	175	10	228	11	235	13	217	13	219	13	219	12	208	14	213	11	251	13	225
2288	M84526_at	4	536	2	881	2	890	2	912	2	944	2	944	2	974	2	975	2	977	2	990
2335	M89957_at					38	64	42	55	41	62	41	62			41	65	41	65	46	56
2363	M93056_at													44	54						
2354	M92287_at	2	646	4	662	4	713	4	714	4	683	4	683	4	726	4	746	4	710	4	722
2402	M96326_rna1_at	18	113	14	186	13	220	12	221	12	243	12	243	11	218	11	230	13	233	12	238
2408	M96803_at															48	56				
2642	U05259_rna1_at	7	318	7	431	7	481	7	441	7	500	7	500	7	483	7	460	6	501	7	487
3252	U46499_at	32	55	21	125	19	167	17	178	18	179	18	179	19	156	17	194	18	163	17	182
4050	X03934_at	23	70	27	84	29	77	33	78	29	85	29	85	29	92	33	78	28	95	30	85
4196	X17042_at					47	52			35	73	35	73	37	65	36	72	38	69	33	78
4211	X51521_at	13	131	12	193	16	180	14	194	15	193	15	193	14	198	13	218	19	158	16	196
4229	X52056_at	24	67	28	79	27	94	24	119	30	84	30	84	25	100	25	106	29	94	26	100
4291	X56468_at							46	53												
4328	X59417_at	11	138	11	214	10	249	10	287	10	270	10	270	10	310	10	306	10	325	10	285
4373	X62320_at	25	67	24	93	23	108	25	112	26	94	26	94	27	98	26	103	24	122	23	132
4377	X62654_rna1_at	16	120	19	128	20	141	20	146	20	150	20	150	21	150	21	131	21	148	20	169
4438	X66401_cds1_at							49	50	43	55	43	55			46	58			40	61
4680	X82240_rna1_at	28	66	22	99	26	105	23	119	22	133	22	133	22	130	24	109	23	133	22	158
4847	X95735_at	3	583	3	697	3	757	3	753	3	765	3	765	3	822	3	770	3	790	3	807
4951	Y07604_at					48	50	41	56							49	55	45	61	49	53
5191	Z69881_at																	50	50	48	53
5501	Z15115_at	22	77	20	126	22	118	21	143	21	137	21	137	20	150	20	161	20	156	21	167
5552	L06797_s_at															47	56			42	59
5772	U22376_cds2_s_at	19	110	13	191	14	195	15	192	14	206	14	206	15	189	15	203	15	196	14	208
6041	L09209_s_at	9	209	8	343	9	323	9	316	8	358	8	358	9	374	9	340	9	352	9	340
6049	U89922_s_at	21	81	26	84	25	107	26	93	24	100	24	100	31	88	30	84	25	106	27	96
6079	U59632_s_at													46	52						
6169	M13690_s_at									47	52	47	52								
6185	X64072_s_at																	44	61	53	50
6200	M28130_rna1_s_at	31	58	38	55	35	69	29	87	27	93	27	93	33	80	27	100	30	93	28	95
6201	Y00787_s_at			31	71	31	72	32	78	36	72	36	72	28	96	32	81	40	66	32	80
6225	M84371_rna1_s_at					39	63	38	65	46	54	46	54	45	54	51	52	46	58	54	50
6271	M33493_s_at					41	61			45	55	45	55			42	65	43	62	38	65
6376	M83652_s_at	20	91	23	99	21	123	22	127	23	125	23	125	23	106	22	120	22	134	24	132
6378	M83667_rna1_s_at			40	53	42	60	43	54	48	51	48	51	41	56	50	54	49	55	50	53
6510	U23852_s_at							48	50							44	62			45	57
6539	X85116_rna1_s_at			41	51																
6702	X97267_rna1_s_at			33	64	40	61	34	74	42	58	42	58	43	54	39	67	33	82	37	65
6855	M31523_at	12	133	18	148	17	174	16	181	16	189	16	189	16	189	16	202	17	172	19	175

Continued on Next Page...

Table B.8 – *Continued*

(d) Threshold-based system																					
		Fitness Evaluation																			
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at			26	72	37	57	35	63	42	61	42	61	41	64	31	80	36	75	38	71
668	D86967_at					39	55			48	52	48	52	42	63	45	60	51	51	43	64
758	D88270_at							36	62	40	65	40	65			42	62	44	63	47	57
760	D88422_at			32	56	36	57	30	73	32	77	32	77			38	64	39	69	35	75
804	HG1612-HT1612_at	5	214	8	280	7	322	8	284	7	290	7	290	34	74	7	298	8	289	8	282
1144	J05243_at													7	313					50	54
1239	L07633_at	18	68	29	61	24	86	24	88	25	104	25	104	23	110	30	85	32	86	31	83
1674	M11147_at							44	53	46	54	46	54	51	50	39	63	41	65	40	70
1685	M11722_at	20	61	16	118	14	145	15	155	15	160	15	160	14	172	14	160	14	161	14	167
1745	M16038_at													44	60	46	57	43	64	53	50
1779	M19507_at	8	171	6	305	6	350	6	368	6	446	6	446	6	444	5	473	5	534	5	487
1796	M20902_at			22	77	21	100	21	110	24	105	24	105	22	113	20	121	25	111	24	111
1829	M22960_at	15	78	20	106	17	116	20	117	17	139	17	139	16	140	21	117	20	133	16	146
1834	M23197_at			28	63	32	64	29	79	28	89	28	89	38	69	43	62	37	74	34	77
1882	M27891_at	3	317	3	415	2	519	3	496	3	537	3	537	3	585	3	572	3	541	3	532
1928	M31303_rna1_at			35	52	34	58	39	60	41	64	41	64	39	65	34	73	42	65	42	64
1941	M31994_at					38	56	46	50	49	51	49	51	40	64	48	56	48	56	46	59
1962	M33680_at	17	75	18	109	16	126	17	126	14	162	14	162	21	114	18	132	16	151	21	127
1975	M34344_at			30	60	27	76	28	79	23	106	23	106	24	106	26	109	21	124	26	107
2020	M55150_at			24	73	25	84	31	70	26	97	26	97	29	97	28	92	26	106	29	91
2111	M62762_at			23	74	28	73	23	97	22	112	22	112	28	97	24	110	24	113	25	108
2121	M63138_at	4	255	5	381	4	447	5	436	5	471	5	471	4	469	4	486	4	538	4	499
2288	M84526_at	6	207	2	424	3	482	2	563	2	572	2	572	2	614	2	639	2	612	2	651
2354	M92287_at	2	344	4	404	5	439	4	475	4	476	4	476	5	461	6	445	6	441	6	484
2363	M93056_at															44	60				
2394	M95678_at															49	55	45	61		
2402	M96326_rna1_at			27	68	26	83	26	81	27	92	27	92	25	105	23	111	27	104	20	127
2546	S82470_at							40	59	43	59	43	59	45	58					52	50
2642	U05259_rna1_at	10	109	9	180	12	175	11	197	12	192	12	192	11	208	13	181	11	209	11	207
3252	U46499_at	21	54	13	142	13	155	14	159	13	188	13	188	13	173	12	182	12	192	13	190
3258	U46751_at					42	50							48	52			47	56	48	54
3320	U50136_rna1_at	22	53	21	83	19	114	19	123	19	116	19	116	17	127	22	116	15	155	15	146
4050	X03934_at													52	50	53	50	49	52		
4095	X06948_at									47	52	47	52	49	52	51	55				
4196	X17042_at	13	84	10	166	9	200	9	226	9	246	9	246	9	236	9	253	9	256	9	275
4211	X51521_at	16	78	15	127	15	139	16	131	18	124	18	124	18	127	16	138	17	145	18	131
4328	X59417_at	11	99	12	163	11	180	12	194	10	211	10	211	10	216	10	214	10	225	10	236
4373	X62320_at			37	50	31	64	32	69	33	75	33	75	36	71	40	63	35	75	33	81
4377	X62654_rna1_at	12	98	14	131	18	116	13	161	21	114	21	114	15	145	17	133	19	143	22	122
4409	X64594_at									44	57	44	57			47	57	50	52	51	53
4847	X95735_at	1	591	1	786	1	849	1	888	1	891	1	891	1	941	1	929	1	959	1	927
4951	Y07604_at	9	117	11	166	10	189	10	200	11	195	11	195	12	204	11	208	13	177	12	195
5501	Z15115_at			33	55	23	91	25	87	36	70	36	70	33	79	33	77	34	78	27	99
5772	U22376_cds2_s_at			25	73	30	67	27	81	35	72	35	72	31	90	32	78	31	87	37	71
5950	M29610_s_at			34	53	29	68	43	58			30	84	37	71	37	66	33	82	36	73
5952	U05255_s_at					41	51	33	67	30	84	8	289	26	101	25	109	23	116	23	115
6041	L09209_s_at	7	179	7	280	8	269	7	291	8	289			8	307	8	268	7	298	7	323
6049	U89922_s_at													50	51			28	96		
6079	U59632_s_at			31	60			34	65	34	72	34	72	35	74	36	67			28	98
6185	X64072_s_at													47	54						
6200	M28130_rna1_s_at					33	60	37	60	29	87	29	87	30	93	27	97	29	95	32	81
6201	Y00787_s_at					35	57	42	58	38	68	38	68	43	60	29	88	46	59	39	70
6215	M19508_xpt3_s_at	19	64	19	106	22	95	22	99	20	114	20	114	20	118	15	140	18	143	19	129
6225	M84371_rna1_s_at																			44	63
6271	M33493_s_at									45	55	45	55			50	55			49	54
6376	M83652_s_at			36	51	40	53	38	60	37	69	37	69	27	99	35	70	38	72	30	90
6539	X85116_rna1_s_at							41	59	31	82	31	82	32	80	41	62	30	94	41	70
6796	J02982_f_at							45	52	39	67	39	67	46	57	52	55	40	69	45	62
6806	X14008_rna1_f_at									50	50	50	50								
6855	M31523_at	14	83	17	118	20	107	18	126	16	147	16	147	19	126	19	122	22	118	17	142

Table B.9: The complete list of ALL/AML genes in the population size 300.

(a) Sigmoid-based system																					
		Fitness Evaluation																			
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at			35	58			41	56	35	61	35	65	42	54	35	67	43	57		
668	D86967_at			36	58	37	55	39	57	34	64	31	72	38	62	38	64	35	71	42	59
758	D88270_at	16	94	18	133	18	162	20	130	21	128	21	137	18	173	20	153	20	144	18	154
760	D88422_at	18	87	16	167	15	175	15	198	16	191	16	178	13	225	13	208	13	210	12	229
804	HG1612-HT1612_at	5	466	5	612	5	647	6	632	6	656	6	673	6	632	6	653	6	651	5	719
1144	J05243_at					38	52	35	67	40	57			39	61	43	52	38	68	47	52
1239	L07633_at	21	64	23	93	23	100	28	91	22	119	23	105	27	94	25	104	23	112	27	91
1630	L47738_at			38	55			37	61	43	52	42	57	35	65	44	51			37	62
1685	M11722_at	6	332	6	572	6	635	5	664	5	690	5	679	5	678	5	674	5	658	6	716
1704	M13792_at			39	51	36	60	48	50	36	60	37	59	34	67	37	64			43	58
1745	M16038_at			32	65			36	61			39	59	41	58	36	66	39	67	38	62
1779	M19507_at	8	218	8	366	8	379	8	391	8	427	8	390	8	467	8	416	8	431	8	433
1796	M20902_at			31	69	33	65	34	70	41	56	32	71	36	64	39	60	33	75	33	73
1809	M21624_at							49	50			40	58					44	50	35	65
1829	M22960_at			24	89	29	83	26	97	29	85	24	98	24	105	23	111	24	100	24	104
1834	M23197_at			25	88	34	63	32	76	32	77	36	64	31	81	28	90	29	86	32	77
1882	M27891_at	1	1184	1	1573	1	1633	1	1621	1	1663	1	1721	1	1592	1	1666	1	1689	1	1667
1928	M31303_rna1_at			34	61	30	80	29	90	25	95	25	93	25	103	31	86	26	98	28	82
1941	M31994_at															40	57				
1962	M33680_at	14	96	13	194	17	172	13	216	13	210	13	217	16	189	14	206	16	198	13	227
1975	M34344_at									37	59	43	57	37	63			40	64	44	58
2111	M62762_at			11	230																
2121	M63138_at	11	146	2	1025	11	249	11	245	11	248	11	249	11	250	11	251	11	271	11	242
2288	M84526_at	4	545	4	887	2	1161	2	1167	2	1180	2	1268	2	1273	2	1221	2	1278	2	1295
2335	M89957_at									44	51										
2354	M92287_at	2	755	15	167	4	902	4	916	4	902	4	903	4	933	4	930	4	952	4	885
2402	M96326_rna1_at	13	101	7	474	16	174	14	206	15	205	15	183	14	211	17	194	14	209	15	222
2408	M96803_at					39	52														
2642	U05259_rna1_at	7	254	21	103	7	516	7	511	7	501	7	518	7	534	7	547	7	515	7	505
3252	U46499_at					20	133	18	149	18	155	19	146	20	143	19	155	19	145	19	141
4050	X03934_at			30	71	26	91	33	73	33	71	34	66	32	77	27	91	30	83	29	81
4196	X17042_at							44	53											41	59
4211	X51521_at	10	148	12	202	12	203	12	218	12	243	12	220	12	232	12	214	12	213	14	225
4229	X52056_at			33	63	32	70	31	79	30	80	30	72	33	68	30	86	31	79	34	72
4328	X59417_at	12	112	10	267	10	279	10	293	10	295	10	306	10	307	10	328	10	335	10	315
4373	X62320_at			26	86	27	88	27	93	28	91	26	91	26	97	22	117	22	114	30	77
4377	X62654_rna1_at	19	73	20	108	21	119	19	140	19	136	18	148	21	135	21	149	21	140	21	128
4438	X66401_cds1_at									39	58										
4680	X82240_rna1_at			27	79	25	91	24	100	27	92	29	80	29	88	26	99	25	100	23	105
4847	X95735_at	3	646	3	946	3	928	3	959	3	967	3	1035	3	964	3	1001	3	966	3	973
4951	Y07604_at							42	55					44	51	42	53	36	70	46	56
5445	X04526_at							45	53			45	53								
5501	Z15115_at			19	108	19	141	22	113	20	135	20	142	19	170	18	160	18	157	20	140
5543	D00749_s_at							46	53					46	51					39	62
5772	U22376_cds2_s_at	17	91	14	181	13	193	16	187	17	191	14	199	15	203	15	202	15	204	17	192
5950	M29610_s_at							47	51												
6041	L09209_s_at	9	168	9	297	9	330	9	351	9	366	9	328	9	364	9	368	9	346	9	353
6049	U89922_s_at	20	73			22	105	23	108	24	99	27	88	23	115	24	108	28	87	26	92
6079	U59632_s_at															45	51				
6200	M28130_rna1_s_at			22	94	28	83	25	97	26	94	28	84	28	92	33	83	34	73	22	113
6201	Y00787_s_at			29	72	31	78	30	84	31	77	44	53	30	85	29	87	32	75	31	77
6215	M19508_xpt3_s_at																			48	50
6271	M33493_s_at			41	50	41	50	40	56	42	53	38	59	45	51	34	70	37	69	36	62
6376	M83652_s_at			28	76	40	50	21	116	23	109	22	115	22	121	32	85	27	92	25	102
6510	U23852_s_at			37	56	24	92	43	54			41	57	43	54			42	59	45	57
6702	X97267_rna1_s_at			40	50	35	61	38	59	38	58	33	68	40	59	41	55	41	60	40	61
6855	M31523_at	15	95	17	158	14	180	17	152	14	206	17	176	17	176	16	200	17	179	16	203

Continued on Next Page...

Table B.9 – *Continued*

(b) Linear-based system																			
		Fitness Evaluation																	
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at	22	60	23	96	25	98	27	95	25	103	27	102	29	101	26	115	23	134	31	92
668	D86967_at			36	57	38	61	38	70	39	64	41	71	40	74	40	73	55	50	34	86
758	D88270_at			28	65	44	55	37	70	33	73	40	72	34	80	33	83	31	92	38	78
760	D88422_at			37	56	34	65	32	75	37	67	37	74	41	72	39	74	41	65	30	96
804	HG1612-HT1612_at	3	399	5	491	5	526	6	538	6	497	6	525	6	558	6	549	6	600	6	564
1144	J05243_at							49	57			54	51	57	50					48	63
1239	L07633_at	16	94	19	123	18	144	19	160	18	151	18	172	19	167	19	161	18	164	18	175
1674	M11147_at															43	67	48	58		
1685	M11722_at	15	97	13	187	16	174	13	209	17	177	17	176	15	198	13	219	13	211	13	234
1745	M16038_at			42	50			45	59			42	63			42	68				
1779	M19507_at	7	232	6	415	6	525	5	610	5	603	5	612	5	590	4	719	5	668	4	723
1796	M20902_at			33	60	28	96	28	86	24	110	28	98	28	101	24	127	28	112	25	118
1829	M22960_at	19	66	21	109	20	136	20	140	20	141	19	160	18	167	21	144	20	145	23	126
1834	M23197_at			29	65	40	59	36	72	41	61	50	53	39	74	44	66	44	62	40	72
1882	M27891_at	6	247	7	364	7	405	7	420	7	463	7	446	7	429	7	476	7	480	7	486
1928	M31303_rna1_at	24	58	25	91	23	100	29	82	29	87	24	114	24	115	31	96	30	100	26	108
1941	M31994_at							54	50	53	50	45	60	49	55			47	59	54	55
1962	M33680_at	12	123	12	220	11	228	11	247	11	238	11	270	12	247	12	264	12	269	11	275
1975	M34344_at					39	60	31	77	40	62	35	80	32	84	32	83	33	91	37	82
2020	M55150_at			31	61	30	81	41	65	31	78	30	88	33	81	34	82	34	90	33	89
2111	M62762_at			32	60	29	82	24	105	27	95	29	90	26	104	25	124	27	117	27	105
2121	M63138_at	4	376	4	562	4	654	4	622	4	658	4	629	4	648	5	718	4	721	5	689
2288	M84526_at	5	345	3	592	3	728	2	781	2	793	2	792	2	898	2	875	2	875	2	914
2335	M89957_at											49	53	53	53	52	52		770	50	59
2354	M92287_at	2	532	2	733	2	767	3	775	3	766	3	732	3	765	3	791	3	50	3	780
2394	M95678_at																	53			
2402	M96326_rna1_at			34	58	33	69	34	73	35	69	31	84	37	75	30	104	32	91	36	84
2408	M96803_at					41	57	46	58	46	58	51	53					46	61		
2642	U05259_rna1_at	9	186	9	279	9	329	9	348	8	368	10	329	9	356	8	372	10	339	8	411
3252	U46499_at	18	73	17	140	17	151	18	166	14	196	16	189	17	177	15	194	15	189	14	201
3258	U46751_at					46	52			52	52	43	62	47	55	38	78	38	71	59	50
3320	U50136_rna1_at			20	116	26	96	21	112	28	93	23	121	27	101	28	113	26	121	28	105
3984	U94855_at					47	52							51	54			49	57	43	68
4050	X03934_at							51	54	45	58	52	52	54	53	51	53	54	50	44	65
4196	X17042_at	21	61	18	125	19	136	17	166	15	188	13	213	16	194	17	180	17	180	16	199
4211	X51521_at	10	179	10	254	12	222	12	228	12	233	12	255	11	281	11	265	11	278	12	268
4229	X52056_at											46	57	45	57	47	59			56	51
4291	X56468_at					51	51	42	63	49	55									58	51
4328	X59417_at	14	104	11	225	10	317	10	309	9	360	9	369	10	343	9	366	9	389	10	356
4373	X62320_at			35	57	32	70	33	74	36	69	39	72	38	74	37	79	35	80	35	85
4377	X62654_rna1_at	17	92	15	164	15	184	16	182	19	145	20	149	20	162	16	183	21	144	20	142
4409	X64594_at									44	59	47	55	52	54	54	51			55	54
4438	X66401_cds1_at					48	52	53	52					50	55	48	58	42	65	53	56
4847	X95735_at	1	1093	1	1356	1	1399	1	1515	1	1489	1	1508	1	1545	1	1486	1	1540	1	1553
4951	Y07604_at	13	123	16	161	13	194	14	202	13	212	14	200	13	208	14	196	14	190	17	195
5445	X04526_at			41	52	45	55	47	58	43	60	55	51	56	51	55	51	50	57	51	58
5501	Z15115_at			22	98	27	96	22	111	21	130	22	130	21	133	23	127	24	128	22	127
5772	U22376_cds2_s_at	23	58	24	92	22	116	25	101	23	115	26	108	25	112	22	128	22	138	21	128
5950	M29610_s_at			43	50	35	64	40	68	51	53	38	74	36	78	46	63	37	73	39	76
5952	U05255_s_at			40	53	36	63	30	79	30	80	32	83	31	95	27	114	29	104	29	97
6041	L09209_s_at	8	213	8	342	8	354	8	355	10	340	8	420	8	380	10	366	8	392	9	361
6049	U89922_s_at					49	51			48	55	53	51					51	56	52	56
6079	U59632_s_at					50	51			42	60	56	50	42	67			45	62	47	63
6184	M26708_s_at			30	62	43	56	44	60	34	71	48	54	44	59	49	56	43	64	46	64
6200	M28130_rna1_s_at			27	86	21	121	23	105	22	127	21	135	22	126	20	147	19	149	19	172
6201	Y00787_s_at					42	56	39	69	38	66	36	77	30	96	36	79	36	79	32	91
6215	M19508_xpt3_s_at	20	63	26	88	24	99	26	96	26	97	25	109	23	117	29	105	25	123	24	123
6225	M84371_rna1_s_at							52	52							53	52	52	53	49	60
6376	M83652_s_at	25	50	39	53	37	62	35	73	50	55	34	82	35	78	35	81	39	71	45	65
6539	X85116_rna1_s_at			38	55	31	70	43	60	32	75	33	82	43	62	41	71	40	68	41	71
6702	X97267_rna1_s_at							48	57			44	60	48	55	45	63			42	70
6796	J02982_f_at							50	57	47	56			55	52	50	54			57	51
6855	M31523_at	11	125	14	171	14	186	15	191	16	183	15	197	14	203	18	177	16	187	15	200
7119	U29175_at													46	56						

Continued on Next Page...

Table B.9 – *Continued*

(c) Tanh-based system																					
		Fitness Evaluation																			
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at					35	60	38	62	41	56	40	53	39	56	38	63	46	51	42	53
668	D86967_at			34	58	36	59	34	70			32	76	32	79	34	74	40	58	34	66
758	D88270_at	21	72	19	121	18	145	21	123	18	157	20	137	20	138	21	129	19	144	20	161
760	D88422_at	14	102	17	145	17	172	16	170	17	193	15	206	17	190	14	213	15	208	15	193
804	HG1612-HT1612_at	5	434	5	620	5	682	6	647	6	653	6	629	6	659	6	643	6	651	6	618
1144	J05243_at			36	51	39	56	39	58	38	61			42	53	36	65	38	59	36	62
1239	L07633_at	18	85	23	104	26	89	26	92	22	108	25	95	22	109	27	92	30	89	27	94
1630	L47738_at			33	60	37	58	41	56	40	56	37	65	37	58	42	53	36	68	38	61
1685	M11722_at	6	337	6	585	6	648	5	655	5	712	5	691	5	709	5	695	5	675	5	672
1704	M13792_at					40	53	37	63	37	61	36	65	35	69	35	67	32	70	39	61
1745	M16038_at					43	50	40	57	42	55			41	53	40	57	37	61	41	56
1779	M19507_at	8	206	8	365	8	405	8	414	8	447	8	406	8	455	8	440	8	420	8	414
1796	M20902_at			37	50	34	62	32	71	34	65	34	74	34	72	31	83	33	70	35	63
1809	M21624_at					41	53	36	64	44	54	39	56							43	52
1829	M22960_at	24	54	29	70	31	79	30	77	33	79	27	91	27	88	28	91	29	90	24	101
1834	M23197_at			32	63	29	81	35	70	31	80	33	74	33	76	41	55	35	69	28	90
1882	M27891_at	1	1177	1	1557	1	1658	1	1599	1	1644	1	1620	1	1665	1	1662	1	1683	1	1676
1928	M31303_rna1_at			27	80	24	92	27	87	29	87	29	86	31	82	32	79	27	95	32	72
1941	M31994_at																	42	56		
1962	M33680_at	13	108	14	178	12	196	13	211	14	203	13	224	13	214	12	237	14	221	13	237
1975	M34344_at					44	50							44	52	45	50	39	58		
2121	M63138_at	11	157	11	237	11	201	10	267	11	249	11	236	11	249	11	263	12	228	11	255
2288	M84526_at	4	538	2	988	2	1127	2	1203	2	1164	2	1254	2	1229	2	1291	2	1242	2	1287
2354	M92287_at	2	813	3	932	4	892	4	946	4	936	4	969	4	940	3	934	4	960	4	877
2402	M96326_rna1_at	16	101	13	180	16	185	14	193	12	223	12	225	12	218	13	221	13	223	14	215
2408	M96803_at															43	53				
2642	U05259_rna1_at	7	287	7	452	7	516	7	520	7	512	7	513	7	530	7	500	7	514	7	534
3252	U46499_at	23	54	22	109	20	124	19	137	20	142	21	128	18	152	18	146	18	152	19	170
4050	X03934_at			31	66	28	85	29	81	28	88	30	78	23	99	30	87	34	69	31	85
4196	X17042_at									45	53					44	52	44	53		
4211	X51521_at	10	166	12	211	15	189	12	233	16	194	14	217	14	214	15	198	11	231	12	246
4229	X52056_at			30	66	33	78	33	70	27	90	31	77	30	86	26	92	28	93	29	85
4328	X59417_at	12	114	10	239	10	279	11	263	10	309	10	305	10	308	9	331	9	328	10	330
4373	X62320_at			24	90	23	97	25	95	24	92	28	90	29	86	25	96	24	100	25	99
4377	X62654_rna1_at	19	78	18	138	21	124	20	136	19	150	19	150	21	137	20	143	21	126	21	138
4438	X66401_cds1_at					45	50													44	51
4680	X82240_rna1_at			26	89	32	78	23	103	23	103	26	94	25	94	24	98	23	109	23	104
4847	X95735_at	3	719	4	898	3	980	3	973	3	956	3	982	3	1003	4	922	3	1020	3	988
4951	Y07604_at					46	50			43	55			43	53					45	51
5191	Z69881_at					38	56														
5501	Z15115_at			21	112	19	140	17	161	21	124	18	162	19	152	19	144	20	143	18	179
5542	M37271_s_at									46	52										
5543	D00749_s_at											41	51	40	54			45	52		
5772	U22376_cds2_s_at	15	102	16	173	14	192	15	181	13	205	16	181	15	199	16	188	17	187	16	188
5950	M29610_s_at											42	50								
6041	L09209_s_at	9	187	9	307	9	322	9	351	9	363	9	366	9	378	10	319	10	325	9	379
6049	U89922_s_at	20	75	20	115	25	89	22	109	25	92	24	101	28	87	29	89	26	96	26	97
6079	U59632_s_at									39	57										
6200	M28130_rna1_s_at			28	70	30	80	28	86	32	79	23	106	26	91	23	104	22	113	30	85
6201	Y00787_s_at			35	51	27	85	31	76	30	83	35	71	38	56	33	74	31	73	33	68
6271	M33493_s_at							43	53	35	63	38	63			39	60	43	54	40	61
6376	M83652_s_at	22	57	25	89	22	98	24	102	26	92	22	115	24	94	22	105	25	97	22	112
6388	S54005_s_at							44	51												
6702	X97267_rna1_s_at					42	51	42	55	36	63			36	65	37	64	41	56	37	61
6855	M31523_at	17	93	15	177	13	192	18	154	15	201	17	169	16	193	17	172	16	199	17	186
7119	U29175_at									47	52					46	50				

Continued on Next Page...

Table B.9 – *Continued*

(d) Threshold-based system																					
		Fitness Evaluation																			
Gene	Accession	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at			29	56	34	59	30	82	32	85	33	81	29	86	36	71	36	74	32	80
668	D86967_at					39	51	39	53	39	58	39	68	41	59	38	61	45	53	47	55
758	D88270_at							41	52			44	56					41	59	41	64
760	D88422_at							40	52							43	55			45	58
804	HG1612-HT1612_at	5	211	7	335	7	351	7	327	7	351	7	358	7	390	7	357	7	382	7	358
1239	L07633_at	13	59	23	76	21	95	28	86	26	97	29	90	34	82	28	99	26	102	28	91
1674	M11147_at							37	54			38	68	31	83	44	53	43	54	38	69
1685	M11722_at	18	52	13	129	16	137	13	164	15	147	13	174	13	176	14	159	15	165	14	174
1745	M16038_at									40	57										
1779	M19507_at	8	122	6	339	6	444	5	516	4	574	3	714	3	657	3	642	3	651	3	650
1796	M20902_at			20	79	18	104	21	118	18	122	18	151	17	139	16	147	18	138	21	127
1829	M22960_at			21	77	20	95	19	119	27	96	24	117	20	122	18	141	25	110	19	144
1834	M23197_at			30	52	37	52			41	54	37	69	42	56	41	58	44	53		
1882	M27891_at	3	240	4	438	4	557	3	599	3	587	4	599	4	606	4	628	5	611	4	593
1928	M31303_rna1_at					38	51			37	63	42	64	35	77	37	64	39	66	43	63
1941	M31994_at									38	59	41	64	39	66	46	51	42	55	37	70
1962	M33680_at	12	65	15	100	14	138	16	133	14	161	16	159	16	148	17	143	16	146	17	150
1975	M34344_at			28	58	35	54	24	92	23	102	21	131	25	109	22	131	20	117	25	104
2020	M55150_at			26	66	28	66	27	86	30	85	32	81	27	89	30	80	29	98	29	88
2111	M62762_at			24	68	26	71	29	82	28	93	25	108	26	99	25	111	23	111	26	97
2121	M63138_at	4	217	5	417	5	474	6	493	6	532	5	577	5	568	5	568	4	618	5	587
2288	M84526_at	6	192	3	489	2	572	2	712	2	698	2	858	2	806	2	779	2	825	2	846
2354	M92287_at	2	331	2	491	3	565	4	576	5	546	6	565	6	529	6	539	6	542	6	559
2394	M95678_at																	40	65		
2402	M96326_rna1_at					36	53	31	73	34	75	26	107	28	86	34	73	32	85	30	87
2546	S82470_at																	46	51		
2642	U05259_rna1_at	11	75	10	143	9	188	10	207	11	203	12	221	11	226	12	207	12	194	12	213
3252	U46499_at			16	91	15	137	14	159	13	167	15	166	14	175	15	148	13	182	13	184
3258	U46751_at											36	70	43	55	47	50	38	68	48	54
3320	U50136_rna1_at			25	68	22	89	17	126	19	121	19	137	21	119	20	133	19	126	18	149
4095	X06948_at											48	51			45	51	48	50	46	56
4196	X17042_at	17	53	12	130	11	175	9	213	9	249	8	286	9	245	9	260	9	266	9	269
4211	X51521_at	10	93	11	135	13	142	15	158	16	139	14	171	15	160	13	190	14	169	15	169
4328	X59417_at	14	58	14	127	12	166	12	200	10	218	10	248	10	243	11	223	10	241	10	242
4373	X62320_at							38	54			45	53	46	51						
4377	X62654_rna1_at	16	55	19	85	24	84	22	105	22	105	23	119	19	126	26	102	27	99	24	106
4409	X64594_at									42	52	49	51					49	50	44	61
4847	X95735_at	1	746	1	964	1	1154	1	1220	1	1196	1	1311	1	1239	1	1294	1	1255	1	1265
4951	Y07604_at	7	126	9	169	10	185	11	203	12	200	11	238	12	196	10	243	11	224	11	226
5445	X04526_at							36	55												
5501	Z15115_at			32	50	30	63	26	91	29	87	28	100	32	83	27	100	30	95	34	77
5772	U22376_cds2_s_at			22	76	23	87	25	92	25	98	43	61	33	82	32	76	33	85	42	64
5950	M29610_s_at					29	65	34	60	33	79	31	86	37	73	29	89	35	77	27	94
5952	U05255_s_at			27	61	25	82	23	103	20	116	17	158	18	131	19	140	17	139	16	161
6041	L09209_s_at			8	212	8	249	8	237	8	292	9	252	8	274	8	304	8	282	8	278
6049	U89922_s_at	9	98									47	52	45	52	42	56	47	51	40	65
6079	U59632_s_at					31	63	35	57	31	85	30	86	38	69	33	75	31	86	31	83
6184	M26708_s_at													44	53						
6200	M28130_rna1_s_at			31	51	27	66	32	72	21	108	22	123	24	112	24	113	22	113	23	120
6201	Y00787_s_at					32	62	33	70	36	70	34	77	36	73	35	71	28	98	33	77
6215	M19508_xpt3_s_at			18	86	19	99	18	122	17	123	20	137	22	119	21	133	24	111	22	123
6376	M83652_s_at											46	53			40	59			39	67
6539	X85116_rna1_s_at					33	61			35	71	35	74	30	84	31	78	37	72	35	75
6796	J02982_f_at					40	50			43	52	40	66	40	61	39	61	34	77	36	74
6855	M31523_at	15	56	17	90	17	115	20	118	24	100	27	107	23	114	23	118	21	114	20	138
7128	M71243_f_at											50	50								

Table B.10: The complete list of SRBCTs genes in the population size 100.

(a) Sigmoid-based system																					
		Fitness Evaluation																			
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652			30	71	23	105	25	115	22	131	30	108	26	129	23	126	27	124	27	124
85	297392					25	100	49	65	32	98	29	108	39	86	31	109	36	102	30	110
107	365826			28	77	38	77	41	77	42	72	34	99	36	90	37	92	30	109	31	107
123	236282					45	64	59	51	54	59	52	63	52	67	58	60	52	69	46	81
129	298062	19	57	19	103	19	142	21	133	19	150	18	181	19	168	19	170	21	144	21	149
153	383188	15	97	15	203	15	215	14	250	14	273	14	259	13	273	15	257	14	280	13	306
165	283315					53	53	43	76	48	67	43	70	44	74	54	66	48	72	49	74
187	296448	9	176	8	356	9	434	9	446	8	497	7	552	7	542	7	576	7	596	7	624
188	435953									63	53					69	50			63	59
236	878280					52	54	46	72	58	57	44	69	62	54	51	72	47	74	39	94
246	377461	3	367	4	622	4	724	4	736	4	805	4	798	4	843	4	874	4	890	5	850
255	325182	11	119	12	245	13	247	13	274	13	273	13	290	14	271	13	296	12	295	12	340
335	1469292	20	54	22	95	24	100	23	119	24	129	24	120	63	54	26	115	26	126	25	127
365	1434905					56	53			62	53			23	134	46	78	66	50	58	61
368	1473131					48	59	47	71	46	69	51	65	48	71	53	66	53	68	47	76
380	289645													60	55	57	60	60	55	60	61
417	395708			42	56	35	78	36	84	41	78	39	80	34	95	39	88	35	103	35	99
430	379708			41	58	39	77	37	81	35	95	37	90	41	81	36	94	42	88	40	93
509	207274	2	423	2	730	2	823	3	824	3	861	3	891	3	929	3	907	2	973	3	928
545	1435862	5	277	6	477	6	572	6	627	6	651	6	640	6	647	6	695	6	702	6	727
554	461425	17	94	17	170	16	215	15	223	17	220	17	219	15	267	16	254	17	254	15	266
566	357031					55	53	42	76	52	63	46	69	50	69	43	80	55	65	64	58
603	42558							53	59			63	53	66	52	47	76			67	54
607	811108																	65	50		
714	245330															71	50				
742	812105	1	527	1	841	1	940	1	935	1	985	1	1030	1	996	1	1055	1	1073	1	1025
758	47475							55	57							65	51			56	62
783	767183					47	61	45	74	44	70	42	75	38	89	49	74	34	103	41	87
836	241412			35	67	37	77	32	93	45	70	38	86	33	99	35	96	31	106	43	87
842	810057			36	65	36	78	39	81	43	71	40	77	43	79	42	83	45	83	45	84
846	183337	13	106	14	204	17	195	17	211	16	228	16	236	16	251	17	214	16	256	16	250
910	839552							50	63	61	55	57	56	51	68			50	72	62	60
951	841620					54	53			51	64	50	65	54	65	44	79	44	85	44	85
976	786084			29	73	27	96	24	115	23	130	20	130	29	122	27	114	24	139	29	115
1003	796258	16	94	16	200	14	228	16	212	15	256	15	248	17	233	14	265	15	262	17	246
1055	1409509			23	88	32	90	27	107	28	109	28	109	25	130	25	120	29	111	26	125
1066	486110																	68	50	68	54
1084	878652			26	86	26	99	26	109	34	96	25	118	24	132	32	109	25	135	23	140
1105	788107							48	66	60	55			53	67	59	60	62	54	53	64
1116	626502			32	69	40	70	44	75	33	97	47	68	40	83	41	87	38	95	36	98
1158	814526			24	87	21	121	22	128	26	119	21	128	20	157	20	150	18	180	22	145
1207	143306			20	101	22	119	20	139	20	141	27	112	28	126	21	148	19	165	20	157
1263	324494							54	58	50	64	65	51					56	62		
1301	346696													61	55					70	54
1319	866702	14	103	13	208	12	248	12	276	12	288	12	319	12	284	12	308	13	285	14	269
1327	491565			33	67	44	64	33	87	37	85	33	101	42	79	34	97	32	105	42	87
1386	745019			39	62	34	83	28	102	29	109	31	107	31	106	30	114	33	104	28	116
1387	770394					57	52			65	50	64	52			60	58			73	51
1389	770394	6	255	5	579	5	694	5	729	5	799	5	796	5	808	5	854	5	872	4	889
1434	784257			43	53	33	84	40	79	31	100	35	91	35	93	29	114	39	95	34	100
1489	505491					49	57									64	51	64	53	74	51
1497	203003			44	50	41	69	52	59	56	58	41	76	55	63	45	79	61	55	52	66
1536	530185													59	56						
1601	629896	7	234	7	429	7	503	7	485	7	513	8	537	8	542	8	496	8	572	8	513
1606	624360	18	61	18	130	18	145	18	151	18	175	19	162	18	176	18	198	20	144	19	161
1613	80338			37	62	42	68	51	61	49	66	48	68	49	70	52	70	46	77	48	76
1645	52076	12	117	11	264	11	325	11	332	11	363	11	345	11	360	11	384	11	359	11	390
1662	377048									47	68	60	54	64	53	63	54	57	61	50	72
1700	796475							56	57			54	60	58	59			63	54	72	51
1708	43733			31	70	31	91	34	85	30	103	36	90	30	110	28	114	40	91	37	96
1738	771323			40	59	51	56	38	81	40	78	53	60	37	90	48	75	43	87	51	71
1764	44563			25	87	30	92	30	100	27	115	23	120	27	128	24	124	28	116	24	139
1776	768246									57	57	55	59	57	60	70	50	51	71	57	62
1884	609663			38	62	28	96	35	85	36	95	32	102	32	104	40	87	41	88	38	95
1909	789357																			65	56
1911	898219					43	64	58	55			59	54	56	61	61	56	67	50	54	64
1915	840942											62	53								
1916	80109			21	100	29	95	31	100	25	127	22	123	21	156	33	108	23	140	32	106
1932	782811			34	67	46	63	29	100	38	83	49	67	47	71	38	90	37	101	33	105
1954	814260	10	146	9	355	8	436	8	450	9	454	9	487	9	498	9	483	9	543	9	500
1955	784224	4	319	3	640	3	768	2	865	2	929	2	932	2	940	2	913	3	934	2	961
1980	841641									64	51	61	54			67	50	59	55	69	54
2046	244618	8	186	10	306	10	344	10	407	10	411	10	431	10	434	10	482	10	438	10	456
2050	295985			27	84	20	122	19	140	21	138	26	117	22	145	22	147	22	141	18	168
2144	308231											66	50	65	53	66	51			55	63
2146	293500															68	50				
2157	244637					58	52	57	56	55	58	58	55	45	73	55	66	49	72	71	53
2162	308163									53	60	67	50			62	55			66	56
2186	208699									59	56	56	59	67	52	56	62	58	58	59	61
2199	135688					50	57			39	78	45	69	46	72	50	74	54	66	61	61

Continued on Next Page...

Table B.10 – *Continued*

(b) Linear-based system																			
		Fitness Evaluation																	
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652			27	92	29	98	27	114	27	127	27	136	29	122	27	131	27	130	26	132
74	193913					49	50	40	74					49	56	25	138	51	58	49	64
85	297392	25	50	25	93	25	113	30	110	25	128	29	116	25	140	53	60	26	159	28	127
107	365826													59	50					51	62
123	236282					44	59			55	50	48	60	48	57	45	69	55	53	48	64
153	383188	10	121	10	224	8	301	10	309	9	345	8	379	11	325	10	345	11	337	8	418
165	283315			37	58	30	92	37	81	31	105	36	95	32	104	33	103	32	113	30	124
166	897177							51	51												
187	296448	14	97	11	209	11	274	8	320	8	349	9	349	8	347	8	360	8	371	9	370
188	435953																			58	50
236	878280							45	64	42	69	41	78	42	76	47	66	49	66	41	89
246	377461	7	166	6	363	6	448	7	464	7	474	6	504	6	515	7	524	6	517	6	513
251	486787													52	53						
255	325182	15	94	16	170	14	237	14	232	14	270	12	288	14	276	13	265	12	337	13	298
335	1469292	20	69	19	113	21	132	20	148	23	154	24	147	22	153	22	159	18	187	21	165
365	1434905									50	56			53	52			52	58	50	64
380	289645															46	67	58	51	57	52
417	395708					41	63	38	77	40	79	38	90	41	81	32	107	38	93	39	98
430	379708									56	50										
509	207274	2	344	2	500	3	539	4	583	3	636	4	663	4	616	4	648	4	650	5	637
545	1435862	3	233	5	396	5	459	5	498	5	597	5	600	5	569	5	608	5	625	4	642
554	461425			28	87	35	82	29	114	30	108	28	130	28	122	29	111	30	123	25	144
585	68977													56	51						
603	42558									46	59					55	55	50	59		
742	812105	1	499	1	757	1	877	1	921	1	984	1	947	1	1001	1	1034	1	1046	1	1024
758	47475																	54	53		
783	767183			35	65	33	87	39	76	41	72	33	99	40	83	36	97	34	104	36	110
836	241412	24	50	33	69	37	78	31	97	39	82	34	99	34	97	39	93	35	101	37	105
842	810057			36	63	31	90	34	84	32	103	30	112	35	95	34	100	37	96	32	112
846	183337	9	135	13	197	12	256	12	256	12	294	14	282	13	293	11	315	13	314	12	298
910	839552					43	60	43	71	52	55	51	51	43	74	49	64	48	67	45	78
951	841620			39	56	36	81	36	82	38	83	37	94	33	100	35	99	39	90	35	110
976	786084	23	53	26	93	28	103	22	135	24	148	25	143	26	138	28	128	24	164	24	145
1003	796258	11	116	8	236	13	255	13	253	13	281	13	286	12	313	14	263	14	294	14	297
1055	1409509			32	72	34	86	33	84	35	89	32	104	38	88	30	109	31	120	34	111
1067	489489					50	50											60	50	55	54
1084	878652			31	80	32	89	35	83	33	103	35	95	31	112	40	89	33	105	38	101
1116	626502			29	84	26	112	28	114	26	127	26	142	27	131	26	137	29	130	29	126
1158	814526	22	53	23	100	24	123	19	154	19	178	20	173	23	149	20	167	19	184	20	169
1203	144881																	57	52		
1207	143306			17	121	22	125	21	142	22	162	19	179	21	158	18	172	20	181	18	176
1295	344134													57	50			53	56	52	58
1319	866702	18	74	21	110	18	154	23	131	21	171	21	160	19	167	24	144	17	191	23	153
1327	491565			30	81	27	104	25	117	29	119	31	104	30	119	31	107	28	130	31	114
1386	745019			34	69	39	67	32	85	36	83	39	88	36	90	38	93	40	86	33	112
1387	770394							50	52	48	59	45	66	45	70	51	62			47	65
1389	770394	6	186	4	435	4	517	3	592	4	620	3	681	3	670	3	701	3	728	3	763
1434	784257					42	61	42	73	34	89	40	79	39	84	37	94	36	98	44	79
1497	203003			38	57	40	65	46	59	43	66	43	76	37	88	42	79	43	81	43	81
1536	530185					48	52			45	60	50	52			57	53	46	68		
1601	629896	5	193	7	347	7	437	6	472	6	499	7	499	7	494	6	537	7	497	7	513
1606	624360	13	102	15	172	16	179	16	201	16	212	15	235	16	238	15	244	15	240	15	232
1613	80338													50	54	54	59	56	52		
1626	811000					46	55	47	56	49	57			47	64	44	70	47	68	46	75
1634	82903															58	51				
1645	52076	16	76	22	103	20	149	17	156	18	188	18	179	18	173	21	165	23	165	19	174
1662	377048											53	51								
1738	771323			40	55	38	70	41	73	37	83	42	78	44	70	41	87	44	79	40	98
1776	768246							48	54	54	52	52	51	58	50	52	61	59	51		
1884	609663	21	57	24	99	23	124	24	124	28	125	23	154	24	142	23	158	25	162	27	130
1911	898219							52	50	47	59	44	68	51	53	50	64	41	86	56	53
1915	840942															56	53				
1916	80109	19	71	20	110	19	149	26	115	20	172	22	156	17	177	19	168	22	169	22	157
1932	782811	17	75	14	183	15	187	15	210	15	234	16	224	15	243	16	220	16	234	16	219
1954	814260	8	137	9	231	10	283	9	314	10	331	10	349	9	337	9	351	9	353	10	334
1955	784224	4	209	3	472	2	622	2	665	2	685	2	729	2	726	2	784	2	752	2	843
2046	244618	12	113	12	203	9	283	11	284	11	302	11	324	10	331	12	293	10	341	11	298
2050	295985			18	117	17	158	18	154	17	204	17	180	20	167	17	197	21	180	17	194
2146	293500													54	52						
2157	244637					45	59	44	70	44	64	47	65	46	66	43	75	42	83	42	83
2186	208699							49	52	51	56	49	59	55	51	48	66	45	69	54	55
2199	135688					47	54			53	55	46	65							53	57

Continued on Next Page...

Table B.10 – *Continued*

(c) Tanh-based system																					
		Fitness Evaluation																			
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652	**8**	*127*	**28**	*66*	**29**	*79*	**28**	*96*	**27**	*97*	**28**	*99*	**34**	*93*	**37**	*84*	**33**	*89*	**27**	*112*
85	297392			**38**	*51*	**42**	*56*	**44**	*60*	**43**	*67*	**44**	*67*	**42**	*75*	**36**	*86*	**47**	*66*	**46**	*70*
107	365826					**33**	*73*	**34**	*80*	**37**	*75*	**35**	*79*	**36**	*87*	**43**	*73*	**26**	*106*	**26**	*113*
123	236282							**41**	*66*	**57**	*51*	**19**	*153*	**47**	*63*	**59**	*53*	**56**	*56*	**48**	*68*
129	298062			**18**	*102*	**18**	*123*	**18**	*136*	**18**	*149*			**19**	*153*	**19**	*176*	**18**	*167*	**17**	*192*
153	383188			**17**	*126*	**16**	*154*	**17**	*157*	**16**	*197*	**16**	*204*	**15**	*227*	**15**	*235*	**15**	*260*	**15**	*246*
165	283315									**54**	*52*					**56**	*53*	**57**	*53*	**53**	*61*
187	296448			**8**	*302*	**8**	*390*	**7**	*462*	**7**	*492*	**7**	*501*	**7**	*554*	**7**	*519*	**7**	*539*	**7**	*544*
236	878280					**38**	*64*	**49**	*53*	**42**	*67*	**34**	*84*	**39**	*80*	**46**	*69*	**43**	*75*	**42**	*81*
246	377461	**3**	*281*	**4**	*535*	**4**	*674*	**4**	*718*	**4**	*735*	**4**	*789*	**3**	*787*	**5**	*731*	**4**	*793*	**4**	*848*
255	325182	**10**	*106*	**14**	*187*	**14**	*212*	**13**	*258*	**12**	*282*	**12**	*252*	**14**	*234*	**13**	*276*	**12**	*303*	**13**	*276*
335	1469292			**27**	*69*	**24**	*92*	**29**	*96*	**26**	*98*	**30**	*93*	**32**	*96*	**28**	*105*	**31**	*92*	**29**	*112*
365	1434905							**47**	*55*	**48**	*57*	**57**	*52*	**52**	*58*	**53**	*55*	**66**	*50*	**61**	*54*
368	1473131			**35**	*54*	**40**	*63*	**35**	*78*	**38**	*73*	**45**	*66*	**40**	*77*	**34**	*89*	**35**	*83*	**41**	*81*
380	289645											**60**	*50*	**63**	*50*			**65**	*50*	**54**	*61*
417	395708			**34**	*55*	**31**	*78*	**33**	*82*	**29**	*94*	**37**	*74*	**29**	*103*	**29**	*97*	**45**	*70*	**33**	*96*
430	379708					**32**	*76*	**32**	*84*	**30**	*92*	**27**	*102*	**30**	*101*	**31**	*96*	**30**	*96*	**35**	*90*
509	207274	**1**	*364*	**2**	*645*	**3**	*685*	**3**	*738*	**2**	*826*	**3**	*812*	**2**	*809*	**2**	*884*	**3**	*845*	**2**	*875*
545	1435862	**5**	*219*	**6**	*423*	**6**	*492*	**6**	*474*	**6**	*552*	**6**	*549*	**6**	*568*	**6**	*602*	**6**	*601*	**6**	*634*
554	461425	**16**	*57*	**13**	*188*	**12**	*224*	**14**	*225*	**14**	*224*	**14**	*241*	**12**	*287*	**14**	*250*	**13**	*267*	**14**	*275*
566	357031			**39**	*51*	**37**	*66*	**37**	*72*	**35**	*77*	**47**	*64*	**33**	*95*	**32**	*92*	**37**	*80*	**36**	*89*
603	42558											**58**	*52*			**62**	*50*	**49**	*64*	**60**	*55*
714	245330											**56**	*52*	**60**	*52*	**58**	*53*			**56**	*59*
742	812105	**2**	*362*	**1**	*702*	**1**	*769*	**1**	*852*	**1**	*850*	**1**	*827*	**1**	*886*	**1**	*917*	**2**	*875*	**1**	*920*
758	47475																	**64**	*50*		
783	767183			**36**	*52*			**48**	*55*	**51**	*55*	**53**	*58*	**59**	*53*	**49**	*63*	**46**	*70*	**38**	*88*
836	241412					**35**	*68*	**42**	*65*	**40**	*70*	**49**	*64*	**45**	*65*	**54**	*54*	**44**	*73*	**47**	*70*
842	810057			**37**	*52*	**39**	*63*	**40**	*67*	**41**	*68*	**48**	*64*	**35**	*92*	**47**	*66*	**40**	*80*	**40**	*83*
846	183337	**15**	*66*	**16**	*148*	**17**	*143*	**16**	*171*	**17**	*168*	**17**	*202*	**18**	*158*	**17**	*193*	**17**	*209*	**18**	*178*
910	839552							**51**	*51*	**52**	*52*	**51**	*60*	**54**	*57*	**52**	*56*	**63**	*51*	**50**	*63*
951	841620											**54**	*58*	**57**	*54*	**44**	*73*	**48**	*65*	**63**	*53*
976	786084			**30**	*66*	**25**	*91*	**25**	*101*	**22**	*120*	**24**	*111*	**28**	*103*	**21**	*127*	**24**	*119*	**25**	*116*
1003	796258	**14**	*67*	**15**	*168*	**15**	*211*	**15**	*203*	**15**	*218*	**15**	*219*	**16**	*210*	**16**	*215*	**16**	*228*	**16**	*242*
1055	1409509			**23**	*87*	**27**	*87*	**24**	*102*	**25**	*114*	**32**	*91*	**22**	*119*	**33**	*91*	**29**	*101*	**28**	*112*
1084	878652			**24**	*83*	**26**	*91*	**27**	*98*	**21**	*123*	**21**	*122*	**21**	*123*	**24**	*111*	**28**	*103*	**31**	*108*
1105	788107											**52**	*58*	**53**	*57*			**55**	*57*	**58**	*56*
1116	626502							**43**	*63*	**44**	*63*	**50**	*61*	**58**	*54*	**60**	*52*	**51**	*61*	**51**	*63*
1158	814526			**29**	*66*	**28**	*87*	**23**	*105*	**31**	*90*	**25**	*110*	**27**	*107*	**27**	*105*	**25**	*108*	**22**	*123*
1159	142788															**63**	*50*				
1207	143306			**25**	*74*	**23**	*93*	**26**	*101*	**28**	*95*	**20**	*126*	**23**	*118*	**26**	*109*	**23**	*120*	**24**	*120*
1263	324494									**56**	*51*							**62**	*51*	**64**	*52*
1301	346696																			**67**	*50*
1319	866702	**13**	*89*	**12**	*193*	**13**	*220*	**12**	*267*	**13**	*269*	**13**	*251*	**13**	*270*	**12**	*284*	**14**	*262*	**12**	*294*
1327	491565			**33**	*57*	**44**	*55*	**38**	*67*	**45**	*62*	**41**	*70*	**38**	*81*	**35**	*87*	**36**	*83*	**43**	*79*
1386	745019			**31**	*61*	**36**	*66*	**36**	*73*	**34**	*81*	**31**	*93*	**43**	*74*	**38**	*81*	**39**	*80*	**44**	*73*
1389	770394	**6**	*198*	**5**	*468*	**5**	*591*	**5**	*657*	**5**	*687*	**5**	*723*	**5**	*725*	**4**	*782*	**5**	*769*	**5**	*764*
1434	784257			**32**	*58*	**41**	*59*	**30**	*88*	**32**	*90*	**26**	*106*	**25**	*110*	**30**	*96*	**32**	*90*	**32**	*100*
1497	203003													**61**	*51*			**59**	*53*		
1601	629896	**7**	*192*	**7**	*356*	**7**	*410*	**8**	*447*	**8**	*441*	**8**	*483*	**9**	*444*	**8**	*457*	**8**	*488*	**9**	*492*
1606	624360			**22**	*90*	**22**	*93*	**21**	*114*	**24**	*116*	**22**	*116*	**31**	*101*	**25**	*110*	**22**	*124*	**21**	*129*
1613	80338					**45**	*53*			**39**	*71*	**36**	*74*	**41**	*76*	**40**	*76*	**41**	*77*	**34**	*93*
1645	52076	**12**	*96*	**11**	*208*	**11**	*258*	**11**	*291*	**11**	*294*	**11**	*300*	**11**	*335*	**11**	*320*	**11**	*331*	**11**	*352*
1662	377048									**55**	*51*	**42**	*68*	**44**	*70*	**45**	*71*	**50**	*62*	**68**	*50*
1700	796475													**62**	*50*	**55**	*54*				
1708	43733			**19**	*101*	**21**	*107*	**22**	*111*	**23**	*118*	**29**	*97*	**24**	*116*	**22**	*117*	**21**	*126*	**23**	*123*
1738	771323					**46**	*52*			**50**	*56*	**59**	*51*	**50**	*60*	**42**	*73*	**54**	*58*	**52**	*62*
1764	44563			**20**	*94*	**19**	*111*	**20**	*115*	**19**	*139*	**18**	*156*	**17**	*174*	**18**	*188*	**19**	*158*	**20**	*160*
1776	768246									**53**	*52*	**55**	*53*	**48**	*63*	**57**	*53*			**66**	*51*
1884	609663			**40**	*50*	**34**	*72*	**39**	*67*	**46**	*61*	**38**	*74*	**37**	*85*	**48**	*65*	**42**	*76*	**39**	*86*
1911	898219							**50**	*52*	**58**	*50*	**43**	*68*	**46**	*64*			**58**	*53*	**57**	*58*
1916	80109			**26**	*72*	**30**	*79*	**31**	*86*	**33**	*81*	**33**	*89*	**26**	*110*	**23**	*114*	**27**	*106*	**30**	*110*
1932	782811							**46**	*55*	**49**	*56*	**40**	*71*	**49**	*60*	**39**	*79*	**34**	*88*	**37**	*89*
1954	814260	**11**	*102*	**10**	*291*	**9**	*362*	**9**	*383*	**9**	*432*	**9**	*468*	**8**	*454*	**9**	*452*	**9**	*480*	**8**	*501*
1955	784224	**4**	*221*	**3**	*579*	**2**	*694*	**2**	*747*	**3**	*773*	**2**	*836*	**4**	*783*	**3**	*815*	**1**	*878*	**3**	*874*
1980	841641									**47**	*59*	**46**	*65*	**56**	*56*	**51**	*58*	**52**	*58*	**45**	*70*
1991	740554																	**61**	*52*		
2046	244618			**9**	*294*	**10**	*331*	**10**	*376*	**10**	*378*	**10**	*438*	**10**	*434*	**10**	*431*	**10**	*412*	**10**	*420*
2050	295985			**21**	*93*	**20**	*110*	**19**	*117*	**20**	*129*	**23**	*113*	**20**	*128*	**20**	*129*	**20**	*130*	**19**	*161*
2144	308231													**64**	*50*					**55**	*59*
2146	293500	**9**	*123*															**60**	*52*	**62**	*54*
2157	244637							**45**	*58*					**51**	*60*	**50**	*59*	**53**	*58*	**65**	*51*
2162	308163																	**67**	*50*	**69**	*50*
2186	208699															**61**	*51*			**59**	*56*
2199	135688					**43**	*55*			**36**	*76*	**39**	*72*	**55**	*57*	**41**	*74*	**38**	*80*	**49**	*64*

Continued on Next Page...

Table B.10 – *Continued*

(d) Threshold-based system																					
		Fitness Evaluation																			
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652	**25**	*50*	**30**	*81*	**26**	*114*	**27**	*118*	**26**	*130*	**26**	*128*	**31**	*112*	**27**	*134*	**26**	*143*	**25**	*146*
74	193913											**61**	*51*	**53**	*62*	**48**	*66*	**48**	*64*	**58**	*59*
85	297392	**24**	*50*	**32**	*74*	**25**	*114*	**24**	*126*	**28**	*120*	**30**	*108*	**28**	*123*	**28**	*129*	**31**	*115*	**30**	*124*
107	365826							**36**	*86*	**51**	*57*	**42**	*72*	**49**	*65*	**42**	*76*	**39**	*88*	**35**	*98*
123	236282			**38**	*55*			**47**	*60*	**46**	*62*	**52**	*56*	**48**	*65*	**46**	*68*	**56**	*57*	**51**	*66*
129	298062					**41**	*71*	**45**	*61*	**43**	*67*	**48**	*59*	**55**	*61*	**47**	*67*	**46**	*72*	**46**	*79*
153	383188	**13**	*107*	**12**	*212*	**12**	*259*	**11**	*284*	**11**	*313*	**11**	*321*	**11**	*314*	**11**	*322*	**11**	*332*	**12**	*315*
165	283315			**36**	*59*	**33**	*84*	**37**	*86*	**32**	*95*	**43**	*70*	**39**	*92*	**35**	*96*	**35**	*95*	**40**	*92*
187	296448	**11**	*123*					**8**	*411*	**8**	*385*	**8**	*421*	**8**	*430*	**8**	*463*	**8**	*432*	**8**	*504*
188	435953			**8**	*295*	**8**	*365*					**60**	*51*			**45**	*69*			**66**	*52*
236	878280					**34**	*79*	**42**	*67*	**44**	*66*	**46**	*61*	**47**	*68*	**44**	*75*	**43**	*54*	**44**	*88*
246	377461	**6**	*215*	**5**	*461*	**4**	*575*	**4**	*613*	**4**	*602*	**5**	*640*	**5**	*695*	**5**	*654*	**5**	*77*	**5**	*712*
251	486787									**56**	*50*			**65**	*50*						
255	325182	**12**	*118*	**11**	*235*	**13**	*236*	**14**	*238*	**12**	*270*	**13**	*290*	**12**	*287*	**12**	*294*	**14**	*682*	**13**	*303*
335	1469292	**23**	*50*	**21**	*112*	**29**	*105*	**28**	*112*	**20**	*159*	**24**	*142*	**24**	*149*	**25**	*148*	**24**	*286*	**24**	*151*
365	1434905									**50**	*58*	**53**	*56*			**53**	*62*	**54**	*151*	**57**	*59*
368	1473131									**52**	*53*							**60**	*58*		
380	289645																			**56**	*59*
417	395708			**39**	*53*	**39**	*71*	**38**	*81*	**37**	*87*	**37**	*85*	**35**	*98*	**43**	*75*	**41**	*81*	**37**	*95*
430	379708											**57**	*52*	**54**	*61*	**65**	*50*	**49**	*62*	**63**	*54*
509	207274	**2**	*391*	**2**	*562*	**3**	*622*	**3**	*688*	**3**	*720*	**3**	*718*	**3**	*731*	**4**	*742*	**3**	*788*	**3**	*779*
545	1435862	**3**	*260*	**4**	*487*	**6**	*539*	**6**	*556*	**6**	*576*	**6**	*598*	**6**	*603*	**6**	*652*	**6**	*642*	**6**	*652*
554	461425			**23**	*110*	**19**	*138*	**20**	*141*	**19**	*159*	**19**	*165*	**22**	*159*	**20**	*161*	**20**	*178*	**21**	*173*
585	68977											**62**	*51*					**64**	*51*		
603	42558															**59**	*58*				
742	812105	**1**	*477*	**1**	*800*	**1**	*942*	**1**	*951*	**1**	*981*	**1**	*999*	**1**	*1048*	**1**	*1027*	**1**	*1028*	**1**	*1085*
758	47475									**54**	*52*	**54**	*53*	**43**	*78*	**61**	*56*	**51**	*59*	**67**	*51*
783	767183					**36**	*75*	**34**	*92*	**39**	*83*	**33**	*95*	**34**	*105*	**34**	*102*	**36**	*94*	**34**	*105*
836	241412			**37**	*59*	**40**	*71*	**44**	*63*	**34**	*91*	**32**	*98*	**38**	*94*	**41**	*79*	**38**	*90*	**45**	*88*
842	810057			**34**	*70*	**37**	*73*	**33**	*93*	**33**	*93*	**34**	*93*	**37**	*94*	**38**	*91*	**34**	*103*	**32**	*111*
846	183337	**10**	*126*	**15**	*189*	**11**	*282*	**13**	*244*	**13**	*260*	**12**	*294*	**13**	*278*	**14**	*270*	**13**	*287*	**11**	*319*
910	839552					**43**	*62*	**50**	*56*	**47**	*61*	**45**	*66*	**42**	*78*	**58**	*58*	**55**	*57*	**47**	*77*
951	841620					**42**	*66*	**41**	*68*	**41**	*73*	**38**	*84*	**41**	*84*	**36**	*94*	**42**	*80*	**42**	*89*
976	786084			**22**	*112*	**23**	*130*	**23**	*131*	**24**	*134*	**22**	*144*	**25**	*147*	**24**	*149*	**27**	*142*	**22**	*167*
1003	796258	**14**	*95*	**14**	*194*	**14**	*227*	**15**	*231*	**15**	*245*	**15**	*240*	**14**	*275*	**15**	*252*	**15**	*252*	**15**	*246*
1055	1409509			**29**	*82*	**35**	*77*	**30**	*104*	**36**	*90*	**35**	*92*	**29**	*118*	**33**	*103*	**37**	*93*	**31**	*119*
1066	486110																	**62**	*51*	**70**	*50*
1084	878652	**22**	*50*	**26**	*93*	**31**	*92*	**31**	*103*	**29**	*116*	**27**	*127*	**32**	*111*	**30**	*117*	**29**	*128*	**33**	*110*
1116	626502			**27**	*88*	**27**	*108*	**25**	*125*	**31**	*103*	**29**	*120*	**26**	*145*	**31**	*113*	**30**	*123*	**29**	*128*
1158	814526			**25**	*97*	**20**	*138*	**19**	*159*	**21**	*147*	**23**	*142*	**19**	*176*	**19**	*162*	**22**	*163*	**18**	*206*
1159	142788																			**71**	*50*
1207	143306			**24**	*110*	**24**	*123*	**26**	*121*	**25**	*133*	**25**	*140*	**21**	*163*	**23**	*152*	**21**	*164*	**26**	*141*
1295	344134											**63**	*50*							**53**	*65*
1319	866702	**18**	*60*	**19**	*119*	**16**	*192*	**17**	*177*	**16**	*218*	**17**	*183*	**16**	*232*	**18**	*212*	**16**	*221*	**16**	*225*
1327	491565			**33**	*73*	**30**	*92*	**35**	*88*	**35**	*90*	**39**	*82*	**33**	*109*	**32**	*112*	**28**	*139*	**28**	*135*
1386	745019			**31**	*80*	**32**	*88*	**32**	*98*	**30**	*110*	**31**	*103*	**30**	*113*	**29**	*121*	**32**	*113*	**38**	*95*
1387	770394			**40**	*53*	**47**	*57*			**49**	*59*			**62**	*52*	**51**	*62*	**59**	*55*	**65**	*53*
1389	770394	**7**	*206*	**6**	*415*	**5**	*551*	**5**	*578*	**5**	*687*	**4**	*686*	**4**	*695*	**3**	*768*	**4**	*706*	**4**	*768*
1434	784257					**38**	*72*	**46**	*60*	**38**	*87*	**41**	*77*	**40**	*88*	**39**	*89*	**40**	*86*	**36**	*97*
1489	505491											**51**	*57*	**60**	*54*					**69**	*51*
1497	203003			**35**	*64*	**44**	*61*	**39**	*71*	**40**	*75*	**40**	*79*	**36**	*95*	**40**	*88*	**33**	*103*	**43**	*89*
1536	530185											**59**	*51*	**59**	*54*	**63**	*53*	**65**	*50*	**48**	*72*
1601	629896	**5**	*222*	**7**	*374*	**7**	*422*	**7**	*458*	**7**	*476*	**7**	*486*	**7**	*541*	**7**	*527*	**7**	*545*	**7**	*532*
1606	624360	**16**	*76*	**17**	*130*	**17**	*172*	**16**	*187*	**17**	*212*	**16**	*200*	**18**	*197*	**16**	*236*	**17**	*212*	**19**	*201*
1613	80338											**55**	*53*	**58**	*54*	**52**	*62*			**62**	*54*
1626	811000													**61**	*52*	**50**	*64*	**57**	*57*	**64**	*54*
1634	82903													**57**	*59*						
1645	52076	**15**	*85*	**13**	*208*	**15**	*220*	**12**	*255*	**14**	*246*	**14**	*254*	**15**	*268*	**13**	*280*	**12**	*291*	**14**	*270*
1662	377048							**52**	*54*			**56**	*53*	**63**	*52*	**60**	*58*	**61**	*52*	**61**	*54*
1700	796475															**66**	*50*				
1708	43733															**62**	*53*				
1738	771323					**45**	*59*	**40**	*70*	**45**	*65*	**36**	*86*	**44**	*75*	**37**	*92*	**44**	*77*	**41**	*91*
1764	44563							**53**	*53*			**50**	*58*	**51**	*63*	**57**	*58*	**63**	*51*	**55**	*61*
1776	768246											**47**	*61*	**56**	*60*	**64**	*52*	**53**	*59*	**52**	*66*
1884	609663	**21**	*54*	**28**	*85*	**28**	*107*	**29**	*106*	**27**	*124*	**28**	*126*	**27**	*125*	**26**	*144*	**25**	*144*	**27**	*140*
1911	898219							**48**	*59*	**53**	*53*	**49**	*59*	**46**	*71*			**50**	*60*	**49**	*69*
1915	840942							**43**	*63*	**55**	*51*			**52**	*62*	**49**	*65*	**52**	*59*	**50**	*68*
1916	80109	**17**	*63*	**16**	*137*	**22**	*131*	**22**	*134*	**23**	*141*	**21**	*153*	**23**	*156*	**22**	*154*	**23**	*161*	**23**	*157*
1932	782811	**20**	*58*	**18**	*126*	**18**	*146*	**18**	*175*	**18**	*173*	**18**	*175*	**17**	*210*	**17**	*215*	**18**	*207*	**17**	*209*
1954	814260	**9**	*131*	**9**	*262*	**9**	*310*	**9**	*337*	**9**	*352*	**9**	*354*	**9**	*390*	**9**	*384*	**9**	*370*	**9**	*413*
1955	784224	**4**	*250*	**3**	*549*	**2**	*649*	**2**	*765*	**2**	*749*	**2**	*754*	**2**	*795*	**2**	*836*	**2**	*845*	**2**	*828*
2022	204545																			**68**	*51*
2046	244618	**8**	*139*	**10**	*245*	**10**	*302*	**10**	*328*	**10**	*342*	**10**	*334*	**10**	*370*	**10**	*360*	**10**	*356*	**10**	*389*
2050	295985	**19**	*60*	**20**	*119*	**21**	*134*	**21**	*141*	**22**	*143*	**20**	*156*	**20**	*175*	**21**	*158*	**19**	*187*	**20**	*188*
2144	308231											**58**	*52*								
2157	244637					**46**	*58*	**49**	*57*	**42**	*69*	**44**	*70*	**45**	*72*	**55**	*60*	**47**	*69*	**39**	*92*
2162	308163																			**60**	*55*
2186	208699							**54**	*52*	**48**	*60*	**64**	*50*	**64**	*52*	**54**	*61*	**45**	*74*	**54**	*63*
2199	135688							**51**	*54*					**50**	*65*	**56**	*59*	**58**	*55*	**59**	*58*

Table B.11: The complete list of SRBCTs genes in the population size 200.

(a) Sigmoid-based system																			
		Fitness Evaluation																	
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652			23	94	25	121	23	138	24	139	20	159	22	157	27	139	26	143	27	138
85	297392			30	67	32	87	28	118	30	122	29	130	27	136	29	126	33	119	25	145
107	365826					42	67			46	67	41	76	43	81	38	93	37	95	42	87
123	236282			39	51	36	77	38	85	47	67	39	86	40	89	44	78	45	79	46	77
129	298062			25	81	24	122	25	123	23	140	22	156	26	142	24	144	25	145	23	158
139	729964																	62	56	65	52
153	383188			14	215	13	310	12	385	14	365	13	397	13	431	12	419	13	392	13	392
165	283315					51	54	44	69	42	74	46	66	44	80	46	72	41	83	45	77
166	897177													63	52						
174	769716									59	51	66	50								
187	296448	8	91	8	552	8	770	7	877	7	953	7	995	7	977	7	1006	7	1000	7	1040
188	435953					53	50			52	60			58	58			54	64	63	52
236	878280			34	56	35	78	33	100	39	79	32	109	37	94	36	101	34	119	35	102
246	377461	3	253	4	916	5	1164	5	1273	5	1274	5	1323			5	1317	5	1373	5	1396
255	325182	13	52	12	260	12	332	13	351	12	392	12	400	5	1358	13	394	12	428	12	429
257	740801											65	50								
326	809910													11	452	56	58				
335	1469292			22	96	19	151	27	122	21	152	25	139	23	157	22	151	24	148	22	173
365	1473131							48	61			50	64	65	51	64	52	56	64		
368	1473131							50	56			54	60	52	66	63	53	51	70	48	70
380	289645									60	50	47	65	47	73	49	62	60	57	52	62
407	195751															69	50				
417	395708					38	74	43	70	34	99	38	89	34	103	39	90	35	100	31	119
430	379708					41	72	46	66	44	70	42	74	50	67	43	79	49	74	47	73
509	207274	1	478	2	1198	2	1449	2	1559	2	1552	3	1526	2	1632	3	1589	3	1577	2	1662
545	1435862	5	245	6	830	6	1030	6	1141	6	1131	6	1111	6	1199	6	1184	6	1216	6	1246
554	461425			16	196	16	254	16	289	16	280	16	294	16	286	16	281	17	276	16	284
566	357031									57	53	61	54	64	51	57	57	55	64	56	58
585	68977											64	52			52	61				
714	245330											62	54			67	50				
742	812105	2	467	1	1396	1	1751	1	1757	1	1801	1	1778	1	1802	1	1768	1	1775	1	1781
783	767183			38	51	30	93	37	86	33	101	37	95	33	104	32	117	39	90	33	117
836	241412			28	74	29	95	31	104	32	103	33	109	36	99	33	107	32	120	28	137
842	810057					49	56	39	79	43	72	43	70	48	71	48	71	47	78	44	81
846	183337	11	55	13	243	14	284	14	297	13	366	14	366	14	362	14	389	14	387	14	387
847	265874									58	52										
910	839552							54	51											57	58
951	841620					50	55	53	51	51	61	51	63	49	70	55	58	44	79	61	55
976	786084			21	100	27	104	22	139	26	131	24	149	25	144	26	140	27	126	26	145
1003	796258			17	170	17	200	18	214	17	257	17	262	17	267	17	250	16	287	17	266
1055	1409509			29	71	33	81	32	101	31	111	31	115	29	128	37	99	29	125	29	130
1066	486110											58	56							68	51
1084	878652			19	112	22	140	21	160	27	129	21	156	21	169	20	161	21	163	21	179
1105	788107															66	52	63	54		
1116	626502					40	73	42	73	35	98	34	107	38	94	35	101	42	82	39	92
1158	814526			27	75	21	144	19	180	19	210	19	173	19	215	19	185	19	204	19	210
1207	143306			24	84	23	124	29	107	28	127	27	137	24	150	28	133	20	172	32	118
1263	324494											67	50							67	51
1301	346696							55	50					62	55	58	57	58	60	70	50
1319	866702			15	202	15	261	15	292	15	317	15	335	15	335	15	302	15	304	15	315
1327	491565			40	50	48	57	49	59	45	68	45	68	39	91	41	89	36	97	43	84
1386	745019			26	80	26	117	24	133	22	147	28	137	28	128	25	143	23	150	24	147
1387	770394					52	52	45	66	54	56	63	53	46	73	61	54	52	69	49	65
1389	770394	4	251	3	981	3	1339	3	1415	3	1502	2	1580	3	1572	2	1607	2	1635	3	1590
1434	784257			36	54	34	80	36	88	36	93	36	100	31	111	31	123	31	121	34	113
1489	505491											68	50					68	50		
1497	203003							47	63	49	64	56	58	61	55	51	62	59	59	53	61
1536	530185													59	55	62	54	57	62		
1601	629896	6	158	7	632	7	781	8	781	8	801	8	860	8	780	8	885	8	871	8	867
1606	624360			18	150	18	180	17	247	18	232	18	211	18	235	18	241	18	245	18	235
1613	80338			37	52	39	74	35	89	38	84	48	65	45	78	47	71	46	78	40	91
1645	52076	12	54	11	296	11	371	11	403	11	398	11	425	12	435	11	436	11	470	11	456
1662	377048											52	61	56	61	53	60	66	52	51	63
1700	796475					46	58			55	56	40	83	55	63	50	62	53	65	55	58
1708	43733			33	58	43	64	41	76	41	79	44	69	41	84	42	84	40	87	38	99
1738	771323							56	50	61	50	69	50	57	60			67	51	69	50
1764	44563			32	58	28	101	26	122	25	138	35	105	30	124	23	145	28	126	30	120
1776	768246											60	55	60	55	59	56	48	76	60	56
1884	609663			31	62	37	75	30	105	37	91	30	128	32	104	34	102	43	82	36	102
1911	898219									53	59	57	57					64	53	50	64
1915	840942															60	54	69	50	66	52
1916	80109			20	110	20	148	20	165	20	170	23	151	20	181	21	159	22	161	20	180
1932	782811					47	57	40	77	40	79	55	59	42	83	40	90	38	93	41	89
1954	814260	10	77	9	436	9	601	9	715	9	708	9	709	9	687	9	752	9	739	9	724
1955	784224	7	139	5	890	4	1210	4	1377	4	1414	4	1447	4	1398	4	1482	4	1523	4	1424
1980	841641															68	50				
1991	740554																	70	50	62	55
2046	244618	9	79	10	346	10	454	10	562	10	521	10	545	10	536	10	593	10	622	10	557
2050	295985			35	56	31	88	34	99	29	124	26	139	35	99	30	124	30	124	37	101
2144	308231																			59	57
2157	244637							51	53	50	63	59	55	51	66	45	74	61	56	54	60
2186	208699					44	62	52	52	48	66	53	61	53	63	65	52	65	53	64	52
2199	135688					45	59			56	55	49	64	54	63	54	60	50	72	58	57

Continued on Next Page...

Table B.11 – *Continued*

(b) Linear-based system																					
		Fitness Evaluation																			
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652			24	113	25	143	22	172	23	183	23	195	20	233	21	222	22	218	25	208
74	193913					20	170	43	67	47	68	50	59	51	63	53	67	57	60	52	63
85	297392			21	122			21	183	21	203	20	209	21	228	20	235	24	212	22	217
94	809603											61	50	47	73					65	56
107	365826																	62	55		
123	236282					49	51	54	52	42	82	46	69			46	79	50	68	48	79
153	383188	10	76	9	299	8	463			8	521	8	562	8	597	8	562	9	579	9	603
165	283315			32	75	35	100	8	525	32	124	35	119	34	126	35	126	34	132	34	142
166	897177							36	95					59	57	68	50	64	54		
187	296448	16	56	10	287	10	427	10	468	9	504	9	552	9	551	9	518	8	592	8	614
188	435953															62	51			60	59
236	878280			36	68	38	79	38	90	38	94	41	81	41	82	42	93	40	101	37	104
246	377461	6	129	6	514	6	677	6	697	6	756	6	789	6	807	7	808	6	886	6	829
251	486787															59	54	52	65	58	61
255	325182			16	207	16	303	16	315	16	342	15	360	16	365	16	353	15	369	15	386
276	868304															61	54	69	50	61	58
335	1469292			19	136	18	179	18	223	19	245	19	235	19	237	19	244	19	251	20	225
365	1434905					45	56					56	54	48	71	50	72	58	59	56	62
380	289645											60	50					53	64		
407	195751																			69	51
417	395708					39	76	40	80	39	89	37	92	39	83	38	106	45	79	39	94
509	207274	2	318	2	750	4	945	4	961	4	992	5	932	4	1029	4	1049	5	979	5	991
545	1435862	3	194	5	624	5	842	5	913	5	904	4	962	5	967	5	1012	4	1007	4	1049
554	461425			28	95	26	143	30	133	27	162	27	175	27	172	27	187	29	174	27	194
585	68977							52	54							64	51	68	51		
589	769657																			55	63
742	812105	1	385	1	1180	1	1567	1	1616	1	1700	1	1738	1	1855	1	1892	1	1895	1	1936
758	47475					48	52	47	60	44	77	43	77	43	80	41	101	38	102	38	96
783	767183			31	78	32	117	32	126	35	116	32	133	33	134	31	144	32	134	33	145
836	241412			27	100	29	132	29	139	29	158	30	156	30	164	30	160	30	156	29	179
842	810057			38	56	36	89	34	113	34	120	34	125	36	119	33	129	33	132	32	147
846	183337	8	98	8	320	9	452	9	494	10	480	10	508	10	516	10	501	10	520	10	545
910	839552									52	55	55	54	57	58	55	65	59	58	57	61
951	841620					46	54	46	63	50	64	40	84	38	84	39	106	41	98	42	87
970	824602													61	53	58	54			67	55
976	786084			30	89	30	127	27	147	26	166	29	165	31	160	32	142	28	175	26	195
1003	796258	13	61	14	255	15	328	13	372	14	377	14	369	14	390	12	455	14	405	14	421
1055	1409509			33	75	34	101	33	114	37	114	36	109	35	123	36	120	37	119	35	137
1067	489489									53	54	51	59	53	61	48	74	49	69	54	63
1084	878652			37	57	33	110	35	109	36	115	33	131	32	135	34	127	36	119	36	133
1116	626502			22	116	23	153	24	154	24	177	24	191	24	191	25	191	25	194	23	213
1158	814526			20	122	22	167	20	211	18	254	17	250	18	244	17	253	17	278	17	303
1159	142788													63	52			63	54	72	50
1203	144881															65	51	60	58		
1207	143306			23	114	21	170	19	219	22	201	21	202	22	212	23	200	20	246	19	234
1295	344134									55	53	49	59	58	57	56	62	70	50	66	55
1301	346696													28	168			65	54	70	51
1319	866702			26	101	24	149	26	147	30	152	28	174	37	113	29	164	31	155	28	186
1327	491565			35	68	37	82	37	94	33	123	38	90	29	165	37	116	35	124	43	84
1386	745019			34	73	31	124	31	128	31	144	31	147			28	175	27	176	30	174
1387	770394					40	71	45	66	40	88	45	73	45	75	40	103	43	85	45	84
1389	770394	4	152	4	689	3	1074	3	1180	2	1395	3	1338	2	1436	2	1475	2	1502	2	1567
1434	784257							44	66	45	73	39	86	46	75	45	80	39	102	40	90
1489	505491													56	58			66	54	73	50
1497	203003			39	51	42	62	39	81	43	81	42	79	44	78	47	77	42	94	41	88
1536	530185					44	56	42	74	48	66	48	60	52	63	49	74	55	62	51	64
1601	629896	5	141	7	499	7	623	7	691	7	751	7	758	7	749	6	810	7	767	7	804
1606	624360	9	91	15	252	14	331	15	341	15	373	16	356	15	375	15	360	16	359	16	379
1613	80338					51	50			56	53	53	57	60	56	60	54	54	63	53	63
1626	811000							55	52			59	52	49	66	52	68	51	68	59	60
1634	82903																	71	50	62	57
1645	52076			29	94	27	140	28	145	28	159	25	179	26	180	26	189	26	192	31	159
1662	377048													64	50	66	50				
1700	796475																			68	52
1738	771323			40	50	41	65	41	77	41	85	44	77	42	82	43	84	46	76	46	83
1776	768246																	61	56	71	50
1884	609663			18	140	19	174	23	171	20	208	22	202	23	207	22	220	21	233	24	210
1911	898219					43	57	49	59	57	51	54	56	54	61	63	51	67	53	50	68
1915	840942					47	54	50	57	51	59	58	52	55	60	51	70	48	70	49	68
1916	80109	14	58	17	167	17	234	17	226	17	280	18	249	17	248	18	245	18	274	18	270
1932	782811	11	66	11	265	13	332	14	365	12	411	13	390	13	410	14	386	12	456	13	421
1954	814260	15	57	12	263	11	361	11	381	13	399	12	397	12	429	13	423	13	444	12	437
1955	784224	7	124	3	745	2	1077	2	1233	3	1337	2	1356	3	1425	3	1469	3	1471	3	1535
2046	244618	12	64	13	263	12	344	12	374	11	420	11	431	11	430	11	459	11	478	11	466
2050	295985			25	102	28	140	25	152	25	168	26	178	25	186	24	192	23	215	21	221
2157	244637							48	60	46	70	47	66	40	82	44	82	44	82	44	84
2186	208699					50	50	51	56	49	65	52	58	50	64	54	67	47	74	47	81
2199	135688							53	52	54	53	57	53	62	52	57	61	56	60	64	56
2230	417226															67	50				
2240	809603																	72	50	63	57

Continued on Next Page...

Table B.11 – *Continued*

(c) Tanh-based system																			
		Fitness Evaluation																	
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652			29	55	31	85	28	102	28	110	32	110	35	92	30	111	31	120	34	105
85	297392					36	68	36	75	35	85	28	112	36	92	27	120	39	87	32	112
107	365826					41	57	42	67	40	75	40	80	39	81	40	84	40	78	36	96
123	236282					39	63	41	68	42	74	39	81	41	77	37	88	45	70	44	79
129	298062			19	84	18	144	19	145	19	156	21	161	20	162	19	168	21	179	19	187
139	729964							55	50									69	50		
153	383188			15	153	15	236	14	263	15	286	14	322	14	340	14	334	14	336	14	353
165	283315													60	52			64	52	60	54
174	769716																			65	52
187	296448			8	413	7	694	7	814	7	929	7	968	6	996	7	944	7	1002	7	942
188	435953									50	57			62	51					59	54
236	878280					33	79	30	93	31	107	27	113	25	127	34	105	29	125	33	106
246	377461	3	166	3	798	3	1125	4	1176	5	1216	5	1220	4	1309	5	1254	5	1339	5	1333
255	325182			12	226	12	342	11	362	12	378	13	335	12	411	12	382	12	401	12	390
335	1469292					24	106	27	104	29	109	24	123	24	128	25	128	27	133	23	140
365	1434905							50	58	57	52			40	78	52	59	44	74	55	59
368	1473131					45	52	39	71	44	71	47	69	51	59	44	76	48	68	48	68
380	289645									54	53							53	60	57	56
417	395708					37	66	34	84	32	104	33	109	33	102	33	109	34	106	30	115
430	379708			28	56			32	87	37	79	41	80	31	103	38	87	36	96	37	93
509	207274	1	322	2	1024	2	1352	2	1378	2	1454	2	1452	2	1494	2	1508	2	1474	1	1603
545	1435862	5	146	6	600	6	825	6	857	6	1014	6	973	7	996	6	993	6	1030	6	1074
554	461425			14	180	14	253	15	263	13	325	16	280	15	308	15	323	16	295	16	277
566	357031					35	71	37	72	41	74	46	72	38	82	41	80	43	74	38	86
575	823886											55	51								
585	68977													61	52	56	55				
714	245330			1	1098									56	56					58	55
742	812105	2	299			1	1399	1	1447	1	1500	1	1526	1	1580	1	1580	1	1577	2	1526
783	767183							40	68	43	72	44	73	42	73	45	72	46	70	39	85
836	241412					34	77	35	76	38	77	43	76	43	72	32	110	35	97	42	80
842	810057					43	57	52	57	49	60	45	73	48	67	51	62	55	60	40	84
846	183337			16	144	16	214	16	226	16	277	15	307	16	269	16	320	15	304	15	282
847	265874													57	55						
910	839552																	56	59		
951	841620									55	52	56	50			46	71	51	60	56	58
976	786084			23	65	26	94	33	85	24	126	26	119	28	123	31	111	24	136	22	140
1003	796258			17	138	17	165	17	192	17	198	17	253	17	242	17	232	17	258	18	217
1055	1409509			25	65	29	86	38	72	34	97	35	95	34	96	35	105	33	114	27	121
1066	486110																	63	53		
1084	878652			22	72	19	144	22	121	21	154	19	195	19	164	23	144	20	181	20	169
1105	788107															57	54	61	54	62	53
1116	626502							44	62	60	50	51	54	45	70	60	52	58	57	45	74
1158	814526					23	106	26	106	22	146	22	141	22	140	21	151	22	160	24	138
1207	143306					25	97	29	102	27	118	31	110	30	103	28	119	25	135	29	118
1263	324494													58	54			67	50		
1301	346696											54	53	59	52	55	56	59	57	63	53
1319	866702			13	202	13	287	13	321	14	324	12	351	13	356	13	355	13	353	13	358
1327	491565					42	57	46	62	45	68	38	84	44	71	53	58	42	75	50	64
1386	745019			27	58	32	81	21	134	30	108	34	100	27	124	36	93	32	115	26	123
1387	770394																	60	56	51	62
1389	770394	4	161	4	718	4	1091	3	1234	3	1323	3	1409	3	1398	3	1358	3	1444	3	1442
1434	784257					28	88	31	89	33	97	29	112	32	103	29	111	28	128	35	98
1536	530185																			66	51
1601	629896	6	108	7	524	8	684	8	775	8	739	8	781	8	781	8	810	8	834	8	821
1606	624360			18	86	22	110	18	163	20	155	20	163	21	155	20	161	19	181	21	157
1613	80338			30	50	40	60	49	59	36	83	37	87	37	90	42	80	37	95	43	80
1645	52076			11	240	10	387	12	355	11	396	11	409	11	424	11	424	11	445	11	453
1662	377048					44	54	48	60	51	56	49	62	53	58	49	63	49	66	53	59
1700	796475									47	63			50	60	59	53	52	60	52	60
1708	43733			26	58	27	91	24	113	26	119	23	133	26	125	24	139	26	133	31	113
1764	44563			21	75	20	137	20	143	18	188	18	207	18	206	18	200	18	197	17	235
1776	768246							45	62	53	55	48	64	54	56	48	64	41	77	49	65
1884	609663					38	64	47	61	39	76	42	78	47	67	43	77	38	94	41	83
1911	898219											53	53			50	62	65	51		
1915	840942									56	52										
1916	80109			24	65	21	113	23	113	23	133	25	121	23	135	22	148	23	138	25	132
1932	782811									52	56			55	56			68	50	64	52
1954	814260			9	357	9	562	9	601	9	680	9	694	9	705	9	789	9	702	9	709
1955	784224	7	79	5	688	5	1049	5	1175	4	1224	4	1231	5	1250	4	1334	4	1378	4	1346
1980	841641									58	51	52	54			47	65	66	50	47	70
1991	740554															58	53	62	53	67	50
2046	244618			10	274	11	373	10	471	10	476	10	523	10	565	10	535	10	583	10	532
2050	295985			20	76	30	85	25	112	25	125	30	110	29	107	26	128	30	120	28	118
2144	308231							51	58					49	66	54	56	54	60	54	59
2157	244637							53	50	48	61	50	58	52	58			50	62	61	53
2186	208699							54	50	59	51							57	58		
2199	135688					46	51	43	63	46	63	36	89	46	68	39	84	47	68	46	72

Continued on Next Page...

Table B.11 – *Continued*

(d) Threshold-based system																					
		Fitness Evaluation																			
Gene	Image	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652			**25**	*109*	**22**	*166*	**22**	*191*	**22**	*191*	**26**	*166*	**21**	*210*	**23**	*210*	**24**	*197*	**25**	*207*
74	193913							**55**	*51*	**54**	*55*	**57**	*57*	**61**	*58*	**61**	*58*	**61**	*59*	**62**	*60*
85	297392			**23**	*112*	**26**	*148*	**24**	*183*	**24**	*187*	**25**	*170*	**24**	*200*	**24**	*200*	**23**	*204*	**22**	*220*
94	365826														*52*	**69**	*52*				
107	236282									**46**	*67*	**50**	*65*	**52**	*68*	**49**	*68*	**58**	*60*	**45**	*80*
123	298062					**38**	*80*	**41**	*80*	**47**	*67*	**44**	*75*	**46**	*99*	**39**	*99*	**40**	*91*	**40**	*93*
129	729964					**51**	*51*	**58**	*50*			**58**	*57*	**66**	*52*	**67**	*52*	**57**	*61*	**70**	*52*
153	383188	**11**	*57*	**12**	*273*	**9**	*433*	**11**	*425*	**11**	*471*	**10**	*482*	**9**	*474*	**12**	*474*	**10**	*530*	**10**	*520*
165	283315			**35**	*61*	**37**	*84*	**34**	*116*	**37**	*97*	**35**	*107*	**34**	*123*	**35**	*123*	**34**	*117*	**35**	*131*
166	897177													**62**	*54*	**65**	*54*			**74**	*51*
187	296448	**9**	*66*	**8**	*419*	**8**	*592*	**8**	*633*	**8**	*656*	**8**	*772*	**8**	*797*	**7**	*797*	**7**	*825*	**8**	*808*
188	435953							**53**	*54*			**61**	*52*	**55**	*61*	**56**	*61*	**52**	*64*	**50**	*69*
236	878280			**37**	*57*	**39**	*72*	**39**	*88*	**41**	*81*	**34**	*107*	**39**	*109*	**36**	*109*	**38**	*102*	**39**	*95*
246	377461	**5**	*157*	**6**	*645*	**6**	*834*	**6**	*914*	**6**	*1001*	**6**	*980*	**6**	*1007*	**6**	*1007*	**6**	*1036*	**6**	*1075*
251	486787									**61**	*50*				*61*	**55**	*61*				
255	325182			**14**	*216*	**13**	*346*	**13**	*348*	**13**	*397*	**13**	*386*	**13**	*402*	**13**	*402*	**13**	*395*	**13**	*398*
276	868304			**40**	*53*	**49**	*55*	**57**	*50*	**51**	*59*	**52**	*64*	**67**	*56*	**64**	*56*	**48**	*72*	**69**	*52*
335	1469292			**19**	*150*	**20**	*183*	**21**	*192*	**21**	*194*	**21**	*202*	**22**	*232*	**19**	*232*	**20**	*222*	**24**	*210*
365	1434905											**63**	*52*	**51**	*50*	**73**	*50*	**65**	*57*	**56**	*64*
368	1473131														*56*	**63**	*56*				
380	289645									**55**	*54*	**54**	*59*							**58**	*61*
407	195751																			**75**	*50*
417	395708					**41**	*67*	**40**	*83*	**36**	*98*	**40**	*89*	**42**	*81*	**44**	*81*	**39**	*94*	**41**	*92*
430	379708									**59**	*51*	**65**	*51*	**63**						**71**	*52*
509	207274	**2**	*400*	**2**	*950*	**4**	*1098*	**4**	*1207*	**4**	*1232*	**4**	*1202*	**4**	*1212*	**4**	*1212*	**4**	*1279*	**4**	*1242*
545	1435862	**3**	*222*	**5**	*716*	**5**	*925*	**5**	*1012*	**5**	*1054*	**5**	*1059*	**5**	*1091*	**5**	*1091*	**5**	*1132*	**5**	*1120*
554	461425			**22**	*115*	**23**	*163*	**18**	*217*	**20**	*198*	**22**	*193*	**23**	*214*	**22**	*214*	**22**	*208*	**23**	*215*
585	68977							**54**	*52*	**56**	*54*			**64**	*50*	**72**	*50*	**53**	*64*	**65**	*57*
589	769657									**60**	*51*				*52*	**70**	*52*	**64**	*58*	**61**	*61*
742	812105	**1**	*443*	**1**	*1290*	**1**	*1572*	**1**	*1741*	**1**	*1736*	**1**	*1827*	**1**	*1881*	**1**	*1881*	**1**	*1777*	**1**	*1942*
758	47475							**43**	*71*	**39**	*84*	**39**	*97*	**37**	*87*	**42**	*87*	**41**	*87*	**44**	*84*
783	767183			**33**	*70*	**32**	*111*	**36**	*106*	**33**	*120*	**33**	*113*	**33**	*134*	**33**	*134*	**33**	*130*	**34**	*137*
800	471266																	**68**	*54*		
836	241412			**27**	*97*	**31**	*130*	**31**	*132*	**32**	*132*	**28**	*152*	**31**	*144*	**32**	*144*	**32**	*141*	**31**	*167*
842	810057			**39**	*54*	**33**	*100*	**35**	*108*	**34**	*111*	**36**	*106*	**35**	*128*	**34**	*128*	**35**	*117*	**36**	*125*
846	183337	**8**	*85*	**10**	*296*	**12**	*374*	**10**	*445*	**10**	*486*	**12**	*446*	**11**	*515*	**10**	*515*	**11**	*483*	**12**	*466*
910	839552													**59**	*63*	**54**	*63*	**60**	*59*	**67**	*54*
951	841620							**50**	*59*	**38**	*92*	**46**	*70*	**40**	*74*	**47**	*74*	**42**	*86*	**48**	*77*
970	824602																	**70**	*53*		
976	786084			**26**	*104*	**30**	*139*	**25**	*173*	**27**	*171*	**29**	*151*	**30**	*185*	**26**	*185*	**26**	*181*	**26**	*181*
1003	796258			**15**	*193*	**15**	*261*	**16**	*240*	**15**	*303*	**16**	*287*	**16**	*316*	**15**	*316*	**16**	*320*	**15**	*324*
1055	1409509			**38**	*57*	**36**	*90*	**33**	*118*	**35**	*110*	**38**	*100*	**36**	*107*	**38**	*107*	**36**	*117*	**32**	*158*
1066	486110											**64**	*52*		*59*	**60**	*59*			**63**	*59*
1067	489489							**48**	*61*			**56**	*57*	**60**	*67*	**51**	*67*	**69**	*53*	**76**	*50*
1084	878652			**30**	*94*	**34**	*96*	**32**	*124*	**31**	*132*	**32**	*140*	**29**	*157*	**30**	*157*	**31**	*149*	**30**	*169*
1116	626502			**28**	*96*	**24**	*151*	**27**	*158*	**26**	*175*	**27**	*164*	**25**	*180*	**28**	*180*	**25**	*197*	**27**	*179*
1158	814526			**21**	*117*	**21**	*175*	**23**	*186*	**19**	*211*	**18**	*250*	**19**	*271*	**18**	*271*	**19**	*227*	**18**	*253*
1159	142788													**57**	*50*	**76**	*50*			**68**	*53*
1207	143306			**29**	*95*	**29**	*141*	**26**	*165*	**29**	*154*	**23**	*181*	**26**	*183*	**27**	*183*	**27**	*180*	**28**	*178*
1263	324494														*52*	**71**	*52*				
1295	344134									**48**	*66*	**60**	*53*	**68**	*71*	**48**	*71*	**62**	*58*	**54**	*64*
1301	346696											**67**	*50*	**50**				**67**	*56*	**72**	*52*
1319	866702			**20**	*122*	**19**	*191*	**20**	*202*	**23**	*191*	**20**	*208*	**20**	*216*	**21**	*216*	**21**	*209*	**21**	*222*
1327	491565			**34**	*61*	**35**	*92*	**37**	*93*	**40**	*83*	**37**	*103*	**38**	*108*	**37**	*108*	**37**	*112*	**37**	*124*
1386	745019			**31**	*91*	**27**	*142*	**30**	*147*	**30**	*153*	**31**	*142*	**27**	*169*	**29**	*169*	**29**	*173*	**29**	*170*
1387	770394			**36**	*59*	**43**	*64*	**42**	*73*	**42**	*78*	**41**	*82*	**43**	*65*	**52**	*65*	**46**	*74*	**46**	*79*
1389	770394	**4**	*158*	**4**	*790*	**3**	*1098*	**3**	*1300*	**3**	*1343*	**2**	*1394*	**3**	*1532*	**2**	*1532*	**3**	*1447*	**2**	*1510*
1434	784257					**45**	*59*	**45**	*68*	**45**	*69*	**49**	*68*	**49**	*97*	**40**	*97*	**44**	*76*	**43**	*88*
1489	505491											**66**	*50*		*56*	**62**	*56*	**71**	*50*		
1497	203003					**40**	*71*	**46**	*67*	**44**	*76*	**42**	*78*	**41**	*75*	**46**	*75*	**47**	*72*	**38**	*99*
1536	530185					**46**	*56*	**49**	*60*	**58**	*51*	**43**	*76*	**58**	*67*	**50**	*67*	**63**	*58*	**55**	*64*
1601	629896	**7**	*125*	**7**	*520*	**7**	*645*	**7**	*711*	**7**	*775*	**7**	*798*	**7**	*770*	**8**	*770*	**8**	*797*	**7**	*812*
1606	624360	**10**	*60*	**13**	*229*	**14**	*291*	**14**	*284*	**14**	*354*	**14**	*336*	**14**	*378*	**14**	*378*	**14**	*334*	**14**	*352*
1613	80338					**42**	*65*	**47**	*64*	**53**	*58*	**51**	*65*	**48**	*53*	**66**	*53*	**56**	*62*	**47**	*78*
1626	811000									**62**	*50*	**59**	*54*		*60*	**59**	*60*	**51**	*66*		
1645	52076			**17**	*166*	**17**	*213*	**15**	*267*	**17**	*258*	**17**	*284*	**17**	*281*	**16**	*281*	**15**	*324*	**17**	*280*
1662	377048											**55**	*57*	**69**	*50*	**75**	*50*	**55**	*63*	**57**	*63*
1700	796475							**56**	*51*					**56**	*60*	**58**	*60*	**54**	*63*	**59**	*61*
1738	771323					**48**	*55*					**48**	*69*	**54**	*63*	**53**	*63*	**45**	*76*	**51**	*69*
1764	44563													**65**				**59**	*59*	**53**	*65*
1776	768246							**52**	*55*	**43**	*78*				*50*	**74**	*50*			**66**	*57*
1884	609663			**24**	*111*	**25**	*150*	**28**	*158*	**25**	*187*	**24**	*172*	**28**	*194*	**25**	*194*	**30**	*152*	**20**	*226*
1911	898219					**50**	*53*											**66**	*56*	**60**	*61*
1914	824704									**50**	*59*									**73**	*52*
1915	840942					**44**	*61*	**38**	*89*	**49**	*63*	**47**	*70*	**44**	*93*	**41**	*93*	**50**	*67*	**42**	*88*
1916	80109			**18**	*152*	**18**	*193*	**19**	*213*	**18**	*214*	**19**	*242*	**18**	*231*	**20**	*231*	**18**	*261*	**19**	*232*
1932	782811			**16**	*167*	**16**	*235*	**17**	*238*	**16**	*279*	**15**	*306*	**15**	*276*	**17**	*276*	**17**	*309*	**16**	*303*
1954	814260	**12**	*53*	**9**	*302*	**11**	*375*	**9**	*454*	**9**	*492*	**9**	*489*	**10**	*549*	**9**	*549*	**9**	*550*	**9**	*565*
1955	784224	**6**	*142*	**3**	*791*	**2**	*1132*	**2**	*1310*	**2**	*1357*	**3**	*1345*	**2**	*1522*	**3**	*1522*	**2**	*1470*	**3**	*1503*
2046	244618			**11**	*277*	**10**	*408*	**12**	*423*	**12**	*442*	**11**	*463*	**12**	*486*	**11**	*486*	**12**	*477*	**11**	*491*
2050	295985			**32**	*72*	**28**	*141*	**29**	*157*	**28**	*165*	**30**	*146*	**32**	*154*	**31**	*154*	**28**	*176*	**33**	*151*
2157	244637							**44**	*69*	**52**	*58*	**53**	*59*	**47**	*82*	**43**	*82*	**49**	*69*	**52**	*66*
2186	208699					**47**	*56*	**51**	*56*	**57**	*52*	**45**	*74*	**45**	*78*	**45**	*78*	**43**	*77*	**49**	*72*
2199	135688											**62**	*52*	**53**	*60*	**57**	*60*			**64**	*58*
2240	809603														*52*	**68**	*52*				

Table B.12: The complete list of SRBCTs genes in the population size 300.

(a) Sigmoid-based system																					
		Fitness Evaluation																			
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652			24	56	24	100	24	133	25	132	24	149	25	148	24	144	22	156	22	163
74	193913																	60	51		
85	297392					29	83	31	92	33	92	30	105	28	122	27	126	25	140	28	116
107	365826									51	55			51	55	53	54	48	60	40	75
123	236282					36	63	35	74	39	68	40	74	34	97	37	91	39	82	33	96
129	298062					27	86	28	106	27	121	26	130	27	124	25	133	29	116	27	123
139	729964									58	51									59	52
153	383188			13	180	14	302	14	361	13	406	12	433	11	434	11	448	12	488	11	469
165	283315							41	63	42	64	47	57	41	82	40	78	41	73	41	73
187	296448			8	357	8	797	7	1021	7	1064	7	1115	7	1198	7	1142	7	1219	7	1194
188	435953							52	51	53	54	48	57	52	55	48	59	55	56	54	58
236	878280					30	75	29	98	32	92	32	99	37	88	29	118	30	110	32	97
246	377461	5	77	4	801	5	1326	5	1520	5	1575	5	1632	5	1572	5	1602	5	1664	5	1624
255	325182			14	170	12	329	13	373	12	422	13	415	14	411	14	417	13	429	13	449
326	809910																			55	56
335	1469292			22	69	22	116	23	147	22	155	20	176	19	194	23	150	24	145	21	163
365	1473131											55	51	61	50	47	59	52	57	53	59
368	1473131							47	54	56	52	41	65					54	56		
380	289645							50	51					54	54	46	59			49	61
407	195751											56	50								
417	395708					38	60	34	77	40	67	31	104	33	98	36	95	38	83	34	93
430	379708									45	62	50	55	46	65	56	53	61	50	51	60
509	207274	1	193	2	1338	2	1784	2	1900	2	1968	2	1991	3	1979	3	1989	3	2019	3	2006
545	1435862	4	91	5	800	6	1205	6	1362	6	1423	6	1500	6	1485	6	1506	6	1480	6	1507
554	461425			16	140	15	249	15	275	16	246	16	310	15	292	16	311	16	299	15	300
566	357031									46	60										
714	245330																	59	52		
742	812105	2	191	1	1436	1	2102	1	2320	1	2305	1	2365	1	2449	1	2366	1	2408	1	2354
783	767183					40	57	37	71	29	98	28	110	35	93	33	101	37	86	35	91
836	241412			25	51	28	85	27	107	34	91	33	98	26	124	30	112	31	108	30	110
842	810057							49	52	43	64	51	54	63	50	51	57	46	64	46	66
846	183337			11	204	13	320	12	402	14	393	11	439	13	420	12	440	11	493	14	444
847	265874													60	50					52	60
951	841620									50	55					58	51			42	72
976	786084					26	92	26	118	26	125	29	108	29	113	28	124	27	129	25	138
1003	796258			18	103	18	194	18	207	18	220	18	226	18	215	18	243	17	245	17	269
1055	1409509					31	75	30	93	31	96	38	83	30	112	34	101	32	101	29	114
1084	878652			20	84	19	148	21	153	20	165	22	158	22	160	22	168	19	192	23	162
1105	788107											54	51	62	50						
1116	626502					33	70	42	59	36	89	39	79	40	84	41	67	36	93	38	82
1158	814526			23	59	23	115	20	168	21	162	19	179	20	184	19	205	20	190	19	195
1207	143306					25	96	25	121	23	151	25	147	24	149	26	131	26	135	26	135
1301	346696													49	58	54	54	45	64		
1319	866702			15	146	16	242	16	266	15	288	15	341	16	287	15	313	15	333	16	299
1327	491565							45	55	49	56	53	52	44	67	42	65	44	69	48	64
1386	745019			21	78	21	121	22	151	24	144	23	153	23	156	20	172	23	151	20	176
1387	770394					41	51	46	55	41	66	42	63	45	66	43	64	49	60	47	64
1389	770394	3	107	3	993	3	1560	3	1860	3	1915	3	1979	2	2073	2	2113	2	2138	2	2137
1434	784257					35	63	33	80	38	87	35	88	32	99	35	97	40	81	37	86
1497	203003									48	58	45	60					51	57	58	53
1536	530185									54	54			57	52	57	52				
1601	629896			7	515	7	806	8	932	8	960	8	986	8	987	8	982	8	974	8	1015
1606	624360			17	122	17	202	17	217	17	240	17	273	17	264	17	264	18	222	18	256
1613	80338					34	68	36	72	35	91	36	87	39	86	32	102	34	95	45	67
1634	82903																				
1645	52076			12	199	11	358	11	419	11	424	14	406	12	434	13	425	14	426	12	452
1662	377048							53	50			49	56	59	50	52	56	50	58		
1700	796475									44	63	46	60	47	62	59	50	53	57	44	69
1708	43733							44	56	52	54	44	60	43	71	49	58			43	72
1738	771323									57	52			58	51						
1764	44563					32	74	32	89	30	97	27	121	31	104	31	104	28	119	31	107
1776	768246													56	53			47	62	57	54
1884	609663					37	62	38	68	28	102	34	90	36	92	39	83	35	94	36	87
1915	840942															50	57				
1916	80109			19	85	20	124	19	171	19	173	21	163	21	173	21	169	21	164	24	158
1932	782811							43	58	55	52	52	53	53	54	44	62	42	71	50	60
1954	814260			9	313	9	581	9	731	9	772	9	770	9	807	9	817	9	820	9	848
1955	784224			6	724	4	1347	4	1546	4	1619	4	1700	4	1766	4	1802	4	1735	4	1742
1991	740554													55	53			56	56		
2046	244618			10	225	10	445	10	503	10	523	10	573	10	603	10	555	10	574	10	608
2050	295985					39	59	40	65	37	89	37	85	38	86	38	87	33	96	39	80
2157	244637							48	53					48	62	55	53	58	52	56	54
2186	208699							39	66	47	60	43	62	42	79	45	60	43	70		
2199	135688							51	51					50	57			57	53	60	51

Continued on Next Page...

Table B.12 – *Continued*

(b) Linear-based system																					
		Fitness Evaluation																			
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652			25	77	26	142	24	188	24	210	21	247	23	210	23	245	23	261	21	255
74	193913					45	56	42	64	54	54	49	62	57	54	48	70	52	66	58	55
85	297392			20	102	23	163	21	218	21	241	23	237	19	288	20	261	20	292	24	246
94	809603											53	55					63	56		
107	365826													59	51						
123	236282					40	59	41	68	41	72	41	87	46	70	41	87	38	99	56	60
153	383188			10	234	12	382	9	544	9	586	10	612	9	639	9	664	8	700	8	708
165	283315			33	55	32	93	32	115	32	138	33	115	32	146	31	155	30	154	32	157
166	897177															56	56	62	56	57	56
187	296448			13	190	10	391	10	472	10	516	9	623	10	619	10	625	10	665	9	679
188	435953											51	56					53	65	54	61
236	878280					38	68	44	63	42	68	39	90	43	75	39	95	44	84	39	107
246	377461			6	385	6	683	6	770	6	822	6	884	6	889	6	997	6	941	6	937
251	486787									45	62							68	53		
255	325182			19	116	16	241	15	347	16	311	16	345	16	362	16	379	16	357	16	357
276	868304					46	52					48	62								
335	1469292			17	121	18	198	17	279	20	252	18	265	20	269	18	309	19	301	19	299
365	1434905							51	52			55	53	53	56	54	59	47	72	62	54
380	289645													55	55					59	54
407	195751																			60	54
417	395708							50	54	40	79	57	51	40	81	43	85	45	82	44	80
509	207274	2	157	2	823	4	1114	4	1203	4	1156	4	1233	4	1230	4	1300	5	1259	4	1293
545	1435862	4	65	5	583	5	941	5	1025	5	1124	5	1142	5	1170	5	1157	4	1263	5	1268
554	461425			24	82	22	174	25	182	23	215	22	241	25	209	24	245	22	263	25	246
585	68977									56	50	56	53	54	55	59	51	54	65		
589	769657									52	57							67	53		
742	812105	1	172	1	1184	1	1939	1	2220	1	2342	1	2449	1	2424	1	2606	1	2561	1	2660
758	47475					44	56	39	73	38	98	34	112	37	118	36	122	34	138	35	119
783	767183			30	57	33	92	29	138	29	159	31	142	31	152	29	169	33	140	31	165
836	241412			23	84	25	152	26	173	28	185	27	202	29	171	26	209	28	196	27	202
842	810057					37	74	37	83	35	111	35	112	34	130	33	137	32	145	37	114
846	183337			8	319	8	523	8	564	8	607	8	658	8	671	8	710	9	679	10	660
910	839552																	64	55	64	50
937	789204															55	58				
951	841620							45	62	48	59	45	67	48	66	46	73	51	66	40	100
976	786084			31	56	30	100	28	141	31	146	30	145	30	156	30	160	29	163	33	140
1003	796258			9	240	13	376	13	418	13	450	14	445	14	454	14	473	12	529	14	473
1055	1409509			32	56	35	80	34	107	34	115	38	91	33	132	34	127	36	128	34	121
1066	486110													58	53						
1067	489489					39	63	49	55	51	57	43	75	42	77	47	73	50	68	52	63
1084	878652					34	83	36	88	33	115	36	104	36	120	38	100	39	97	36	118
1116	626502			21	102	24	163	23	201	25	204	25	217	26	203	25	243	25	229	26	237
1158	814526			22	89	19	193	19	246	19	256	17	279	17	308	17	338	17	316	17	327
1159	142788																			55	60
1203	144881																			66	50
1207	143306			27	72	20	187	20	226	18	263	20	255	21	248	21	260	21	267	22	255
1263	324494																	61	56		
1295	344134							53	50							58	55	48	69	53	61
1301	346696											58	50							65	50
1319	866702			26	74	31	98	33	113	36	111	32	136	35	128	35	124	31	152	30	171
1327	491565					43	58	40	68	37	109	37	92	41	79	37	106	40	95	38	113
1386	745019			29	58	27	131	27	172	27	186	26	217	24	209	27	198	26	223	20	258
1387	770394	3	83	3	657	36	75	35	89	39	86	40	89	38	94	40	93	37	101	41	90
1389	770394					3	1303	2	1637	2	1737	2	1898	2	1879	2	2044	2	2076	2	2109
1434	784257							48	56	46	60	52	55	52	59	42	87	59	60	47	70
1497	203003					42	58	38	76	47	60	42	76	39	85	44	83	41	90	42	86
1536	530185							47	57	50	59	47	64	51	60	49	68	42	90	45	74
1601	629896			7	385	7	661	7	707	7	818	7	857	7	848	7	842	7	913	7	879
1606	624360			11	231	11	384	12	440	12	483	12	477	13	485	12	514	14	507	12	549
1613	80338							46	60	49	59			49	62	50	62	55	64	46	70
1626	811000							54	50			59	50	56	55	57	56	49	69	51	64
1634	82903																	56	61		
1645	52076			28	66	28	117	30	134	30	147	29	145	27	173	28	197	35	136	29	175
1738	771323					41	59	43	64	43	66	44	73	45	72	45	80	43	88	43	82
1884	609663			18	119	21	176	22	212	22	217	24	237	22	241	22	249	24	253	23	249
1910	628336																			63	52
1911	898219																	60	50	65	54	61	54
1915	840942									55	54	50	58	50	62	51	62	66	53	49	67
1916	80109			15	162	17	224	18	278	17	270	19	256	18	290	19	295	18	305	18	321
1932	782811			12	193	9	397	11	441	11	494	11	519	11	512	11	569	11	538	11	553
1954	814260			16	155	15	280	16	330	15	374	15	441	15	401	15	442	15	433	15	411
1955	784224	5	60	4	613	2	1314	3	1554	3	1671	3	1793	3	1811	3	1894	3	1949	3	1990
2046	244618			14	170	14	282	14	390	14	441	13	461	12	500	13	511	13	521	13	523
2050	295985					29	114	31	126	26	187	28	164	28	171	32	150	27	209	28	199
2157	244637									53	54	54	53	47	68	53	59	57	60	50	65
2186	208699							52	51	44	64	46	65	44	73	52	61	46	78	48	69
2199	135688																	58	60		
2240	809603																	60	59		

Continued on Next Page...

Table B.12 – *Continued*

(c) Tanh-based system																					
		Fitness Evaluation																			
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652					28	69	25	103	24	117	30	93	30	101	38	88	29	112	31	110
85	297392					36	50	26	93	32	94	36	82	34	98	36	91	36	86	35	93
107	365826									48	52	41	64	50	56	43	73	48	60	40	73
123	236282							37	61	41	68	40	67	36	87	39	82	34	96	37	78
129	298062					21	89	20	133	21	138	22	123	22	137	25	121	22	140	28	120
139	729964													45	62						
153	383188			15	105	14	220	12	338	14	309	13	371	13	364	13	395	13	399	13	372
165	283315															49	55	50	59		
187	296448			8	274	8	715	7	934	7	1058	7	1141	7	1254	7	1233	7	1180	7	1206
188	435953									46	57							49	60	51	56
236	878280					30	61	29	88	28	106	26	116	24	134	27	111	30	106	22	143
246	377461			3	720	4	1198	4	1468	4	1545	4	1576	4	1571	4	1640	4	1613	4	1585
255	325182			12	131	12	279	13	332	12	387	11	420	12	419	11	446	12	413	11	469
257	740801									50	52										
335	1469292					23	83	28	90	34	90	29	95	33	99	26	112	26	119	26	125
365	1434905							47	50			45	57	47	58	50	55	44	72		
368	1473131							41	57	44	62			38	72	37	89	37	78	44	69
380	289645															54	52	47	64	50	56
417	395708							31	79	31	96	28	95	35	95	31	102	28	112	30	116
430	379708							42	57	47	54	37	80	44	63	44	72	40	75	39	76
509	207274	1	91	2	1076	2	1603	2	1796	2	1870	2	1865	2	1941	2	1982	2	2001	2	1965
545	1435862			5	522	6	974	6	1093	6	1152	6	1251	6	1347	6	1307	6	1343	6	1345
554	461425			14	111	13	232	16	279	16	300	16	284	16	295	16	284	16	287	16	296
566	357031					37	50	38	60	45	60	39	72	40	69	41	78	45	66	41	73
585	68977											52	51					53	56		
742	812105	2	88	1	1078	1	1745	1	1911	1	1963	1	2019	1	2065	1	2123	1	2107	1	2097
783	767183							45	50	42	64	42	63	41	69	34	96	46	65	43	69
836	241412					31	58	34	74	35	87	32	89	39	70	32	102	39	76	27	125
842	810057									52	51	49	55	56	50					53	54
846	183337			16	95	15	211	15	279	13	315	15	321	15	308	15	312	15	332	15	326
951	841620													51	54	56	50	56	53		
976	786084					25	74	27	93	25	114	23	118	27	115	24	121	24	125	25	133
1003	796258			17	84	17	141	17	180	17	197	17	210	18	196	17	239	18	196	17	202
1055	1409509					34	53	40	59	29	100	25	117	25	134	29	110	33	97	32	103
1066	486110													57	50						
1084	878652			18	70	20	111	21	128	19	173	20	175	23	137	19	197	20	179	20	170
1116	626502											44	60	43	63	51	53	51	57	45	67
1158	814526					26	72	22	113	27	111	24	117	20	142	23	132	21	143	23	141
1207	143306					33	54	30	83	30	100	35	85	32	100	30	109	32	100	36	89
1263	324494													52	53						
1319	866702			13	123	16	210	14	287	15	300	14	335	14	335	14	350	14	346	14	337
1327	491565															48	56	59	50	55	51
1386	745019					22	87	23	106	23	126	21	141	21	141	20	160			21	145
1387	770394									43	62							27	116		
1389	770394	3	50	4	712	3	1339	3	1621	3	1774	3	1763	3	1873	3	1866	3	1867	3	1862
1434	784257					32	57	35	74	33	92	33	88	28	106	28	110	25	120	29	119
1601	629896			7	422	7	755	8	918	8	990	8	955	8	1047	8	1012	8	1048	8	1005
1606	624360			19	65	18	133	19	139	20	140	19	175	19	158	21	147	19	188	19	182
1613	80338					27	72	36	65	36	78	31	92	31	100	40	81	35	89	38	77
1645	52076			11	172	11	324	11	378	11	423	12	398	11	457	12	433	11	423	12	457
1662	377048									38	77	43	62	46	60	42	74	58	51	54	53
1700	796475									49	52	53	50	53	52	45	68	42	73	48	59
1708	43733					29	69	33	76	26	112	34	88	29	102	33	96	31	105	34	97
1721	40643															52	53				
1764	44563					19	121	18	145	18	186	18	193	17	207	18	207	17	202	18	201
1776	768246							43	55			47	56			53	53	54	56	47	65
1884	609663							44	53	39	73	46	57	49	57	47	63	43	73	46	65
1911	898219																	52	57		
1915	840942																			52	55
1916	80109					24	78	24	104	22	130	27	109	26	130	22	133	23	131	24	136
1954	814260			9	242	9	536	9	662	9	745	9	784	9	802	9	825	9	858	9	842
1955	784224			6	435	5	1020	5	1279	5	1376	5	1452	5	1521	5	1490	5	1562	5	1580
1980	841641													54	51			57	52	56	50
1991	740554									51	52							55	55		
2046	244618			10	186	10	344	10	458	10	465	10	516	10	521	10	528	10	548	10	552
2050	295985					35	52	32	76	37	77	38	76	37	77	35	94	41	74	33	101
2144	308231											50	52					60	50	49	57
2157	244637											51	52	48	57	55	52			57	50
2186	208699							46	50	40	68			55	51						
2199	135688							39	59			48	55	42	66	46	68	38	77	42	71

Continued on Next Page...

Table B.12 – *Continued*

(d) Threshold-based system																					
		Fitness Evaluation																			
Gene	Imgage	5000		10000		15000		20000		25000		30000		35000		40000		45000		50000	
Index	Id.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652			22	90	27	135	25	171	20	222	23	213	22	223	24	210	24	203	24	221
74	193913							51	54	50	59	55	55	47	68	53	64	58	57	46	77
85	297392			24	88	22	160	22	196	21	214	18	252	20	253	23	215	20	247	20	257
107	365826													58	55			60	54	60	55
123	236282					42	52	42	73	45	65	40	89	41	87	41	87	43	73	40	94
129	298062											52	57					62	53	64	50
153	383188			10	215	10	429	10	507	10	515	11	508	9	584	10	576	10	581	10	558
165	283315					34	89	32	119	36	99	34	113	34	127	34	126	33	139	35	125
166	897177											51	59			59	52	64	51	54	61
187	296448			9	273	8	545	8	708	8	839	8	898	8	902	8	893	7	939	7	944
188	435953									56	51	58	50			49	70	59	55	59	57
236	878280					41	60	36	89	46	65	38	100	38	91	42	83	35	109	36	111
246	377461			6	529	6	893	6	1053	6	1124	6	1193	6	1204	6	1233	6	1254	6	1247
251	486787							54	52							61	51				
255	325182			17	118	15	257	14	328	14	361	15	357	14	393	14	421	15	351	15	346
276	868304							49	56			47	70	55	57	50	69	50	65	45	77
335	1469292			18	117	18	185	20	236	24	207	21	237	21	241	20	262	22	233	21	252
365	1434905													56	57	58	53	63	51	57	60
368	1473131																	57	59		
380	289645									55	55					60	51				
407	195751																			63	50
417	395708							41	77	51	58	42	78	45	72	48	70	44	72	41	92
509	207274	1	152	2	1056	2	1424	4	1565	4	1586	4	1573	4	1585	4	1615	4	1635	4	1700
545	1435862	3	81	5	648	5	1086	5	1254	5	1297	5	1342	5	1389	5	1317	5	1398	5	1422
554	461425			25	86	24	152	21	203	23	209	22	217	23	213	21	256	21	244	22	245
585	68977											54	55	53	61	56	57			53	62
589	769657													54	60			49	66		
742	812105	2	151	1	1315	1	1986	1	2248	1	2370	1	2439	1	2491	1	2443	1	2540	1	2560
758	47475							44	64	37	94	37	100	35	120	36	123	40	92	38	105
783	767183			32	51	31	99	34	107	31	138	31	132	32	141	33	145	32	144	34	125
836	241412			27	82	25	146	29	138	26	182	29	166	28	176	29	179	29	183	28	206
842	810057					36	72	38	83	33	123	36	108	36	114	37	112	38	101	37	111
846	183337			8	295	9	435	9	513	9	559	9	599	10	579	9	594	9	644	9	620
937	789204																	65	51		
951	841620											45	76	48	67	55	60	46	67	52	63
976	786084			29	63	30	112	28	138	30	155	30	159	30	162	30	166	30	178	31	155
1003	796258			16	125	16	254	17	266	16	306	14	359	16	342	16	352	17	315	16	333
1055	1409509					32	99	35	91	35	109	35	110	37	105	35	125	36	104	33	134
1067	489489									49	60	50	60	62	51	47	72	52	63	56	60
1084	878652			30	63	33	96	30	138	32	128	33	124	31	147	31	156	31	157	30	165
1116	626502			21	93	21	164	26	170	25	200	25	190	25	192	26	205	28	194	25	215
1158	814526			23	89	20	178	19	242	19	227	20	238	18	290	18	286	16	318	17	288
1159	142788																	61	54		
1207	143306			31	54	28	129	31	137	29	171	27	179	27	185	28	188	26	202	29	188
1295	344134									54	56	53	56	50	63	52	65	45	68	51	64
1301	346696													51	63					58	57
1319	866702			28	64	29	115	27	157	28	174	26	181	29	172	27	190	23	222	27	208
1327	491565							37	88	39	76	44	77	44	78	43	78	39	96	43	82
1386	745019			26	82	23	154	24	178	27	181	24	206	24	201	25	207	27	198	23	222
1387	770394					38	67	40	82	41	73	43	78	40	87	39	91	37	104	42	92
1389	770394	4	62	3	709	3	1302	2	1625	2	1862	2	1908	2	1913	2	2009	2	1991	2	2049
1434	784257					43	50	43	68	44	66	46	74	42	82	54	62	51	64	48	67
1497	203003					39	62	46	61	38	78	41	81	43	81	44	76	47	67	44	79
1536	530185					40	62	47	59	48	61	56	54	49	66	51	69	53	63	55	60
1601	629896			7	381	7	678	7	782	7	858	7	920	7	927	7	898	8	923	8	939
1606	624360			13	187	13	305	13	410	13	389	13	420	13	395	13	447	13	414	13	394
1613	80338							48	57	47	63	49	60	52	62	40	87	42	84	49	67
1626	811000							50	56	57	51			61	52			55	61		
1634	82903											57	51								
1645	52076			19	104	19	181	18	243	18	239	19	251	19	259	17	301	19	269	19	261
1662	377048									53	57										
1700	796475							53	53									48	67	62	51
1738	771323									40	75			59	54	57	55	66	50		
1884	609663			20	96	26	141	23	187	22	211	28	176	26	190	22	216	25	202	26	209
1910	628336																	67	50		
1915	840942					37	72	39	82	43	66	39	96	39	88	38	101	41	86	39	104
1916	80109			14	160	17	227	16	274	17	283	17	288	17	312	19	276	18	296	18	285
1932	782811			15	132	14	280	15	283	15	352	16	342	15	364	15	370	14	382	14	349
1954	814260			11	206	12	361	11	432	11	486	10	537	12	526	11	564	11	547	11	529
1955	784224			4	688	4	1287	3	1618	3	1751	3	1797	3	1794	3	1908	3	1887	3	1882
2046	244618			12	195	11	367	12	421	12	478	12	475	11	537	12	531	12	510	12	488
2050	295985			33	50	35	83	33	117	34	110	32	130	33	130	32	145	34	132	32	147
2117	139957																			61	52
2157	244637							45	62	52	57			46	69	45	73	56	60	47	72
2186	208699							52	54	42	68	48	62	57	57	46	72	54	61	50	66
2199	135688													60	53						

Table B.13: The complete list of ALL/AML genes with different precision levels.

Gene Index	Accession Number	SIGMOID-BASED SYSTEM						LINEAR-BASED SYSTEM						TANH-BASED SYSTEM						THRESHOLD-BASED SYSTEM					
		Precision Level						Precision Level						Precision Level						Precision Level					
		100%		98%		95%		100%		98%		95%		100%		98%		95%		100%		98%		95%	
		Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
412	D42043_at	35	67					26	115	42	59			38	63					36	71				
668	D86967_at	38	64					40	73	41	59			34	74	40	50			38	61				
758	D88270_at	20	153	19	161	15	86	33	83	35	81			21	129	21	134	18	71			36	65		
760	D88422_at	13	208	11	230	12	139	39	74	33	92	16	98	14	213	13	212	12	135	43	55	32	81	17	85
804	HG1612-HT1612_at	6	653	6	481	13	138	6	549	9	380	14	104	6	643	6	513	13	132	7	357	10	230	26	63
1144	J05243_at	43	52	37	50									36	65	37	54								
1239	L07633_at	25	104	30	77			19	161	31	101			27	92	27	83			28	99	43	61		
1400	L21954_at																					46	55		
1630	L47738_at	44	51											42	53										
1674	M11147_at							43	67	28	112	12	136							44	53	13	189	9	201
1685	M11722_at	5	674	5	636	4	262	13	219	14	184	23	69	5	695	5	640	4	279	14	159	18	165	27	61
1704	M13792_at	37	64	27	82									35	67	33	65								
1745	M16038_at	36	66	36	54			42	68	45	54			40	57	31	74					42	62		
1779	M19507_at	8	416	8	366	7	189	4	719	3	932	3	579	8	440	8	386	7	160	3	642	2	1133	3	611
1796	M20902_at	39	60					24	127					31	83					16	147	45	56		
1829	M22960_at	23	111	24	90	22	50	21	144	17	149	17	98	28	91	29	79			18	141	17	166	13	107
1834	M23197_at	28	90	31	75			44	66	37	77	25	63	41	55	28	81			41	58	38	64		
1882	M27891_at	1	1666	1	1469	2	594	7	476	6	485	5	334	1	1662	1	1434	2	670	4	628	5	695	4	446
1928	M31303_rna1_at	31	86	28	78			31	96	34	81			32	79	25	90			37	64	40	65		
1941	M31994_at	40	57																	46	51				
1962	M33680_at	14	206	14	195	19	64	12	264	12	231			12	237	14	204	16	81	17	143	23	129		
1975	M34344_at							32	83					45	50					22	131				
2020	M55150_at							34	82	27	116									30	80	31	81	25	63
2111	M62762_at							25	124	25	123	20	77							25	111	19	159	14	96
2121	M63138_at	11	251	15	187	14	89	5	718	4	700	4	400	11	263	12	222	14	89	5	568	4	730	5	376
2288	M84526_at	2	1221	2	1179	1	1008	2	875	2	1075	1	846	2	1291	2	1155	1	951	2	779	3	1089	1	1002
2335	M89957_at			35	58			52	52	43	55					35	55								
2354	M92287_at	4	930	4	785	5	209	3	791	5	600	9	182	3	934	4	726	5	245	6	539	6	452	12	126
2363	M93056_at																							28	53
2394	M95678_at									46	52											30	86		
2402	M96326_rna1_at	17	194	12	225	9	146	30	104	22	134	11	155	13	221	11	233	9	154	34	73	14	185	7	230
2408	M96803_at													43	53										
2546	S82470_at									40	62	32	51									39	65		
2642	U05259_rna1_at	7	547	7	444	6	204	8	372	10	338	13	134	7	500	7	495	6	201	12	207	16	175	15	88
3252	U46499_at	19	155	18	172	10	145	15	194	13	229	8	218	18	146	17	163	11	143	15	148	11	225	10	177
3258	U46751_at							38	78	29	107	28	59							47	50	24	114		
3320	U50136_rna1_at							28	113	26	120	19	78							20	133	21	153	18	80
3984	U94855_at									44	55														
4050	X03934_at	27	91					51	53					30	87										
4095	X06948_at																			45	51				
4196	X17042_at			34	58	23	50	17	180	11	312	6	252	44	52	39	53			9	260	7	437	6	346
4211	X51521_at	12	214	20	155			11	265	18	148			15	198	19	154			13	190	25	113		
4229	X52056_at	30	86	25	88	21	56	47	59	38	63	29	56	26	92	32	65	22	53						
4328	X59417_at	10	328	10	275	11	141	9	366	8	408	10	177	9	331	10	337	10	148	11	223	8	310	11	149
4373	X62320_at	22	117	23	90	20	61	37	79	30	103	21	76	25	96	22	103	21	57			41	62	19	77
4377	X62654_rna1_at	21	149	21	110	18	64	16	183	20	142	15	99	20	143	20	141	23	53	26	102	29	87	20	76
4409	X64594_at							54	51																
4438	X66401_cds1_at							48	58																
4680	X82240_rna1_at	26	99	26	87									24	98	23	97	19	68						
4847	X95735_at	3	1001	3	945	3	366	1	1486	1	1464	2	841	4	922	3	876	3	339	1	1294	1	1339	2	712
4951	Y07604_at	42	53					14	196	23	132									10	243	22	151		
5062	Z14982_rna1_at									47	52														
5445	X04526_at							55	51																
5501	Z15115_at	18	160	17	177	17	65	23	127	24	131			19	144	15	179	15	85	27	100	27	92		
5552	L06797_s_at															36	54								
5772	U22376_cds2_s_at	15	202	13	198	16	80	22	128	32	100			16	188	16	174	17	76	32	76	37	65		
5950	M29610_s_at							46	63											29	89				
5952	U05255_s_at							27	114											19	140	33	70		
6041	L09209_s_at	9	368	9	324	8	162	10	366	7	460	7	243	10	319	9	341	8	158	8	304	9	504	8	221
6049	U89922_s_at	24	108											29	89	38	53			42	56				
6079	U59632_s_at	45	51																	33	75				
6184	M26708_s_at							49	56																
6185	X64072_s_at											30	54									47	50	21	74
6200	M28130_rna1_s_at	33	85	29	78			20	147	16	160	18	81	23	104	26	85			24	113	15	177	22	71
6201	Y00787_s_at	29	87	32	73			36	79	19	143	22	70	33	74	30	78			35	71	12	194	16	87
6215	M19508_xpt3_s_at							29	105	21	142	27	62							21	133	20	154	23	70
6225	M84371_rna1_s_at							53	52																
6271	M33493_s_at	34	70											39	60							44	57		
6539	X85116_rna1_s_at							41	71	36	79									31	78	28	89		
6376	M83652_s_at	32	85	22	96			35	81	39	62	24	65	22	105	24	92			40	59	35	65		
6702	X97267_rna1_s_at	41	55	33	66			45	63					37	64	34	65								
6855	M31523_at	16	200	16	181			18	177	15	167	26	62	17	172	18	158	20	63	23	118	26	97	29	52
6796	J02982_f_at							50	54											39	61				
6806	X14008_rna1_f_at											31	52									34	66	24	67
7119	U29175_at													46	50										

Table B.14: The complete list of SRBCTs genes with different precision levels.

Gene Index	Image Id.	SIGMOID-BASED SYSTEM						LINEAR-BASED SYSTEM						TANH-BASED SYSTEM						THRESHOLD-BASED SYSTEM					
		Precision Level						Precision Level						Precision Level						Precision Level					
		100%		98%		95%		100%		98%		95%		100%		98%		95%		100%		98%		95%	
		Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
1	21652	**24**	*144*	**28**	*119*	**28**	*91*	**23**	*245*	**21**	*243*	**20**	*198*	**38**	*88*	**34**	*88*	**31**	*79*	**24**	*210*	**24**	*200*	**22**	*170*
74	193913							**48**	*70*	**51**	*64*	**53**	*52*							**53**	*64*				
85	297392	**27**	*126*	**24**	*138*	**24**	*115*	**20**	*261*	**19**	*274*	**19**	*228*	**36**	*91*	**30**	*106*	**27**	*97*	**23**	*215*	**21**	*225*	**20**	*183*
107	365826	**53**	*54*	**44**	*65*					**47**	*74*	**37**	*102*	**43**	*73*	**41**	*72*	**43**	*56*			**47**	*71*	**41**	*78*
123	236282	**37**	*91*	**37**	*78*	**42**	*57*	**41**	*87*	**39**	*93*	**46**	*68*	**39**	*82*	**35**	*84*	**32**	*74*	**41**	*87*	**46**	*77*	**44**	*77*
129	298062	**25**	*133*	**33**	*90*	**41**	*59*							**25**	*121*	**28**	*111*	**28**	*88*					**10**	*431*
153	383188	**11**	*448*	**10**	*476*	**10**	*371*	**9**	*664*	**9**	*682*	**9**	*543*	**13**	*395*	**10**	*458*	**11**	*349*	**10**	*576*	**10**	*565*		
165	283315	**40**	*78*	**54**	*50*			**31**	*155*	**35**	*141*	**35**	*111*	**49**	*55*					**34**	*126*	**39**	*101*	**45**	*72*
166	897177							**56**	*56*	**66**	*52*									**59**	*52*				
187	296448	**7**	*1142*	**7**	*1291*	**6**	*1162*	**10**	*625*	**8**	*787*	**8**	*682*	**7**	*1233*	**5**	*1386*	**5**	*1227*	**8**	*893*	**7**	*990*	**7**	*935*
188	435953	**48**	*59*	**52**	*51*					**55**	*60*	**45**	*74*							**49**	*70*	**52**	*57*		
236	878280	**29**	*118*	**20**	*157*	**20**	*155*	**39**	*95*	**36**	*120*	**33**	*118*	**27**	*111*	**18**	*189*	**17**	*167*	**42**	*83*	**33**	*136*	**28**	*130*
246	377461	**5**	*1602*	**5**	*1461*	**5**	*1173*	**6**	*997*	**5**	*1086*	**5**	*901*	**4**	*1640*	**4**	*1484*	**4**	*1245*	**6**	*1233*	**6**	*1199*	**6**	*1025*
251	486787																			**61**	*51*				
255	325182	**14**	*417*	**14**	*323*	**14**	*215*	**16**	*379*	**17**	*343*	**16**	*298*	**11**	*446*	**13**	*303*	**13**	*238*	**14**	*421*	**16**	*333*	**16**	*243*
276	868304									**57**	*59*									**50**	*69*	**48**	*65*		
335	1469292	**23**	*150*	**21**	*156*	**21**	*131*	**18**	*309*	**20**	*252*	**21**	*180*	**26**	*112*	**27**	*114*	**25**	*103*	**20**	*262*	**19**	*240*	**23**	*166*
365	1473131	**47**	*59*					**54**	*59*					**50**	*55*					**58**	*53*				
368	1473131													**37**	*89*	**49**	*55*	**44**	*53*						
380	289645	**46**	*59*	**40**	*69*									**54**	*52*	**50**	*54*			**60**	*51*	**57**	*50*		
417	395708	**36**	*95*	**34**	*85*	**33**	*78*	**43**	*85*	**40**	*92*	**38**	*95*	**31**	*102*	**26**	*115*	**21**	*117*	**48**	*70*	**37**	*103*	**38**	*86*
430	379708	**56**	*53*											**44**	*72*	**36**	*83*	**45**	*51*						
509	207274	**3**	*1989*	**3**	*1797*	**3**	*1419*	**4**	*1300*	**6**	*1032*	**7**	*718*	**2**	*1982*	**1**	*1828*	**2**	*1530*	**4**	*1615*	**5**	*1369*	**4**	*1062*
545	1435862	**6**	*1506*	**6**	*1372*	**7**	*1099*	**5**	*1157*	**4**	*1373*	**4**	*1036*	**6**	*1307*	**7**	*1277*	**7**	*1036*	**5**	*1317*	**4**	*1396*	**5**	*1049*
554	461425	**16**	*311*	**16**	*210*	**18**	*140*	**24**	*245*	**26**	*191*	**31**	*125*	**16**	*284*	**16**	*219*	**18**	*157*	**21**	*256*	**25**	*194*	**29**	*128*
566	357031													**41**	*78*	**46**	*63*	**38**	*67*						
585	68977							**59**	*51*	**52**	*62*	**54**	*52*							**56**	*57*	**50**	*61*		
742	812105	**1**	*2366*	**2**	*2020*	**1**	*1643*	**1**	*2606*	**1**	*2596*	**1**	*2164*	**1**	*2123*	**3**	*1752*	**3**	*1397*	**1**	*2443*	**1**	*2249*	**2**	*1877*
758	47475			**48**	*57*	**36**	*65*	**36**	*122*	**18**	*275*	**18**	*253*							**36**	*123*	**23**	*200*	**26**	*144*
783	767183	**33**	*101*	**39**	*74*	**39**	*63*	**29**	*169*	**33**	*151*	**36**	*102*	**34**	*96*	**40**	*75*	**41**	*61*	**33**	*145*	**32**	*137*	**30**	*124*
836	241412	**30**	*112*	**29**	*105*	**26**	*98*	**26**	*209*	**28**	*188*	**24**	*163*	**32**	*102*	**33**	*98*	**39**	*66*	**29**	*179*	**30**	*150*	**32**	*117*
842	810057	**51**	*57*	**50**	*53*			**33**	*137*	**31**	*153*	**34**	*114*							**37**	*112*	**36**	*105*	**34**	*102*
846	183337	**12**	*440*	**13**	*336*	**13**	*240*	**8**	*710*	**10**	*630*	**11**	*415*	**15**	*312*	**14**	*299*	**15**	*212*	**9**	*594*	**11**	*556*	**11**	*373*
910	839552									**60**	*57*														
937	789204							**55**	*58*																
951	841620	**58**	*51*	**43**	*64*	**44**	*52*	**46**	*73*	**43**	*91*	**41**	*82*	**56**	*50*	**43**	*67*	**33**	*73*	**55**	*60*	**42**	*87*	**40**	*82*
976	786084	**28**	*124*	**22**	*151*	**32**	*83*	**30**	*160*	**34**	*150*	**29**	*134*	**24**	*121*	**23**	*133*	**24**	*106*	**30**	*166*	**28**	*163*	**31**	*119*
1003	796258	**18**	*243*	**19**	*182*	**19**	*137*	**14**	*473*	**16**	*420*	**17**	*285*	**17**	*239*	**19**	*160*	**23**	*109*	**16**	*352*	**17**	*280*	**17**	*221*
1055	1409509	**34**	*101*	**41**	*69*	**43**	*56*	**34**	*127*	**38**	*109*	**44**	*74*	**29**	*110*	**42**	*70*	**42**	*57*	**35**	*125*	**43**	*80*	**49**	*57*
1066	486110									**62**	*55*														
1067	489489							**47**	*73*	**58**	*58*	**47**	*65*							**47**	*72*	**53**	*55*	**46**	*72*
1084	878652	**22**	*168*	**27**	*121*	**25**	*106*	**38**	*100*	**45**	*79*	**49**	*63*	**19**	*197*	**21**	*144*	**26**	*99*	**31**	*156*	**34**	*120*	**37**	*86*
1090	755145									**56**	*59*														
1116	626502	**41**	*67*	**36**	*80*	**31**	*89*	**25**	*243*	**23**	*237*	**22**	*178*	**51**	*53*	**48**	*62*	**35**	*71*	**26**	*205*	**22**	*206*	**25**	*151*
1158	814526	**19**	*205*	**18**	*195*	**16**	*177*	**17**	*338*	**14**	*450*	**13**	*360*	**23**	*132*	**22**	*140*	**19**	*132*	**18**	*286*	**15**	*336*	**13**	*318*
1159	142788									**67**	*50*														
1207	143306	**26**	*131*	**30**	*101*	**30**	*90*	**21**	*260*	**25**	*223*	**30**	*132*	**30**	*109*	**39**	*76*	**36**	*67*	**28**	*188*	**29**	*151*	**33**	*109*
1295	344134							**58**	*55*	**61**	*56*									**52**	*65*	**56**	*51*	**48**	*58*
1263	324494			**53**	*50*																				
1301	346696	**54**	*54*																						
1319	866702	**15**	*313*	**15**	*265*	**15**	*209*	**35**	*124*	**29**	*163*	**26**	*148*	**14**	*350*	**15**	*277*	**14**	*224*	**27**	*190*	**27**	*168*	**21**	*172*
1327	491565	**42**	*65*	**46**	*61*	**37**	*64*	**37**	*106*	**37**	*116*	**39**	*88*	**48**	*56*			**46**	*50*	**43**	*78*	**38**	*103*	**39**	*85*
1386	745019	**20**	*172*	**23**	*144*	**23**	*116*	**27**	*198*	**27**	*189*	**28**	*140*	**20**	*160*	**29**	*106*	**30**	*85*	**25**	*207*	**26**	*179*	**27**	*133*
1387	770394	**43**	*64*	**45**	*65*			**40**	*93*	**41**	*92*	**42**	*82*							**39**	*91*	**41**	*90*	**47**	*66*
1389	770394	**2**	*2113*	**1**	*2057*	**2**	*1605*	**2**	*2044*	**2**	*2360*	**2**	*2125*	**3**	*1866*	**2**	*1826*	**1**	*1567*	**2**	*2009*	**2**	*2131*	**1**	*1924*
1434	784257	**35**	*97*	**32**	*92*	**35**	*71*	**42**	*87*	**42**	*91*	**48**	*65*	**28**	*110*	**25**	*119*	**29**	*88*	**54**	*62*	**45**	*79*	**43**	*77*
1497	203003			**47**	*58*			**44**	*83*	**44**	*90*	**43**	*80*							**44**	*76*	**44**	*80*	**42**	*77*
1536	530185	**57**	*52*					**49**	*68*	**59**	*58*	**51**	*58*							**51**	*69*	**54**	*55*		
1601	629896	**8**	*982*	**8**	*851*	**8**	*668*	**7**	*842*	**7**	*926*	**6**	*770*	**8**	*1012*	**9**	*871*	**9**	*649*	**7**	*898*	**8**	*886*	**8**	*779*
1606	624360	**17**	*264*	**17**	*206*	**17**	*172*	**12**	*514*	**15**	*433*	**15**	*329*	**21**	*147*	**20**	*152*	**20**	*127*	**13**	*447*	**13**	*380*	**14**	*284*
1613	80338	**32**	*102*	**42**	*65*			**50**	*62*	**50**	*65*	**50**	*60*	**40**	*81*	**38**	*76*			**40**	*87*	**49**	*63*		
1626	811000							**57**	*56*	**68**	*50*														
1634	82903									**63**	*54*														
1645	52076	**13**	*425*	**12**	*347*	**12**	*291*	**28**	*197*	**32**	*153*	**27**	*142*	**12**	*433*	**12**	*401*	**12**	*297*	**17**	*301*	**18**	*259*	**19**	*186*
1662	377048	**52**	*56*	**51**	*51*									**42**	*74*	**45**	*66*								
1700	796475	**59**	*50*											**45**	*68*	**51**	*51*								
1708	43733	**49**	*58*	**38**	*75*	**38**	*64*							**33**	*96*	**31**	*103*	**37**	*67*						
1721	40643													**52**	*53*										
1738	771323							**45**	*80*	**54**	*60*	**52**	*55*							**57**	*55*				
1764	44563	**31**	*104*	**25**	*134*	**27**	*95*							**18**	*207*	**17**	*201*	**16**	*168*						
1776	768246													**53**	*53*										
1884	609663	**39**	*83*	**35**	*82*	**40**	*63*	**22**	*249*	**24**	*230*	**25**	*150*	**47**	*63*			**47**	*50*	**22**	*216*	**31**	*139*	**24**	*160*
1909	789357									**64**	*53*														
1911	898219							**60**	*50*	**49**	*69*														
1915	840942	**50**	*57*					**51**	*62*	**46**	*74*	**40**	*83*							**38**	*101*	**40**	*93*	**36**	*87*
1916	80109	**21**	*169*	**26**	*133*	**22**	*123*	**19**	*295*	**22**	*242*	**23**	*173*	**22**	*133*	**24**	*122*	**22**	*113*	**19**	*276*	**20**	*236*	**18**	*188*
1932	782811	**44**	*62*	**49**	*54*	**34**	*73*	**11**	*569*	**12**	*534*	**12**	*396*							**15**	*370*	**14**	*344*	**15**	*273*
1954	814260	**9**	*817*	**9**	*755*	**9**	*668*	**15**	*442*	**11**	*559*	**10**	*481*	**9**	*825*	**8**	*896*	**8**	*729*	**11**	*564*	**9**	*594*	**9**	*552*
1955	784224	**4**	*1802*	**4**	*1639*	**4**	*1318*	**3**	*1894*	**3**	*2240*	**3**	*1919*	**5**	*1490*	**6**	*1381*	**6**	*1190*	**3**	*1908*	**3**	*2043*	**3**	*1680*
2022	204545											**55**	*51*												
2046	244618	**10**	*555*	**11**	*464*	**11**	*348*	**13**	*511*	**13**	*502*	**14**	*358*	**10**	*528*	**11**	*426*	**10**	*359*	**12**	*531*	**12**	*484*	**12**	*339*
2050	295985	**38**	*87*	**31**	*96*	**29**	*91*	**32**	*150*	**30**	*155*	**32**	*120*	**35**	*94*	**32**	*100*	**40**	*64*	**32**	*145*	**35**	*118*	**35**	*92*
2157	244637	**55**	*53*											**55**	*52*	**47**	*63*			**45**	*73*	**51**	*61*	**50**	*55*
2186	208699	**45**	*60*																	**46**	*72*	**55**	*52*		
2199	135688													**46**	*68*	**37**	*77*	**34**	*71*						

Table B.15: The complete list of genes based on the raw and the normalised ALL/AML data sets.

Gene	Accession	RAW ALL/AML DATA SET								NORMALISED ALL/AML DATA SET (MIN-MAX NORMALISATION)							
		SIGMOID		LINEAR		TANH		THRESHOLD		SIGMOID		LINEAR		TANH		THRESHOLD	
Index	Number	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
235	D14664_at									**59**	*66*					**77**	*57*
412	D42043_at	**35**	*67*	**26**	*115*	**38**	*63*	**36**	*71*	**43**	94	**44**	100	**43**	90	**43**	92
461	D49950_at									**17**	207	**17**	211	**20**	160	**18**	186
490	D50918_at									**65**	62	**63**	64	**64**	58	**94**	50
635	D84294_at													**73**	52		
668	D86967_at	**38**	*64*	**40**	*73*	**34**	*74*	**38**	*61*	**73**	*57*	**69**	58			**86**	52
758	D88270_at	**20**	*153*	**33**	*83*	**21**	*129*			**60**	*66*	**60**	*69*	**50**	77	**60**	72
760	D88422_at	**13**	*208*	**39**	*74*	**14**	*213*	**43**	*55*	**49**	83	**41**	108	**52**	71	**53**	80
804	HG1612-HT1612_at	**6**	*653*	**6**	*549*	**6**	*643*	**7**	*357*	**20**	167	**18**	182	**24**	152	**19**	179
1144	J05243_at	**43**	*52*			**36**	*65*			**47**	89	**39**	113	**38**	97	**35**	117
1207	L05148_at									**82**	52	**86**	51	**76**	50	**76**	58
1239	L07633_at	**25**	*104*	**19**	*161*	**27**	*92*	**28**	*99*	**51**	77	**53**	85	**51**	*76*	**48**	87
1260	L09717_at									**32**	114	**31**	126	**45**	87	**39**	106
1291	L11669_at									**37**	107	**28**	133	**34**	102	**38**	110
1598	L41559_at											**72**	*56*				
1615	L42379_at									**68**	61	**77**	54				
1630	L47738_at	**44**	*51*			**42**	*53*									**84**	53
1669	M10612_at											**68**	58			**68**	65
1674	M11147_at			**43**	*67*			**44**	*53*	**18**	179	**23**	155	**19**	162	**21**	164
1685	M11722_at	**5**	*674*	**13**	*219*	**5**	*695*	**14**	*159*	**27**	136	**22**	156	**25**	133	**31**	125
1704	M13792_at	**37**	*64*			**35**	*67*										
1745	M16038_at	**36**	*66*	**42**	*68*	**40**	*57*			**12**	263	**11**	305	**11**	271	**11**	270
1779	M19507_at	**8**	*416*	**4**	*719*	**8**	*440*	**3**	*642*	**8**	366	**8**	446	**8**	321	**8**	385
1796	M20902_at	**39**	*60*	**24**	*127*	**31**	*83*	**16**	*147*	**35**	110	**27**	138	**39**	95	**24**	143
1829	M22960_at	**23**	*111*	**21**	*144*	**28**	*91*	**18**	*141*	**9**	315	**9**	362	**9**	284	**9**	346
1834	M23197_at	**28**	*90*	**44**	*66*	**41**	*55*	**41**	*58*	**4**	553	**4**	603	**5**	462	**5**	524
1882	M27891_at	**1**	*1666*	**7**	*476*	**1**	*1662*	**4**	*628*	**6**	473	**7**	496	**6**	433	**7**	441
1928	M31303_rnal_at	**31**	*86*	**31**	*96*	**32**	*79*	**37**	*64*	**28**	127	**25**	147	**22**	154	**29**	133
1941	M31994_at	**40**	*57*					**46**	*51*	**66**	62	**59**	71	**72**	53	**69**	63
1962	M33680_at	**14**	*206*	**12**	*264*	**12**	*237*	**17**	*143*	**30**	118	**35**	121	**33**	103	**34**	118
1975	M34344_at			**32**	*83*	**45**	*50*	**22**	*131*	**75**	*56*					**52**	81
2020	M55150_at			**34**	*82*			**30**	*80*	**13**	260	**13**	273	**15**	199	**14**	253
2111	M62762_at			**25**	*124*			**25**	*111*	**14**	252	**16**	230	**13**	241	**12**	263
2121	M63138_at	**11**	*251*	**5**	*718*	**11**	*263*	**5**	*568*	**10**	293	**10**	315	**10**	274	**10**	337
2134	M63589_at									**85**	51	**56**	*76*	**77**	50	**74**	*59*
2242	M80254_at									**83**	51	**87**	51			**89**	51
2288	M84526_at	**2**	*1221*	**2**	*875*	**2**	*1291*	**2**	*779*	**16**	220	**14**	255	**18**	171	**16**	247
2295	M85169_at									**78**	*54*						
2335	M89957_at			**52**	*52*							**74**	*56*	**65**	*57*	**62**	68
2354	M92287_at	**4**	*930*	**3**	*791*	**3**	*934*	**6**	*539*	**15**	228	**15**	248	**14**	241	**15**	252
2363	M93056_at									**41**	98	**37**	117	**35**	101	**41**	103
2402	M96326_rnal_at	**17**	*194*	**30**	*104*	**13**	*221*	**34**	*73*	**26**	140	**36**	120	**27**	125	**30**	129
2408	M96803_at					**43**	*53*			**80**	*52*	**92**	*50*				
2422	M98045_at									**81**	*52*					**91**	*51*
2439	S46622_at									**74**	*57*	**67**	*59*			**87**	*52*
2441	S50223_at									**39**	105	**46**	*95*	**40**	94	**49**	84
2475	S72008_at									**86**	*51*					**73**	*60*
2546	S82470_at									**62**	*62*	**50**	*87*	**67**	*55*	**56**	*78*
2642	U05259_rnal_at	**7**	*547*	**8**	*372*	**7**	*500*	**12**	*207*	**22**	149	**20**	171	**23**	154	**20**	171
2739	U10868_at									**48**	87	**49**	89	**53**	*69*	**51**	82
3012	U30255_at															**78**	*56*
3183	U41635_at									**40**	101	**42**	107	**47**	84	**40**	103

Continued on Next Page...

Table B.15 – *Continued*

Gene Index	Accession Number	RAW ALL/AML DATA SET								NORMALISED ALL/AML DATA SET (MIN-MAX NORMALISATION)							
		SIGMOID		LINEAR		TANH		THRESHOLD		SIGMOID		LINEAR		TANH		THRESHOLD	
		Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.	Rank	Freq.
3252	U46499_at	**19**	*155*	**15**	*194*	**18**	*146*	**15**	*148*	**3**	614	**2**	745	**2**	599	**2**	683
3258	U46751_at			**38**	*78*			**47**	*50*	**63**	62	**71**	57			**72**	60
3320	U50136_rna1_at			**28**	*113*			**20**	*133*	**11**	278	**12**	304	**12**	262	**13**	258
3722	U77948_at									**64**	62	**64**	64	**59**	62	**70**	61
3847	U82759_at									**21**	156	**21**	168	**16**	180	**17**	188
4050	X03934_at	**27**	*91*	**51**	*53*	**30**	*87*										
4095	X06948_at							**45**	*51*								
4141	X13973_at															**83**	54
4196	X17042_at			**17**	*180*	**44**	*52*	**9**	*260*	**2**	635	**3**	733	**3**	582	**3**	661
4211	X51521_at	**12**	*214*	**11**	*265*	**15**	*198*	**13**	*190*	**55**	71	**58**	74	**36**	100	**61**	68
4229	X52056_at	**30**	*86*	**47**	*59*	**26**	*92*			**44**	92	**52**	85	**56**	64	**59**	74
4291	X56468_at									**70**	60					**81**	55
4328	X59417_at	**10**	*328*	**9**	*366*	**9**	*331*	**11**	*223*	**19**	175	**19**	177	**17**	180	**22**	157
4366	X61587_at									**24**	141	**33**	125	**46**	85	**27**	138
4373	X62320_at	**22**	*117*	**37**	*79*	**25**	*96*										
4377	X62654_rna1_at	**21**	*149*	**16**	*183*	**20**	*143*	**26**	*102*	**57**	68	**61**	68	**54**	69	**57**	77
4389	X63469_at									**69**	60	**55**	77			**54**	80
4409	X64594_at			**54**	*51*												
4438	X66401_cds1_at			**48**	*58*					**58**	67	**57**	74	**57**	64	**58**	77
4447	X66867_cds1_at											**83**	52			**71**	61
4644	X80230_at											**88**	51				
4680	X82240_rna1_at	**26**	*99*			**24**	*98*										
4847	X95735_at	**3**	*1001*	**1**	*1486*	**4**	*922*	**1**	*1294*	**1**	882	**1**	925	**1**	759	**1**	858
4936	Y00433_at													**69**	53		
4951	Y07604_at	**42**	*53*	**14**	*196*			**10**	*243*	**7**	452	**6**	552	**7**	421	**6**	478
4973	Y08612_at									**34**	113	**40**	110	**41**	94	**45**	91
5094	Z24727_at									**52**	74	**48**	89	**37**	98	**47**	87
5107	Z29067_at									**71**	59	**79**	54	**60**	60	**65**	66
5280	J02783_at											**76**	55				
5335	M21535_at											**81**	53	**71**	53		
5445	X04526_at			**55**	*51*												
5501	Z15115_at	**18**	*160*	**23**	*127*	**19**	*144*	**27**	*100*	**61**	63	**66**	59	**74**	51	**75**	58
5552	L06797_s_at															**90**	51
5766	HG2562-HT2658_s_at													**62**	59		
5772	U22376_cds2_s_at	**15**	*202*	**22**	*128*	**16**	*188*	**32**	*76*	**42**	96	**45**	98	**31**	110	**37**	114
5949	M29610_at									**76**	56						
5950	M29610_s_at			**46**	*63*			**29**	*89*			**65**	62				
5952	U05255_s_at			**27**	*114*			**19**	*140*	**67**	62	**80**	54			**66**	66
6005	M32304_s_at									**45**	91	**47**	92	**58**	63	**44**	91
6041	L09209_s_at	**9**	*368*	**10**	*366*	**10**	*319*	**8**	*304*	**5**	510	**5**	560	**4**	466	**4**	544
6049	U89922_s_at	**24**	*108*			**29**	*89*	**42**	*56*								
6079	U59632_s_at	**45**	*51*					**33**	*75*			**73**	56			**93**	50
6169	M13690_s_at									**36**	108	**29**	131	**32**	106	**32**	124
6184	M26708_s_at			**49**	*56*							**89**	51	**55**	66	**64**	66
6185	X64072_s_at											**78**	54	**63**	58	**80**	55
6200	M28130_rna1_s_at	**33**	*83*	**20**	*147*	**23**	*104*	**24**	*113*								
6201	Y00787_s_at	**29**	*87*	**36**	*79*	**33**	*74*	**35**	*71*	**25**	141	**30**	128	**29**	123	**25**	142
6215	M19508_xpt3_s_at			**29**	*105*			**21**	*133*	**77**	54	**90**	50	**70**	53		
6225	M84371_rna1_s_at			**53**	*52*					**23**	145	**24**	148	**21**	154	**23**	153
6271	M33493_s_at	**34**	*70*			**39**	*60*										
6281	M31211_s_at									**46**	90	**43**	103	**44**	90	**55**	80
6283	M65214_s_at									**72**	59	**62**	64	**49**	78	**50**	83
6302	X13955_s_at															**92**	50
6347	M63838_s_at									**50**	77	**54**	78	**61**	60	**46**	90
6373	M81695_s_at									**84**	51	**82**	53	**68**	55	**63**	66
6376	M83652_s_at	**32**	*85*	**35**	*81*	**22**	*105*	**40**	*59*	**53**	72	**85**	52	**75**	51	**85**	53
6405	M98399_s_at											**91**	50	**66**	57	**79**	56
6539	X85116_rna1_s_at			**41**	*71*			**31**	*78*	**33**	113	**32**	126	**42**	91	**33**	120
6702	X97267_rna1_s_at	**41**	*55*	**45**	*63*	**37**	*64*										
6796	J02982_f_at			**50**	*54*			**39**	*61*								
6797	J03801_f_at									**54**	72					**67**	65
6801	L49229_f_at											**70**	58			**82**	54
6803	M19045_f_at									**79**	53	**75**	55				
6806	X14008_rna1_f_at									**56**	69	**51**	85	**48**	80	**42**	99
6855	M31523_at	**16**	*200*	**18**	*177*	**17**	*172*	**23**	*118*	**29**	121	**26**	139	**28**	124	**28**	137
6919	X16546_at									**38**	105	**38**	116	**30**	110	**26**	138
6974	M28170_at									**31**	116	**34**	125	**26**	127	**36**	116
7119	U29175_at					**46**	*50*					**84**	52			**88**	51

Table B.16: The complete list of attributes selected by GANN in the bioassay data set. For AID362 data set, a top-28 attributes were selected and for AID688, a top-31 attributes were selected.

AID362			AID688		
Index	Attribute description	Data type	Index	Attribute description	Data type
66	HBD_03_ARC	Binary	147	PSA	Real
142	MW	Real	151	MW	Real
57	HBD_05_HBD	Binary	113	ARC_06_HYP	Binary
139	NumRot	Integer	114	ARC_07_HYP	Binary
93	ARC_01_ARC	Binary	101	HBA_07_HYP	Binary
143	BBB	Binary	146	XLogP	Real
67	HBD_04_ARC	Binary	112	ARC_05_HYP	Binary
23	NEG_07_ARC	Binary	107	ARC_06_ARC	Binary
22	NEG_06_ARC	Binary	89	HBA_06_HBA	Binary
61	HBD_04_HBA	Binary	98	HBA_04_HYP	Binary
62	HBD_05_HBA	Binary	119	HYP_05_HYP	Binary
15	NEG_05_HBA	Binary	131	WBN_EN_H_0.25	Real
65	HBD_02_ARC	Binary	106	ARC_05_ARC	Binary
24	NEG_02_HYP	Binary	99	HBA_05_HYP	Binary
7	NEG_03_POS	Binary	86	HBA_03_HBA	Binary
14	NEG_04_HBA	Binary	95	HBA_07_ARC	Binary
74	HBD_05_HYP	Binary	42	POS_03_HBD	Binary
29	NEG_07_HYP	Binary	74	HBD_02_ARC	Binary
33	POS_07_POS	Binary	103	ARC_02_ARC	Binary
91	HBA_06_HYP	Binary	148	NumRot	Integer
18	NEG_02_ARC	Binary	153	BadGroup	Integer
31	POS_04_POS	Binary	32	NEG_03_HYP	Binary
16	NEG_06_HBA	Binary	64	HBD_03_HBD	Binary
122	WBN_EN_H_0.25	Real	79	HBD_07_ARC	Binary
13	NEG_07_HBD	Binary	25	NEG_02_ARC	Binary
70	HBD_07_ARC	Binary	91	HBA_03_ARC	Binary
10	NEG_06_POS	Binary	39	POS_05_POS	Binary
34	POS_03_HBD	Binary	40	POS_06_POS	Binary
			23	NEG_06_HBA	Binary
			96	HBA_02_HYP	Binary
			19	NEG_07_HBD	Binary

Appendix C

Related Works

This appendix contains relevant studies in the microarray data sets that were being used in this thesis.

Table C.1: Some relevant works in ALL/AML microarray data.

Author	Data preprocessing	No. of genes to be analysed	Method	Validation mechanism	Informative genes
Culhane et al. (2002)	*Not specified*	7159	COA/BGA; PCA/BGA	LOOCV	50
Bø and Jonassen (2002)	Similar to Dudoit et al. (2002)	3934	variety selections; FDA; KNN; DLDQ	Sample-splitting; LOOCV	50
Li et al. (2001a)	Log transformation	7129	GA/KNN	Sample-splitting	50
Dudoit et al. (2002)	Filtering; log transformation; normalisation	3571	BSS/WSS; DT; FDA; KNN	Sample-splitting; LOOCV	40
Guyon et al. (2002)	Log transformation; normalisation	7129	SVM/RFE	Sample-splitting; LOOCV	16
Li and Yang (2002)	Filtering; log transformation	1000	LRM	Sample-splitting	37
Tibshirani et al. (2002)	-	7129	NSC	Sample-splitting; 10-fold CV	21
Cho et al. (2003b)	*Not specified*	7129	FDA	5-fold CV; LOOCV	6
Futschik et al. (2003)	Normalisation; log transformation	*Not specified*	EFuNN; PCA; bootstrap	LOOCV; N-fold CV	-
Lee and Lee (2003)	Similar to Dudoit et al. (2002)	3571	BSS/WSS; SVM	Sample-splitting; LOOCV	40
Wang et al. (2003)	Log transformation; normalisation	7129	SOM; fuzzy C-means	-	-
Li et al. (2004)	Normalisation	7129	variety selections; DT; NB; SVM; KNN	Sample-splitting; 4-fold CV	150
Weber et al. (2004)	Filtering	37	LRM	-	2
Chu et al. (2005)	Filtering; normalisation	1169	NB; Wilcoxon ranksum	10-fold CV	14
Dabney (2005)	*Not specified*	3857	modified NSC (ClaNC)	*Not specified*	10
Jirapech-Umpai and Aitken (2005)	*Not specified*	7070	modified GESSES; GA/KNN	Sample-splitting; LOOCV	55
Mao et al. (2005)	Similar to Dudoit et al. (2002)	3571	SVM-RFE	LOOCV	20
Mramor et al. (2005)	*Not specified*	7074	ReliefF; S2N	-	20
Takahashi et al. (2005)	Filtering	5401	Projective adaptive resonance theory (PART); fuzzy ANN	Sample-splitting	10
Zhou and Mao (2005)	Similar to Dudoit et al. (2002); filtering	1000	Least squares SVM	LOOCV	12
Zhou et al. (2005)	Similar to Dudoit et al. (2002)	3571	Bayesian methods	*Not Applicable*	20
Sakhinia et al. (2006)	*Not applicable*	15	PCR; housekeeping; Wilcoxon ranksum	*Not applicable*	15
Cho and Won (2007)	Normalisation	7129	Ensemble ANNs	Sample-splitting	50
Xu et al. (2007)	Filtering	1000	Particle Swarm Optimisation (PSO); Ellipsoid ARTMAP (EAM)	10-fold CV; LOOCV	63-97

Table C.2: Some relevant works in SRBCTs microarray data.

Author	Data preprocessing	No. of genes to be analysed	Method	Validation mechanism	informative genes
Lidén et al. (2002)	-	2308	Rule-induction methods; IG	Sample-splitting	32-128
Culhane et al. (2002)	*Not specified*	2308	COA/BGA	Sample-splitting	50
Tibshirani et al. (2002)	-	2308	NSC	Sample-splitting; 10-fold CV	43
Cho et al. (2003b)	*Not specified*	2308	FDA	5-fold CV; LOOCV	21
Deutsch (2003)	-	2308	GESSES (EA approach)	Sample-splitting	15-28
Lee and Lee (2003)	Log transformation; normalisation	2308	BSS/WSS; SVM	Sample-splitting; LOOCV	20
Li et al. (2004)	Normalisation	2308	variety selections; DT; NB; SVM; KNN	Sample-splitting; 4-fold CV	150
Weber et al. (2004)	Filtering	62	LRM	-	7
Androulakis (2005)	*Not specified*	2303	Ensemble Trees	Sample-splitting	3-19
Dabney (2005)	*Not specified*	2307	modified NSC (ClaNC)	*Not specified*	8
Liu et al. (2005b)	Log transformation; normalisation	2308	Entropy-based correlation	Sample-splitting; LOOCV	58
Mao et al. (2005)	Filtering	200	SVM-RFE	LOOCV	20
Mramor et al. (2005)	*Not specified*	2308	ReliefF; S2N	-	20
Chen et al. (2007)	*Not applicable*	39	PCR; housekeeping genes; ANN	LOOCV	39
Pal et al. (2007)	-	2308	fuzzy C-means/ANN; RBF; SVM	Sample-splitting	7-20
Wang et al. (2007)	-	2308	t-test; fuzzy ANN	Sample-splitting; 5-fold CV	50
Yu et al. (2007)	-	2308	GP; DLDA; KNN	Sample-splitting; k-fold CV	54