

GENE SELECTION FOR SAMPLE SETS WITH BIASED DISTRIBUTIONS

by

Abu Hena Mustafa Kamal

A Thesis Submitted to the Faculty of
The College of Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Science

Florida Atlantic University

Boca Raton, Florida

April 2009

GENE SELECTION FOR SAMPLE SETS WITH BIASED DISTRIBUTIONS

by

Abu Hena Mustafa Kamal

This thesis was prepared under the direction of candidate's thesis advisor, Dr. Xingquan Zhu, Department of Computer Science and Engineering, and has been approved by the members of his supervisory committee. It was submitted to the faculty of The College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Master of Science.

SUPERVISORY COMMITTEE:

**Thesis Advisor
Dr. Xingquan Zhu**

Dr. Abhijit Pandya

Dr. Sam Hsu

Chairman, Department of Computer Science and Engineering

Dean, College of Engineering and Computer Science

Dean, Graduate College

Date

ACKNOWLEDGEMENTS

I express my heart-felt gratitude to Dr. Xingquan Zhu for his unparalleled guidance throughout this mammoth task of organizing the thesis. Whenever I was in confusion, he gave me the inspiration and direction to accomplish this milestone. Very special thanks to Dr. Sam Hsu. I remember the very first time I met him in his office, me with an anxious mind seeking a position in the department to achieve my academic goal. I must acknowledge the support and guidance from Dr. Abhijit Pandya whose advice helped me to take important decisions that will shape my academic and professional career. I am thankful to my advisors and the members of the TCN research group for welcoming me in the group.

I am also grateful to my uncles and my cousins living in West Palm Beach for making me feeling at home here. Last but not the least, I respectfully remember my parents whose encouragement provided me the strength to continue my academic pursuit far away from my home. I also thank my elder sister and my three brothers who always provided the emotional support to achieve this goal.

ABSTRACT

Author: Abu Hena Mustafa Kamal
Title: Gene Selection for Sample Sets with Biased Distributions
Institution: Florida Atlantic University
Thesis Advisor: Dr. Xingquan Zhu
Degree: Master of Science
Year: 2009

Microarray expression data which contains the expression levels of a large number of simultaneously observed genes have been used in many scientific research and clinical studies. Due to its high dimensionalities, selecting a small number of genes has shown to be beneficial for many tasks such as building prediction models from the microarray expression data or gene regulatory network discovery. Traditional gene selection methods, however, fail to take the class distribution into the selection process. In Biomedical science, it is very common to have microarray expression data which is severely biased with one class of examples (e.g., diseased samples) significantly less than other classes (e.g., normal samples). These sample sets with biased distributions require special attention from researchers for identification of genes responsible for a particular disease. In this thesis, we propose three filtering techniques, Higher Weight ReliefF, ReliefF with Differential Minority Repeat and ReliefF with Balanced Minority Repeat to

identify genes responsible for fatal diseases from biased microarray expression data. Our solutions are evaluated on five well-known microarray datasets, Colon, Central Nervous System, DLBCL Tumor, Lymphoma and ECML Pancreas. Experimental comparisons with the traditional ReliefF filtering method demonstrate the effectiveness of the proposed methods in selecting informative genes from microarray expression data with biased sample distributions.

TABLE OF CONTENTS

FIGURES.....	viii
TABLES.....	x
1 PREFACE.....	1
1.1 Motivation.....	1
1.2 Problem Statement.....	2
1.3 Scope of the Work.....	3
1.4 Author's Contribution.....	4
1.5 Structure of the Thesis.....	5
1.6 Summary of Important Notations.....	6
2 BACKGROUND AND RELATED WORK ON GENE SELECTION.....	7
2.1 Gene, the Blueprint of Life.....	7
2.2 Microarray Expression Data.....	15
2.3 Gene Selection in Molecular Biology.....	19
2.4 Gene Selection in Data Mining and Machine Learning.....	21
2.5 Class Imbalance Problem.....	26
2.6 Related Work on Gene Selection with Imbalanced Microarray Data.....	30
3 FOUNDATION OF THE THESIS AND PROPOSED SOLUTIONS.....	32
3.1 The ReliefF Filtering Algorithm.....	32
3.2 Higher Weight ReliefF.....	36

3.3	ReliefF with Differential Minority Repeat	38
3.4	ReliefF with Balanced Minority Repeat	41
4	EXPERIMENTAL SETTINGS AND ANALYSIS OF RESULTS	45
4.1	The Microarray Datasets of Experiments	45
4.2	WEKA.....	46
4.3	Performance Metrics.....	48
4.4	Analysis of performance on Colon and Central Nervous System Datasets.....	49
4.5	Performance Evaluation on DLBCL Tumor and Lymphoma Microarray Expression Data	54
4.6	Performance Evaluation on Pancreas Dataset.....	58
4.7	Overall Ranking of Four Filtering Methods	65
5	CONCLUSION.....	68
	APPENDIX A LIST OF TABLES CONTAINING PERFORMANCE RESULTS	70
	APPENDIX B LIST OF TABLES CONTAINING AUC.....	91
	REFERENCE:	94

FIGURES

Figure 2.1.1: The Basic Structure of a Cell	8
Figure 2.1.2: The location of Chromosome and DNA inside a cell	10
Figure 2.1.3: Chemical structure of DNA.....	12
Figure 2.1.4: A molecular gene.....	13
Figure 2.1.5: Splicing process of Introns	14
Figure 2.1.6: replication, transcription and translation.....	15
Figure 2.2.1: The colors of Microarray.....	17
Figure 2.2.2: Typical two color microarray experiment.....	18
Figure 3.1.1: The ReliefF algorithm	34
Figure 3.2.1: The Higher Weight ReliefF algorithm	38
Figure 3.3.1: Differential Minority Repeat on ReliefF.....	40
Figure 3.3.2: The Relief with Differential Minority Repeat algorithm	41
Figure 3.4.1: Balanced Minority Repeat with 10 minority and 90 majority examples	43
Figure 3.4.2: The Relief with Balanced Minority Repeat algorithm	44
Figure 4.2.1: A simple WEKA experiment window.....	47
Figure 4.4.1: TPR for Naïve Bayes Classifier on Central Nervous System Data.....	50
Figure 4.4.2: TPR for RF classifier on Central Nervous Data.....	51
Figure 4.4.3: TPR for IB1 classifier on Central Nervous data.....	51

Figure 4.4.4: Accuracy for IB1 classifier on Central Nervous data.....	52
Figure 4.4.5: TNR for IB1 classifier on Central Nervous data.....	53
Figure 4.4.6: BER for IB1 classifier on Central Nervous data	53
Figure 4.5.1: TPR for Naïve Bayes classifier on Lymphoma dataset.....	55
Figure 4.5.2: TPR for Random Forest classifier on Lymphoma data.....	55
Figure 4.5.3: TPR for IB1 classifier on Lymphoma data	56
Figure 4.5.4: Accuracy for Naïve Bayes classifier on Lymphoma dataset.....	57
Figure 4.5.5: TNR for Naïve Bayes classifier on Lymphoma dataset	57
Figure 4.5.6: BER for Naïve Bayes classifier on Lymphoma data.....	58
Figure 4.6.1: TPR for Naïve Bayes classifier on Pancreas data	59
Figure 4.6.2: TPR for Random Forest classifier on Pancreas data	59
Figure 4.6.3: TPR for IB1 classifier on Pancreas data.....	60
Figure 4.6.4: The 5th run of 10-fold CV on Pancreas Data.....	61
Figure 4.6.5: Accuracy for Naïve Bayes classifier on Pancreas data	63
Figure 4.6.6: TNR for Naïve Bayes classifier on Pancreas data.....	64
Figure 4.6.7: BER for Naïve Bayes classifier on Pancreas data.....	64

TABLES

Table 1.6.1: Summary of Important Notations Used.....	6
Table 4.1.1: The Microarray Expression datasets for experiments	45
Table 4.3.1: The Confusion Matrix of Classification	48
Table 4.7.1: Overall Ranking using AUC on TPR performance metric.....	65
Table 4.7.2: Overall Ranking using AUC on TNR performance metric	66
Table 4.7.3: Overall Ranking using AUC on Accuracy performance metric.....	67
Table 4.7.4: Overall Ranking using AUC on BER performance metric	67

Chapter 1

PREFACE

1.1 Motivation

Computer scientists and microbiologists have been working for decades to extract relevant genes responsible for fatal diseases from microarray expression data. A considerable amount of work has been done to select subset of genes from microarray data. Research is still going on to develop better methods for filtering these huge number of genes obtained from microarray data to a subset of genes informative of a particular disease. Unfortunately, this data often contains less number of diseased examples than normal ones, the situation known to scientists as data imbalance or class imbalance.

The imbalanced data makes it difficult for researchers and computer scientists to build a good classification model that can identify the rare or minority class properly. In this thesis, the minority class means the class which contains the least number of samples. In microarray expression data, the minority class often consists of diseased samples. Although the classification model built from

imbalanced data may attain significant overall accuracy threshold by identifying nearly all the majority classes accurately or simply predicting all input test cases to be of majority classes, however, in real-life situations the goal often is to determine the minority class from the test data.

If we consider each single gene of the microarray expression data as one individual feature, the gene selection for microarray data can be regarded as the feature selection problem in traditional data mining research, except that for microarray expression data we have a very large number of genes (e.g., more than 10,000) whereas the number of samples is relatively small (e.g, less than 100). Many feature filtering algorithms available in data mining and machine learning literature do not address the imbalanced class distribution in dataset and show very poor performance in classifying novel examples into appropriate class. The feature subset selection for imbalanced dataset is currently hot topic to bioinformatics and data mining researchers.

1.2 Problem Statement

The data from microarray expression analysis is very high dimensional and contains several thousands of attributes or genes. Often this data suffers from imbalanced class distribution. Appropriate preprocessing of that high dimensional and low volume data poses significant challenge to the research community as

eliminating any important feature may degrade the performance of the classification algorithm severely.

Our goal is to devise a solution that can extract the subset of genes that are sufficient to predict whether the patient under treatment may suffer from the disease or not. In this thesis, we propose three solutions based on ReliefF [1] preprocessing algorithm to handle the imbalanced data. These solutions significantly improve the ranking ability of ReliefF. The comparison of these three proposed solutions with the original ReliefF results in major discoveries that can be used to better handle the class imbalance problem.

1.3 Scope of the Work

Although the focus of our research targets only the microarray expression data with imbalanced class distribution, the filtering algorithms that we propose can be used in many diverse fields where class imbalance is prevalent and can have severe adverse impact. One major field of application could be Credit Card Fraud Detection. The number of fraudulent credit card transaction is very less than number of legitimate transactions. The process of detecting very infrequent number of fraudulent credit card transactions from a huge number of legal transactions can be benefited by applying the proposed filtering algorithms.

Network Intrusion Detection is another hot area where the goal is to make the network resources available to the registered users of the network and simultaneously protect the resources from the intruders who are not registered for the services. Here also the number of intruders is significantly smaller than the registered users. In broader sense, the whole community of Security and Networks can be benefited from the filtering algorithms that we suggest.

The next large area where it can be applied is in production-line assembly of different industries. Generally, the number of defective products coming out of the assembly line is very lower compared to non-defective ones, may be one out of thousands. Detecting this defective product can be crucial especially in medical industries, automobiles and transportation, foods and beverages etc. where the malfunctioning product or device can cause injury to humans leading to death sometimes.

1.4 Author's Contribution

In this thesis paper we propose three feature subset selection techniques for microarray expression data especially with imbalanced class distribution. Higher Weight ReliefF simply determines prior class distribution from training data and assigns higher weight on attributes from minority class. The ReliefF with Differential Minority Repeat maneuvers the training data in such a way that the dataset becomes balanced with respect to class distribution and then the original

ReliefF filtering method is applied to determine the goodness of the attributes. The third and final algorithm is ReliefF with Balanced Minority Repeat which modifies and splits the dataset into some smaller datasets so that each dataset contains all the minority examples with equal number of majority examples, the method then uses ReliefF to weigh all the attributes. Later these attributes or genes are sorted and predefined subsets of genes are selected to build the classification model to measure the performance characteristics.

1.5 Structure of the Thesis

The rest of this thesis is structured as follows, in chapter 2, we cover the theoretical background of this work, introduce some breakthrough related work, in chapter 3 we elaborate the foundation filtering technique and discuss about our proposed solutions, in chapter 4, we describe experimental settings, implementation details and analysis of results on different well-known datasets and in chapter 5 we conclude this report with suggestions on choosing the filtering techniques proposed.

1.6 Summary of Important Notations

Table 1.6.1: Summary of Important Notations Used

Notation	Meaning
S	Training Dataset
I	Individual Instance in Dataset
A	Attribute Vector
$W[A]$	Weight Vector for A
R	A randomly selected instance
C	The class attribute value of an instance
H	Set of instances that belong to the same class as R
M	Set of instances that belong to a different class than R
$S_{minority}$	Set of minority examples in the training dataset
$S_{majority}$	Set of majority examples in the training dataset
$N_{minority}$	Number of minority examples in the training dataset
$N_{majority}$	Number of majority examples in the training dataset
D_n	Difference between number of majority and minority examples

Chapter 2

BACKGROUND AND RELATED WORK ON GENE SELECTION

2.1 *Gene, the Blueprint of Life*

The genes in their simplest forms are the key molecules that store the characteristic information of any living organism. The science of molecular biology and genetics has evolved into a mature stage since its inception by Augustinian monk Gregor Mendel in 1865. In 1953 James D. Watson and Francis Crick, with their discoveries on DNA molecule structure, established the Central Dogma of modern Molecular biology. Since then numerous research works and discoveries in the field of genetics and molecular biology have significantly helped scientists to better understand the evolution, adaptation and mutation of different living organisms, specifically human beings.

Although, the term gene is generally used to denote the units of hereditary and characteristic information storage for any living organisms (plants, animals and

bacteria etc.), in this thesis, we will relate and describe genes that are typically particular to humans. To understand gene, we have to understand the structure of cells which are the building blocks of different parts (skin, hair, kidney, blood, etc.) of human body. The following figure [figure 2.1.1] shows the basic structure of a cell.

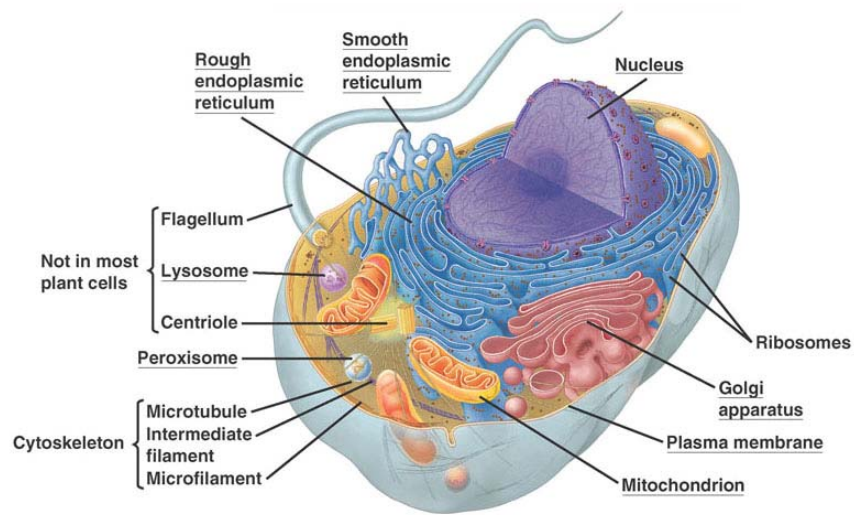


Figure 2.1.1: The Basic Structure of a Cell

Each cell is separated from its neighboring ones by a plasma membrane consisting of phospholipids. The phosphate end of the membrane is hydrophilic (attracted to water) and the lipid end is hydrophobic (repelled by water). Each cell has two layers of these phospholipids molecules and it ensures that water and other cell materials do not leak through the membrane except through some special pores [2, 3].

Cytoplasm is the gel-like fluid enclosed by the membrane which holds all the cell materials (nucleus, mitochondrion, ribosome etc.) in place and provides the shape of the cell. Mitochondrion acts as the power house of a cell and converts food into energy. The endoplasmic reticulum is involved in the production of cell membrane itself. The Golgi apparatus are involved in packaging materials that will be exported from the cell. Lysosomes contain substances that are used to digest proteins. Ribosomes are very important structures of a cell and are composed of 65% ribosomal RNA and 35% ribosomal proteins. They translate messenger RNA (mRNA) to build polypeptide chains (e.g. proteins) using amino acids delivered by transfer RNA (tRNA) [4]. It is the place where translation from genetic information into protein takes place.

The nucleus is the largest organelle of a cell and contains the most genetic information in the form of long DNA stretch inside the chromosome. In a growing or differentiating cell the nucleus is metabolically active and produces DNA and RNA. Each chromosome consists of a long single molecule of DNA associated with an equal mass of proteins. Collectively, the entire set of DNA and their associated proteins together are called chromatin [figure 2.1.2].

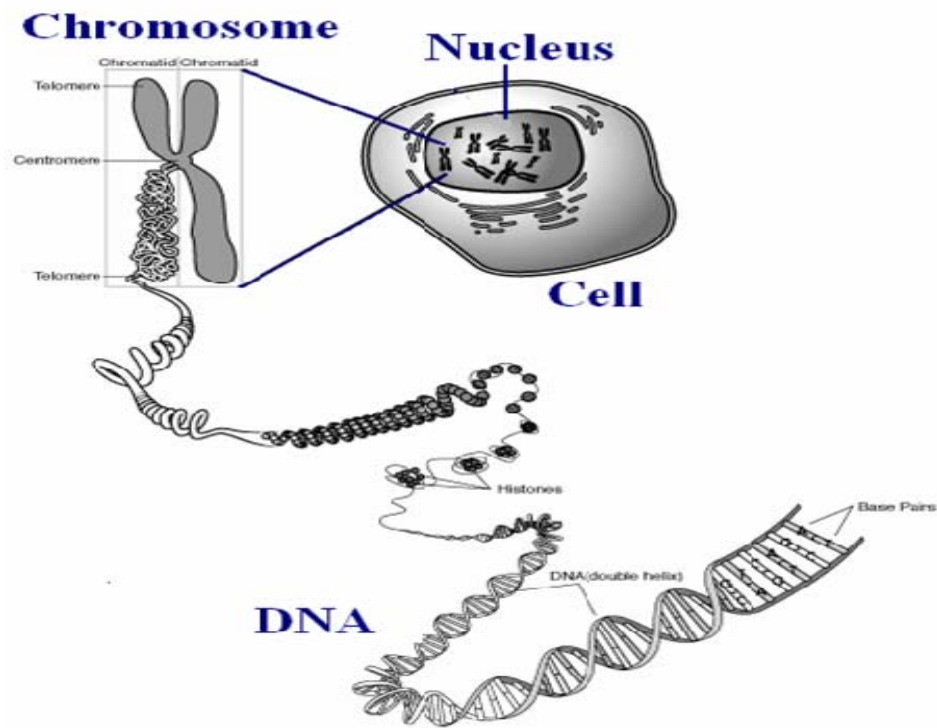


Figure 2.1.2: The location of Chromosome and DNA inside a cell, source (www.mtsinai.on.ca/pdmg/Genetics/basic.htm)

All the cells of human body the skin, hair, bone, blood are made up of proteins; they provide the structural support of the body; they act as the enzymes so that the chemical reactions necessary for existence of life can take place inside the cell; they act as antibodies to protect us from viruses and bacteria; they act as sensors that see and taste and smell; they are the switches that control whether the genes are turned on or off. Finding the proteins that make up a living organism and understand their functionalities are the foundations in molecular biology.

The structure of all proteins consists of 20 naturally occurring amino acids. Different orderings of these twenty different types of amino acids account for

different types of proteins in living organisms. The sequences of these amino acids are often called the primary structures of proteins. The primary structures of all possible proteins of a particular organism are coded inside the genetic materials of the cell or more specifically inside the DNA of the cell.

Chromosomes generally appear in pairs and reside inside the cell nucleus. Human cells have 23 pairs of chromosomes. Each chromosome is a long chain of DNA molecules and contains different types of DNA bound proteins which hold the DNA structure. The DNA chain is a polymer of four simple nucleic acid units or nucleotides called Adenine (A), Guanine (G), Cytosine (C) and Thymine (T). In DNA Adenine forms chemical bond exclusively with Thymine (A-T pair) and Guanine bonds exclusively with Cytosine (G-C pair). These complementary pairs of A-T and G-C are remarkably known as base-pairs and are units of measuring the length of DNA structure. These complementary nucleotide bonds are held together by hydrogen bonds and this arrangement of base-pairs has a helical structure which is termed as DNA double helix [figure 2.1.3]. The direction of each strand of DNA is identified by a head (the 5' end) and a tail (the 3' end).

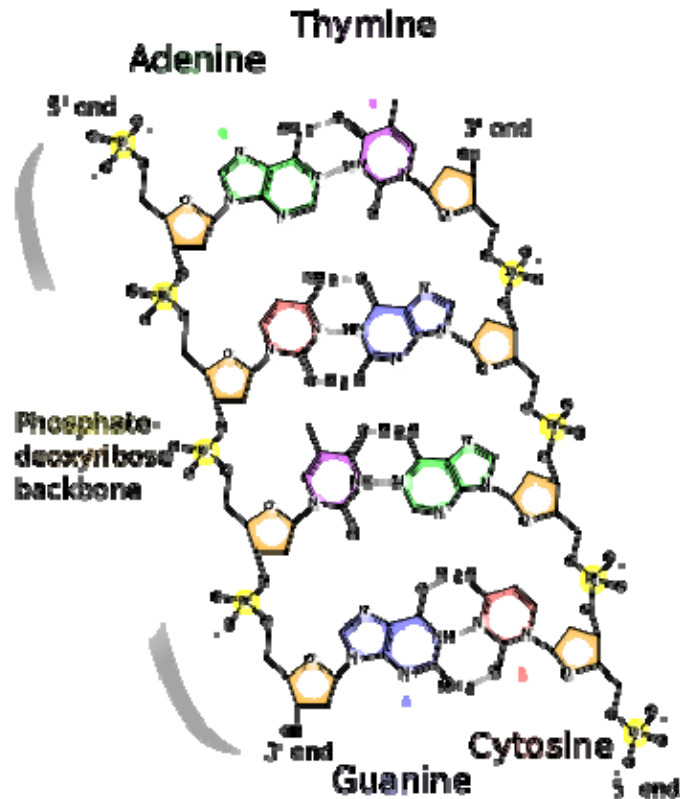


Figure 2.1.3: Chemical structure of DNA; Hydrogen bonds are shown as dotted lines, (source: http://en.wikipedia.org/wiki/Image:DNA_chemical_structure.svg)

A fragment of DNA that contains the code for protein synthesis and other functional products of the cell is called gene. Genes in a cell are either expressed or not-expressed. In that way, they can be thought as switches that are turned-on or turned-off based on the need of a particular type of a cell. For example, in skin cells the type of genes that contain code for producing proteins necessary for skin cells are expressed but genes for bones, genes for blood cells or other types of genes are not-expressed. In fact, each type of cell contains all types of genes but only a particular set of genes are expressed while all others are not-expressed. It is the expressed genes that are responsible for different types of cells (skin, hair, bone, blood etc.) in our body.

The promoter region adjacent to the gene controls the transcription and translation of gene into protein with the help of the enzyme called RNA polymerase. Another fundamental aspect of DNA is that it can replicate itself so that two identical copies of double-stranded DNA are formed from one single double-stranded DNA. This DNA replication occurs before cells divide and is a precondition to forming new cells. The replication, transcription and translation process form the central dogma of molecular biology. Generally, a gene consists of a head, sequences of introns and exons and a tail [figure 2.1.4].

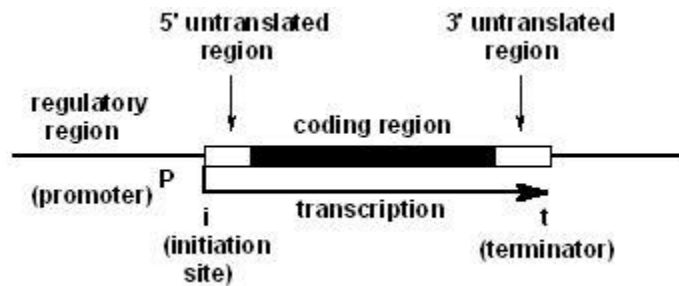


Figure 2.1.4: A molecular gene

The horizontal line in figure 2.1.4 is the double stranded DNA; a section of this DNA marked by the rectangle is identified as a gene. The gene contains both coding and non-coding sequences of base-pairs. The non-coding sequences of DNA or the introns regulate the gene expression but do not take part actively in protein synthesis. The coding sequences or the exons contain the actual code for protein synthesis. The enzyme RNA polymerase binds to promoter region of DNA. The promoter region sends signals to RNA polymerase which catalyzes a reaction that causes the DNA to be used as a template to produce complementary strand of RNA molecule. This RNA molecule is called primary transcript and

contains sequences of exons and introns. These non-coding regions or introns are eliminated by splicing and the exons are joined together to form the mature mRNA [figure 2.1.5].

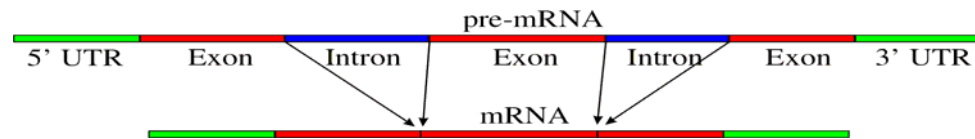
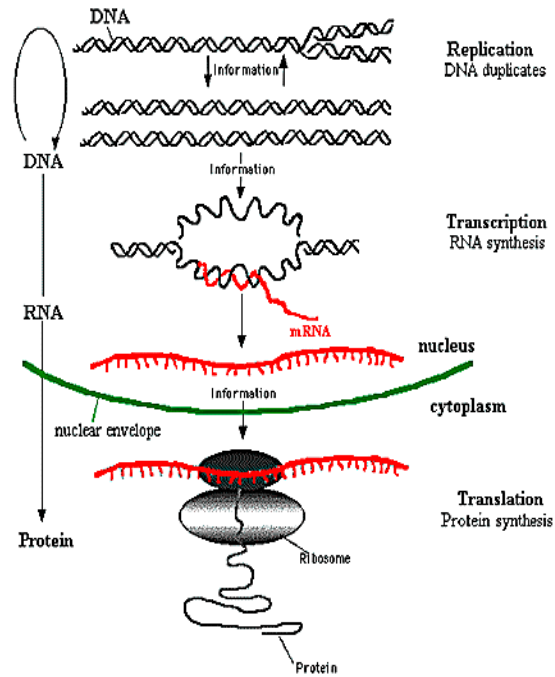


Figure 2.1.5: Splicing process of Introns

This mRNA is now transported to the cell cytoplasm through nucleus pores and is added to the ribosome. It now is used as blueprint for protein production of a particular type. This process of producing protein from mRNA is called translation. This whole process of replication, transcription and translation is summarized in figure 2.1.6.



The Central Dogma of Molecular Biology

Figure 2.1.6: replication, transcription and translation

2.2 Microarray Expression Data

A microarray is a type of chip made of glass slide or silicon chip on which fragments of DNA or genes are attached and is used to identify the expression levels of target genes from patient's cells. Two widely used microarray expression analyses are Spotted and Oligonucleotide microarrays [5]. More than 104 – 105 genes on a 1-2 cm² array can be examined in a massively parallel fashion. High precision optical detection methods are used to gather expression data after hybridization of probe and target.

A Probe is a single stranded gene (cDNA) or mRNA whose expression level is to be measured. The probes are immobilized and printed on predefined locations of a substrate (glass slide, quartz or nylon filters). As the mRNAs are not stable they are converted to more stable cDNA sequences. The Target is the cDNA representations of mRNAs taken from patient's tissues (e.g. human pancreatic carcinoma cells). In another way we can say that the probe is the known value and the target is the unknown value and their interaction results in the values that help us to discover the unknown values.

The probes and the targets are differentially labeled with fluorescent dyes. Usually Cy3 (green) and Cy5 (red) fluorescent dyes are used to label probes and targets. Then probe microarray and target microarray are hybridized and the hybridized array is scanned by the optical slide reader that recognizes the intensity levels of the dyes. These intensity levels of the dyes are directly related to the expression level of the genes.

For example, if genes on a probe microarray are labeled with Cy3 (green) and genes on the target microarray are labeled with Cy5 (red), after hybridization, an excess of green over red for a particular gene means that gene is more expressed in normal cells, an excess of red over green means that gene is more expressed in target or diseased cell, a yellow spot means that gene is equally expressed in both normal and diseased cell whereas a black spot on the hybridized microarray indicates that gene is not hybridized [figure 2.2.1].

The Colors of a Microarray



Reproduced with permission from the Office of Science Education, the National Institutes of Health.

In this schematic:

GREEN represents **Control DNA**, where either DNA or cDNA derived from normal tissue is hybridized to the target DNA.

RED represents **Sample DNA**, where either DNA or cDNA is derived from diseased tissue hybridized to the target DNA.

YELLOW represents **a combination of Control and Sample DNA**, where both hybridized equally to the target DNA.

BLACK represents areas where **neither the Control nor Sample DNA** hybridized to the target DNA.

Each spot on an array is associated with a particular gene. Each color in an array represents either healthy (control) or diseased (sample) tissue. Depending on the type of array used, the location and intensity of a color will tell us whether the gene, or mutation, is present in either the control and/or sample DNA. It will also provide an estimate of the expression level of the gene(s) in the sample and control DNA.

Figure 2.2.1: The colors of Microarray, source
(<http://www.ncbi.nlm.nih.gov/About/primer/microarrays.html>)

A high resolution image of the hybridized microarray is produced using laser technology. This image is then used to detect the expression levels of genes as the color intensity of the spots in the image is directly proportional to the expression

levels of the genes. This whole process of probe (Normal cell) and target (Cancer Cell) microarray hybridization technique is shown in figure 2.2.2.

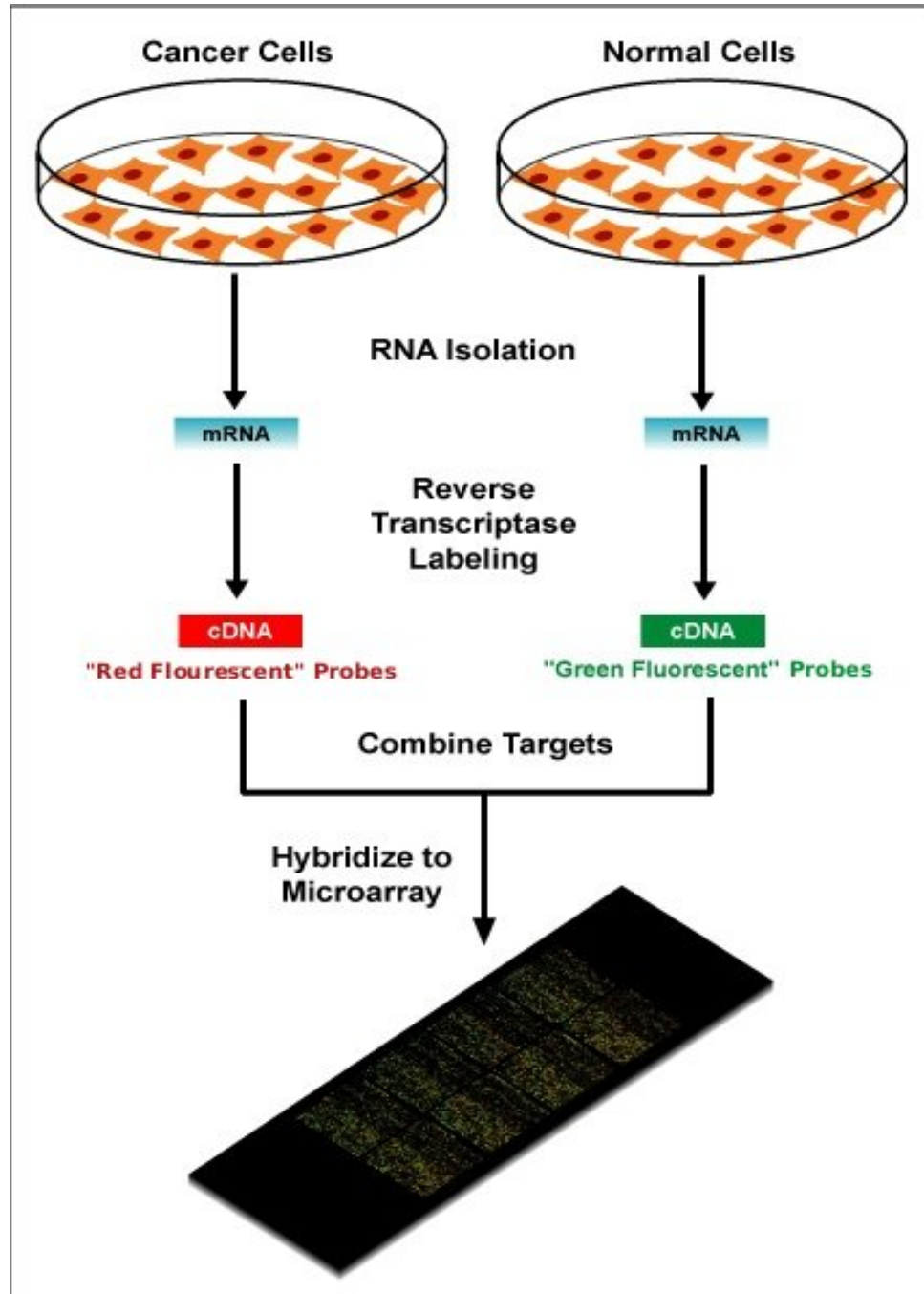


Figure 2.2.2: Typical two color microarray experiment, source (<http://upload.wikimedia.org/wikipedia/en/c/c8/Microarray-schema.jpg>)

2.3 *Gene Selection in Molecular Biology*

The set of all genes in any organism is called genome. As we all know, each cell of a human contains 23 pairs of chromosomes (46 chromosomes in total). For example, chromosome 1 contains about 3,148 genes; chromosome 2 contains about 902 genes etc. It is found that in total the human genome contains about thirty thousand (30,000) different genes. Sequences of these genes are turned on or off to control the cell division and protein synthesis process. The cells divide in a controlled and orderly manner which is the key to the growth of the body. However, sometimes due to some internal and external factors this process starts malfunctioning and shows uncontrolled growth of cells in a particular organ. This causes tumors to be formed in that organ and eventually the patient is under attack of cancer.

The genes can be broadly categorized into two different types:

- a. Genes whose protein products actively take part in stimulating the cell division process are called proto-oncogenes. The damaged or mutated versions of these genes are called oncogenes and may contribute to tumor growth by prohibiting the natural death of cells.
- b. Genes whose protein products prevent cell division are called tumor-suppressors [6].

Tumor suppressors regulate the transcription, DNA repair and cell-cell communication of an organism. The proto-oncogenes are used to initiate and stimulate the cell-division process whereas the suppressors are used like a brake to stop the process when necessary. In normal cells these two types of genes work hand-in-hand to control the growth of our body in a regulated way and control different types of biological processes. Nevertheless, damages in any of these two types of genes or both may result in abnormal behavior in cells and in fact, all types of cancers show alterations in one or more types of tumor suppressors and oncogenes.

Out of all 30,000 or more genes in human genome only a small subset of genes are found to be responsible for causing cancer. As an example, scientists at the University of Illinois at Chicago have found that there are about 57 genes involved in breast cancer growth [7]. Another team of researchers at the University of Michigan Comprehensive Cancer Center reports to find 158 genes specific to pancreatic cancer [8]. Their findings are thought to be the most accurate to date as they were able to distinguish genes involved in pancreas cancer from those involved in chronic inflammatory diseases like pancreatitis, a disease often mistaken for cancer. The team later narrowed down the list of genes from 158 to 80 that were three times more expressed in pancreatic cancer cells than in non-cancerous or pancreatitis cells.

2.4 Gene Selection in Data Mining and Machine Learning

Gene or feature selection is fundamental to many classification problems in Data Mining and Machine Learning arena. A large number of irrelevant features may introduce noise into the datasets and may even lead to wrong classification of the target sample. Moreover, a large number of potentially unnecessary features or genes make it difficult for the domain experts to extract significant and relevant information from the classification model. A fewer number of attributes or genes can help to build the classification model faster and with less computing resources.

Selecting an optimal or sub-optimal subset of necessary features for the problem at hand has long been ventured by a number of researchers. It has been found that many classification algorithms perform badly in presence of irrelevant features. [9, 10, 11] have shown that sample complexity (number of training examples needed to reach a given accuracy level) for nearest neighbor classification algorithm grows exponentially with increasing number of irrelevant features. Same holds true for decision tree classification. For example, C4.5 [12, 29] decision tree algorithm results in large trees due to irrelevant attributes. [13] proves that removing irrelevant and redundant attributes results in smaller trees. The naïve Bayes classifier also suffers from the pitfall of redundant attributes [14].

With this obvious importance of extracting relevant features, a number of feature selection algorithms can be found in literature and significant amount of research is going on in this hot area. Feature selection algorithms can be categorized into two broad categories, 1) Feature Filters and 2) Feature Wrappers. In Filter methods the feature selection process is independent of the classification algorithms whereas in wrapper methods the feature selection strategy is dependent on classification algorithms.

FOCUS [15] is a feature filter method that finds the minimum combination of features that can classify all the training data into distinct classes. This minimum set of features is called 'min-feature bias'. This algorithm may lead to exhaustive feature search in the feature space and overfitting may occur as the algorithm tries to add features to repair a single inconsistency [16]. LVF [17] filtering algorithm described by Liu and Setiono generates a random subset S from the feature subset space during each round of execution. Now if number of features in S is less than number of features in current best subset, then the inconsistency rate of S is compared with the inconsistency rate of current best subset. If S is at least as consistent as the current best subset, then S replaces current best subset. Rough Sets theory [18, 19] uses similar concept of consistency like LVF to eliminate redundant features.

The Chi2 [20] algorithm is a method that uses discretization to reduce the features. Another approach uses one learning method as a filter and a different

learning method for classification. Cardie [21] uses decision tree algorithm to select features for instance based learner. Singh and Provan [22] uses a greedy oblivious decision tree algorithm to filter attributes for building Bayesian network.

An Information Theoretic Feature Filter recently suggested by Koller and Sahami [23] benefits from information theory and probabilistic reasoning. This algorithm starts with all the features and employs a backward elimination search to remove, at each stage, the feature that causes least change between two distributions. Specifically, if C is a set of classes, V is a set of features, X is a subset of V , v is an assignment of values (v_1, \dots, v_n) to the features in V , v_x is the projection of values in X , then the algorithm tries to find X , so that, $Pr(C | X = v_x)$ is as close as $Pr(C | V = v)$.

There exists a set of correlation based filtering approaches that consider feature–class correlation and feature–feature inter–correlation. These filtering methods try to remove irrelevant and redundant features. A feature V_i is relevant if and only if there exists some v_i and c where $Pr(V_i = v_i) > 0$ and

$$Pr(C = c | V_i = v_i) \neq Pr(C = c)$$

A feature that is not relevant according to the above equation is said to be irrelevant and should be removed from the dataset before applying the learning method [24, 25]. Now a feature is redundant if it is highly correlated with other

features in the dataset. A good subset of features must contain features that are highly correlated with the class but uncorrelated with each other.

The Correlation based Feature subset Selector (CFS-subset) filters the irrelevant and redundant features so that the chosen subset of features contains features highly correlated with the class. The evaluation function for choosing a subset S of features is as follows,

$$M_s = \frac{k \overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad (2.1)$$

Here, M_s denotes the merit of the feature subset S with k number of features. $\overline{r_{cf}}$ is the mean feature-class correlation and $\overline{r_{ff}}$ is the mean feature-feature inter-correlation. The numerator tells how predictive of the class the set of features are and the denominator tells how much redundancy there is among the features.

Symmetrical Uncertainty [26] employs entropy based ranking of the features. The entropy of a feature Y , where y is a vector of the values of attribute Y , is given as,

$$H(Y) = -\sum_{y \in Y} p(y) \log_2(p(y)) \quad (2.2)$$

If X is another attribute in the training dataset then we can calculate the conditional entropy for Y as,

$$H(Y | X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y | x) \log_2(p(y | x)) \quad (2.3)$$

Now the information gain for attribute Y is the change in entropy of Y before observing X and after observing X .

$$gain = H(Y) - H(Y | X) \quad (2.4)$$

This information gain is symmetric; the amount of information gained for Y after observing X is equal to the amount of information gained for X after observing Y . This gain is biased for attributes with more values but a good measure for feature-feature inter-correlation. Symmetrical Uncertainty eliminates the bias for features with more values and normalizes the values to the range $[0, 1]$.

$$Symmetrical\ uncertainty\ coefficient = 2.0 \times \left[\frac{gain}{H(Y) + H(X)} \right] \quad (2.5)$$

Relief set of algorithms (Relief, ReliefF & RReliefF) are also correlation based filter algorithms which are covered in depth in section 3. Minimum Descriptor Length (MDL) [27] is another filter algorithm that also fits into this category.

The next major class of feature selection methods is called Wrapper method. This Wrapper method for feature subset selection uses the same learning algorithm for selecting the features and building the classification model. This method shows better performance than Filters as the feature subset selection algorithm is tuned to the classification method. It is slower than filters as the induction algorithm needs to be called repeatedly and must be re-run each time the classification method changes. This is a major drawback compared to filters method.

Early experiments on ID3 and C4.5 based decision wrapper methods are conducted by [28, 29] and the performance characteristics are estimated by 25-fold cross validation. Although the performance does not change that much but the major success using wrapper method is smaller trees. Cherkaur and Shavlik [30] develops an algorithm SET-Gen that uses a fitness function to select the attributes. This method results in smaller decision trees with higher accuracy. An algorithm OBLIVION [31] uses backward feature elimination with the help of an oblivious decision tree learner. This algorithm is able to eliminate redundant features and learn faster than C4.5.

Aha and Blankert [32] uses beam search on IB1 (a nearest neighbor classifier) for datasets with many features and reports improved performance. Domingos [33] describes an algorithm RC, a context sensitive instance-based wrapper approach of learning that uses backward elimination to select subset of features and cross-validation to measure accuracy. Kohavi [24] devised wrapper approach Decision Table Majority (DTM) classifier also performs significantly well compared to C4.5 learners. [22, 31] use Naïve Bayes and Bayesian Networks to select feature subset and report improved performance with smaller feature subset.

2.5 Class Imbalance Problem

Even though there are a significant number of classification algorithms prevalent in data mining and machine learning literature most of them fail to achieve

acceptable performance standard when the data is imbalanced. Imbalanced class distribution is quite common in many real world situations. Number of fraudulent credit card transactions is very few compared to legitimate transactions; the number of defective products in a manufacturing organization may be significantly lower than non-defective ones. Especially, in microarray expression analysis we frequently face data sets with imbalance class distribution as the process of data collection is complex, expensive, time-consuming and sensitive to patient privacy.

The performance measures that are common to measure the performance of classifiers may not be suitable and may contain less information when measuring the performance for the same classifiers on imbalanced data. For example, total accuracy rate and total error rate are two common performance measures for classification algorithms. But for imbalanced data, this is not the case. Say, for example, a test data set contains 100 examples out of which 99 examples are negative (majority class) and 1 example is positive (minority class). If a classification model is able to identify all the 99 majority examples from the test data but fails to identify the minority one, it will still achieve 99% accuracy and 1% error rate. If a researcher selects accuracy rate as the main performance criteria then he may end up accepting this model for the problem at hand. But for many real world applications identifying the minority example is more important than achieving an accuracy threshold as may be the case in credit card fraud

detection, defective product detection in product line assembly or identifying the diseased patient (positive example) from non-diseased ones (negative examples).

In two class problem, the class with fewer examples in training data is denoted as positive class or the minority class and class with majority class distribution is called negative class. For two class problem with imbalanced data sets, True Positive Rate (TPR) is taken as most significant parameter for performance measurement of the classification model. As it tells what is the fraction of the minority class in the test data that the model is able to identify accurately. True Negative Rate (TNR) of the model tells the fraction of the majority class examples that the model identifies accurately. False Positive Rate (FPR) and False Negative Rate (FNR) are other two parameters of interest for imbalanced data sets classification. FPR is the number of negative (majority) examples that were predicted as positive examples out of total number of negative examples. Similarly, FNR is the fraction of positive class examples that were predicted to be negative class by the classification model. The equations to calculate, the above four parameters are summarized below,

$$TPR = TP / (TP + FN) \quad (2.6)$$

$$TNR = TN / (TN + FP) \quad (2.7)$$

$$FPR = FP / (FP + TN) \quad (2.8)$$

$$FNR = FN / (FN + TP) \quad (2.9)$$

The above four parameters are also used to calculate the overall accuracy rate and balanced error ratio for the model which we will elaborate later. These two parameters also give important performance characteristics for the model.

Receiver Operating Curve (ROC) graphically compares TPR and FPR of a classifier and can be used as a performance parameter for imbalanced data set classifier. TPR is plotted along y-axis and FPR is plotted along x-axis of a ROC curve. Some critical points in ROC curve have important characteristics. When $TPR=0, FPR=0$, the model predicts every instance as a negative class, when $TPR=1, FPR=1$, the model predicts every instance as a positive class, when $TPR=1, FPR=0$, the model predicts all positive instances as positive which is the ideal case. ROC is particularly helpful for relative comparison of different classification algorithms.

Another widely used technique handles imbalanced datasets by modifying the datasets either by undersampling or oversampling. In undersampling, the data set is modified so that a subset of majority examples, equal to number of minority examples, is randomly chosen. In oversampling, the minority examples are replicated until the dataset becomes balanced.

2.6 *Related Work on Gene Selection with Imbalanced Microarray Data*

Out of a large number of works on feature subset selection only a few has addressed feature selection on dataset with imbalanced class distribution. Feature Assessment by Sliding Threshold (FAST) is a very recent algorithm proposed by Xue and Michael in August, 2008 KDD conference which spotlights on improving feature selection for Imbalanced datasets [40]. It uses area under a ROC curve as a means to rank features. The features are scored in the range 0.5 to 1.0, where a score close to 0.5 indicates that the feature is irrelevant to the classification and a score close to 1.0 indicates higher possibility of that feature to determine the class of the test example.

Embedded Gene Selection for EasyEnsemble (EGSEE) and Embedded Gene Selection for Individuals for EasyEnsemble (EGSIEE) are two algorithms that use Prediction Risk R_i to determine the goodness of i -th feature. The features with least value of R_i are removed from the final feature subset [41].

$$R_i = AUC - AUC(\bar{x}^i) \quad (2.10)$$

Both these embedded gene selection methods use EasyEnsemble classifier [42] to generate Ensemble model that is used to determine the Area Under Curve (AUC). The Fuzzy-Granular Gene Selection by Yuanchen He et. al. groups genes in different function granules utilizing Fuzzy C -Means algorithm (FCM) [43]. Basically, this process categorizes genes into three classes, Informative genes, Redundant genes and Irrelevant genes. The algorithm then eliminates redundant

and irrelevant genes and selects informative genes using Signal to Noise metric (S2N).

Chapter 3

FOUNDATION OF THE THESIS AND PROPOSED SOLUTIONS

3.1 *The ReliefF Filtering Algorithm*

The Relief algorithm is one of the very successful filtering algorithms to date. Most of the filtering methods developed in data mining and machine learning assume conditional independence among the attributes. But microarray data analysis involves an exceedingly higher number of attributes or genes with significantly lesser number of instances or examples. According to microbiologists a set of genes form a network and interact together to accomplish the biological tasks. So, filtering algorithms that assume conditional independence of the attributes are not suitable for microarray expression analysis.

The Relief algorithms are very useful as they do not make the assumption that the attributes are independent of each other. These algorithms are context sensitive and can correctly estimate the quality of attributes in problems with strong

dependencies among the attributes [1]. This makes Relief algorithms one of the most successful preprocessing algorithms [34]. Besides being used as a preprocessing algorithm it can be used to select splits at intermediate nodes in decision tree learning [35], in inductive logic programming [36], to guide the constructive induction in learning of regression trees [37].

The first version of Relief algorithm [38] was able to deal only with binary classification problems. It could estimate the rank of an attribute whether that attribute is nominal or numerical but it could not deal with missing values of attributes in the dataset. Later in his 1994 paper [39], Igor Kononenko proposed an algorithm, an extension of original Relief algorithm, called ReliefF that can deal with multi-class classification problem and missing attribute values in the dataset. The pseudo code for ReliefF is shown in figure 3.1.1.

Algorithm ReliefF:

Input: for each training instance a vector of attribute values and the class value

Output: the vector W of estimations of the qualities of attributes

1. set all weights $W[A] = 0.0$;
2. **FOR** $i=1$ **to** m **DO**
3. randomly select an instance R_i ;
4. find k nearest hits H_j ;
5. **FOR** each class $C \neq class(R_i)$ **DO**
6. from class C find k nearest misses $M_j(C)$;

7. **END FOR;**
8. **FOR** A=1 **to a DO**
9. $W[A] = W[A] - \sum_{j=1}^k \text{diff}(A, R_i, H_j) / (m \cdot k) +$
10. $\sum_{C \neq \text{class}(R_i)} \left[\frac{P(C)}{1 - P(\text{class}(R_i))} \sum_{j=1}^k \text{diff}(A, R_i, M_j(C)) \right] / (m \cdot k);$
11. **END FOR;**
12. **END FOR;**

Figure 3.1.1: The ReliefF algorithm

Let the dataset is represented by the instances I_1, I_2, \dots, I_n and the vector of attributes $A_i, i=1, 2, \dots, a$ where a is the total number of attributes present in the dataset. W is the vector that holds the quality estimation of the attributes. The algorithm starts by setting the weights for all the attributes to zero in line 1 of figure 3.1.1. It then selects an instance R_i randomly from the instance space. For that R_i , it then finds k nearest number of instances from the same class as R_i called Hits and denoted by H_j and k nearest number of instances for each class other than the class of R_i , called Misses and denoted by $M_j(C)$. In line 7 to 9, the algorithm updates the weight for each attribute for that random sample R_i . The algorithm repeats this whole process for m times where m is a user defined parameter and can be set to number of instances in the dataset.

The way that the algorithm updates the weight for each attribute is simple and intuitive. For each attribute in randomly selected sample R_i , if that attribute has a

value different for that same attribute for a sample in H_j , then the algorithm takes this as not desirable and subtracts the distance from the weight of that attribute.

The distance measures for nominal and numerical attributes are as follows,

$$diff(A, I_1, I_2) = \begin{cases} 0; & value(A, I_1) = value(A, I_2) \\ 1; & otherwise \end{cases} \quad (3.1)$$

$$diff(A, I_1, I_2) = \frac{|value(A, I_1) - value(A, I_2)|}{\max(A) - \min(A)} \quad (3.2)$$

Similarly, for all the k -misses in $M_j(C)$, the algorithm first takes one instance from the $M_j(C)$ and checks whether an attribute has different value in R_i and $M_j(C)$. If the value of the attribute is different in R_i and $M_j(C)$, the algorithm decides this situation as desirable and calculates the distance for that attribute and adds that distance to the weight for that attribute. The distances are always normalized so that the weights for the attributes are always in the same range and comparable to each other. The contribution for each class of the misses is multiplied by the prior probability of that class $P(C)$ to keep the weight in the range $[0, 1]$ and symmetric $[0, -1]$. The selection of k -hits and k -misses for each randomly selected sample gives robustness and more stability for ReliefF than original Relief algorithm. The process of updating the weight for each attribute can be stated in the following way,

$$W[A] = Pr(diff. value of A | nearest inst. from diff. class) - Pr(diff. value of A | nearest inst. from same class) \quad (3.3)$$

Here Pr denotes the conditional probability. Simply speaking, if both instances are from different class and their attribute values do not match then the algorithm increases the weight and if both instances are from same class but their attribute values are different then the algorithm decreases the weight.

In case of missing values the ReliefF algorithm modifies the ‘diff’ function to calculate the distance for a particular attribute. It calculates the probability that two given instances (I_1, I_2) have different attribute values over the class value. For example if I_1 has missing value but I_2 does not then,

$$diff(A, I_1, I_2) = 1 - Pr(value(A, I_2) | class(I_1)) \quad (3.4)$$

But if both I_1 and I_2 have missing values then the distance for attribute A is,

$$diff(A, I_1, I_2) = 1 - \sum_V^{\#values(A)} (Pr(V | class(I_1)) \times Pr(V | class(I_2))) \quad (3.5)$$

3.2 Higher Weight ReliefF

The algorithm ReliefF is an instance-based filtering method. It means that the ability to estimate the quality of attributes by ReliefF is highly dependent on the number of instances from different classes in the dataset. Even though, the algorithm is dependent on class distribution of instances, it does not consider the class distribution while ranking the attributes. For balanced datasets this filtering procedure works fine. However, the performance degrades significantly when the dataset is imbalanced or biased towards any specific class. As microarray

expression analysis often provides researchers with imbalanced dataset, it is highly recommended that any filtering approach we use should consider whether the data is imbalance or not and provide a way to still show acceptable performance for imbalanced dataset.

To compensate the degradation due to data imbalance we propose a method for calculating the weight for each attribute. In this proposed version of ReliefF which we termed as Higher Weight ReliefF, we first determine the class distribution in the training data. Then while updating the weight for each attribute, if the randomly selected instance R_i is from minority class, we put higher weight by using the following equation,

$$W[A] = W[A] + \sum_{C \neq \text{class}(R_i)} \left[\left\{ \frac{P(C)}{1 - P(\text{class}(R_i))} \right\} \sum_{j=1}^k \text{diff}(A, R_i, M_j(C)) \left(1 + \left(1 - \frac{\text{minor}}{\text{total}} \right) \right) \right] \quad (3.6)$$

However, if R_i is from majority class, we keep the original weight function to estimate the weight for the attributes. In this way, we are able to handle the minority class more effectively than the original ReliefF algorithm.

The proposed Higher Weight ReliefF filtering method is stated below,

Algorithm Higher –Weigh –ReliefF :

Input: for each training instance a vector of attribute values and the class value

Output: the vector W of estimations of the qualities of attributes

1. set all weights $W[A] = 0.0$;
2. find class distribution in the training dataset, set minor and total count;

3. **FOR** $i=1$ **TO** m **DO**
4. randomly select an instance R_i ;
5. find k nearest hits H_j ;
6. **FOR** each class $C \neq \text{class}(R_i)$ **DO**
7. from class C find k nearest misses $M_j(C)$;
8. **END FOR**;
9. **FOR** $A=1$ **TO** a **DO**
10.
$$W[A] = W[A] - \sum_{j=1}^k \text{diff}(A, R_i, H_j) / (m \cdot k) +$$
11.
$$\sum_{C \neq \text{class}(R_i)} \left[\frac{P(C)}{1 - P(\text{class}(R_i))} \sum_{j=1}^k \text{diff}(A, R_i, M_j(C)) \left(1 + \left(1 - \frac{\text{minor}}{\text{total}} \right) \right) \right] / (m \cdot k);$$
12. **END FOR**;
13. **END FOR**;
14. **END**;

Figure 3.2.1: The Higher Weight ReliefF algorithm

3.3 ReliefF with Differential Minority Repeat

The second method that we propose is ReliefF with Differential Minority Repeat. In this method, we modify the dataset in such a way that it becomes balanced, so that the minority class examples are not neglected while calculating the weight for the attributes. Let S is the entire instance space consisting of N number of instances. We split the instance space S into S_{minority} and S_{majority} instance space

consisting of only minority and majority examples where $S_{minority}$ contains $N_{minority}$ number of instances and $S_{majority}$ contains $N_{majority}$ number of instances. Let, D_n is the difference between the number of majority examples, $N_{majority}$ and minority examples $N_{minority}$ such that,

$$D_n = N_{majority} - N_{minority} \quad (3.7)$$

We modify the minority instance space by randomly selecting D_n instances from the minority instance space and adding these D_n minority examples along with the original minority instances to the new minority instance space $S'_{minority}$.

$$S'_{minority} = S_{minority} + D_n \text{ randomly selected minority examples from } S_{minority} \quad (3.8)$$

Then we merge these new minority dataset, $S'_{minority}$ and majority dataset, $S_{majority}$ to get our new Differential Minority Repeat Dataset, S' . We use this new dataset to estimate the quality of attributes using the ReliefF algorithm.

$$S' = S_{majority} + S'_{minority} \quad (3.9)$$

The Differential Minority Repeat process can be illustrated by an example. Let our training instance space consists of 100 examples out of which 30 examples are from minority class and 70 examples are from majority class. We calculate $D_n = 70 - 30$ or $D_n = 40$. Now, we randomly add 40 minority examples to the

minority instance set which gives $S'_{minority} = 70$ examples. Then we merge these two sets and get the new instance set as $S' = S_{majority} + S'_{minority} = 70 + 70 = 140$. So the new dataset S' consists of original majority examples and original and replicated minority examples. This procedure is further shown below in figure 3.3.1.

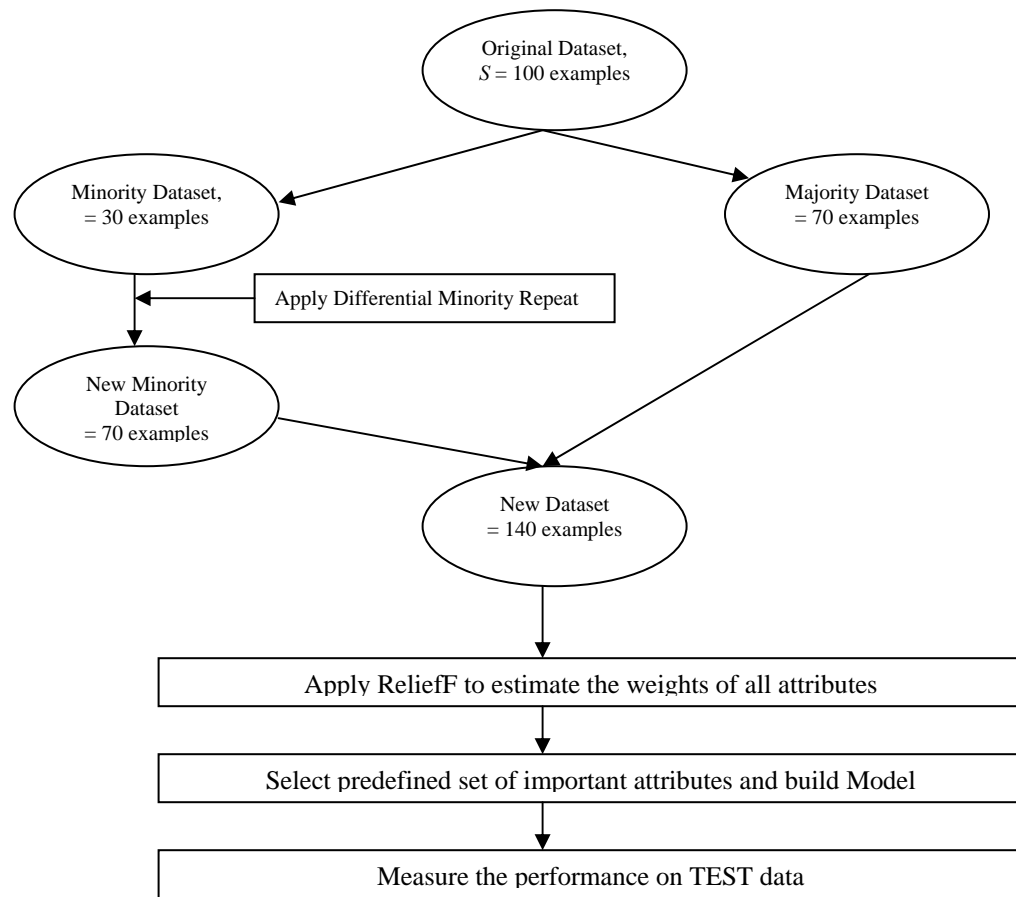


Figure 3.3.1: Differential Minority Repeat on ReliefF

The algorithm for Differential Minority Repeat is given below,

Algorithm ReliefF with Differential Minority Repeat :

Input: Imbalanced training dataset, each instance containing a vector of attributes and a class value

Output: the vector W of estimations of the qualities of attributes

1. *find class distribution in the training dataset, set $N_{majority}$ & $N_{minority}$;*

2. *set the difference count, $D_n = N_{majority} - N_{minority}$;*

3. *update the minority dataset,*

$S'_{minority} = S_{minority} + D_n$ *randomly selected minority examples from $S_{minority}$*

4. *use original ReliefF to estimate the weights of attributes, $W[A]$;*

5. *end;*

Figure 3.3.2: The Relief with Differential Minority Repeat algorithm

3.4 ReliefF with Balanced Minority Repeat

Our final proposed solution is ReliefF with Balanced Minority Repeat. Here, we split the entire instance space, S into minority and majority instance spaces, $S_{minority}$ and $S_{majority}$ where $S_{minority}$ contains $N_{minority}$ number of instances and $S_{majority}$ contains $N_{majority}$ number of instances. Then we create m datasets S'_1, S'_2, \dots, S'_m by combining the examples from $S_{minority}$ and $N_{minority}$ number of examples from $S_{majority}$. For this we split the $S_{majority}$ dataset into m equal parts. Now, m can be set as the following equation,

$$m = \text{ceil}\left(\frac{N_{majority}}{N_{minority}}\right) \quad (3.10)$$

So, each dataset S'_1, S'_2, \dots, S'_m contains examples as shown below,

$$S'_i = S_{minority} + N_{minority} \text{ number of instances from instance space } S_{majority} \quad (3.11)$$

If the final split of dataset S'_m contains majority examples less than number of minority instances then we select majority examples from $S'' = S'_1, S'_2, \dots, S'_{m-1}$ to make it balanced. So, our final dataset consists of m balanced datasets of equal number of minority and majority examples.

$$S' = S'_1 + S'_2 + \dots + S'_m \quad (3.12)$$

The algorithm for Balanced Minority Repeat is illustrated in figure 3.4.1.

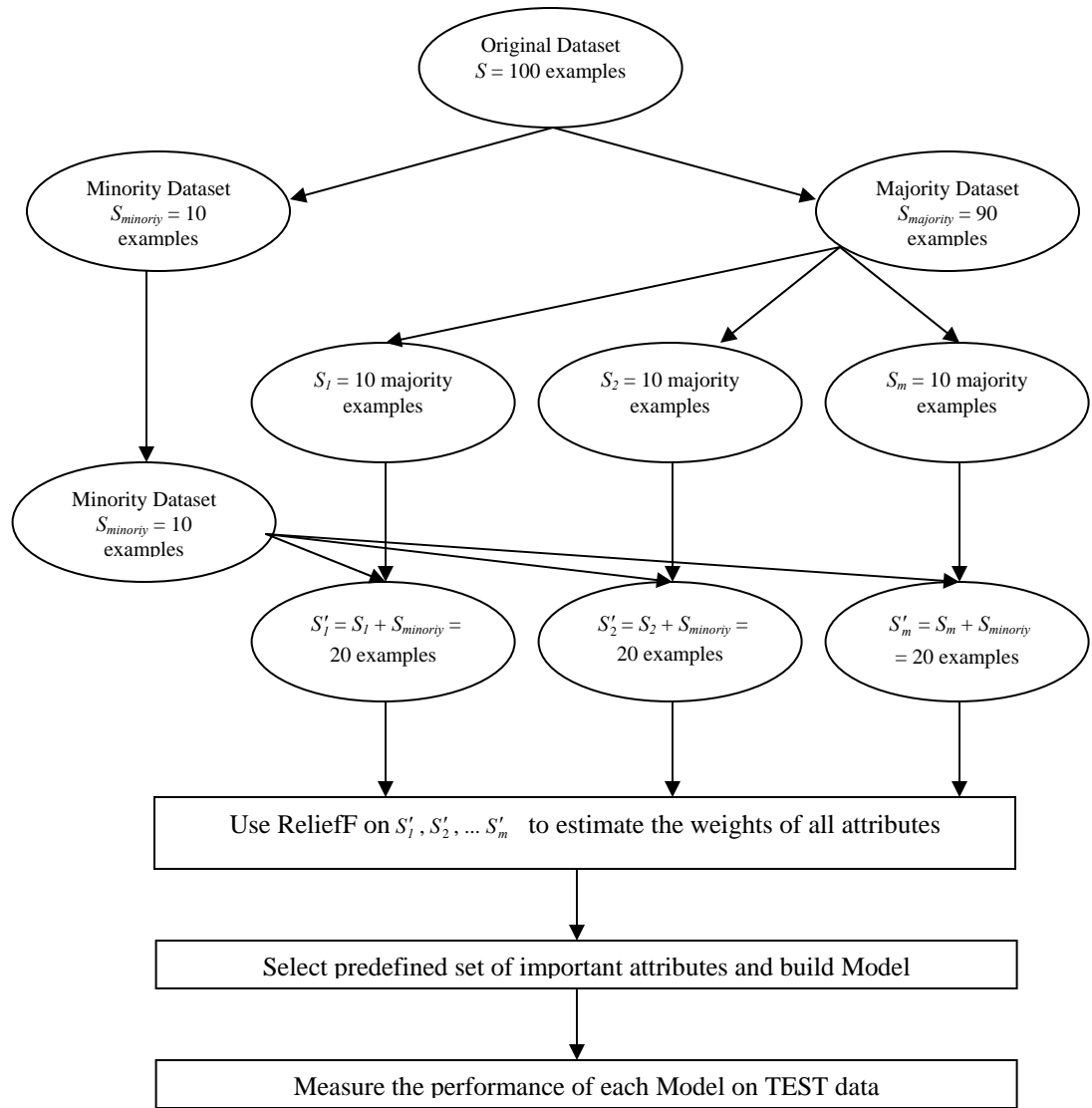


Figure 3.4.1: Balanced Minority Repeat with 10 minority and 90 majority examples

In the following figure we summarize this algorithm,

Algorithm ReliefF with Balanced Minority Repeat :

Input: Imbalanced training dataset, each instance containing a vector of attributes and a class value

Output: the vector W of estimations of the qualities of attributes

1. split the training set S into $S_{minority}$ and $S_{majority}$;
2. split $S_{majority}$ into S_1, S_2, \dots, S_m individual datasets where $m = \text{ceil}\left(\frac{N_{majority}}{N_{minority}}\right)$;
3. combine $S_{minority}$ and each dataset from S_1, S_2, \dots, S_m to get a set of balanced datasets, $S'_i = S_{minority} + S_i$, for $i = 1$ to m
4. use original ReliefF on $S' = S'_1 + S'_2 + \dots + S'_m$ to estimate the weights of attributes, $W[A]$;
5. end;

Figure 3.4.2: The Relief with Balanced Minority Repeat algorithm

Chapter 4

EXPERIMENTAL SETTINGS AND ANALYSIS OF RESULTS

4.1 *The Microarray Datasets of Experiments*

The proposed filtering techniques are evaluated on five microarray expression datasets; they are Colon dataset, Central Nervous system dataset, DLBCL Tumor dataset, Lymphoma dataset and finally ECML Pancreas dataset.

Table 4.1.1: The Microarray Expression datasets for experiments

Database Name	No of Minority Examples	No of Majority Examples	No of Genes	Class Distribution	
				(%) Minority	(%) Majority
Colon	22	40	2000	35.48	64.52
Central Nervous System	21	39	7,129	35.00	65.00
DLBCL Tumor	19	58	7,129	24.68	75.32
Lymphoma	23	73	4,026	23.96	76.04
ECML Pancreas	8	82	27,679	8.89	91.11

In table 4.1.1, we summarize the class distribution for each dataset along with number of attributes or genes that are associated with that dataset.

4.2 WEKA

WEKA is very powerful and rich data mining software available for download free of cost from the site <http://www.cs.waikato.ac.nz/ml/weka/>. The standalone software provides tools for all kinds of data mining and machine learning tasks such as data pre-processing, classification, regression, clustering etc. Moreover, it is open source software written in java programming language. All the source code for different data mining and machine learning algorithms are accessible to the research community. So, researchers can easily use these algorithms and make necessary modifications suitable to any particular data mining problem. The resource for WEKA is continually growing as programmers and computer scientists specialized in data mining are always contributing by implementing any new data mining algorithms and making it available to all. The following figure shows a snapshot of simple experiment window in WEKA application,

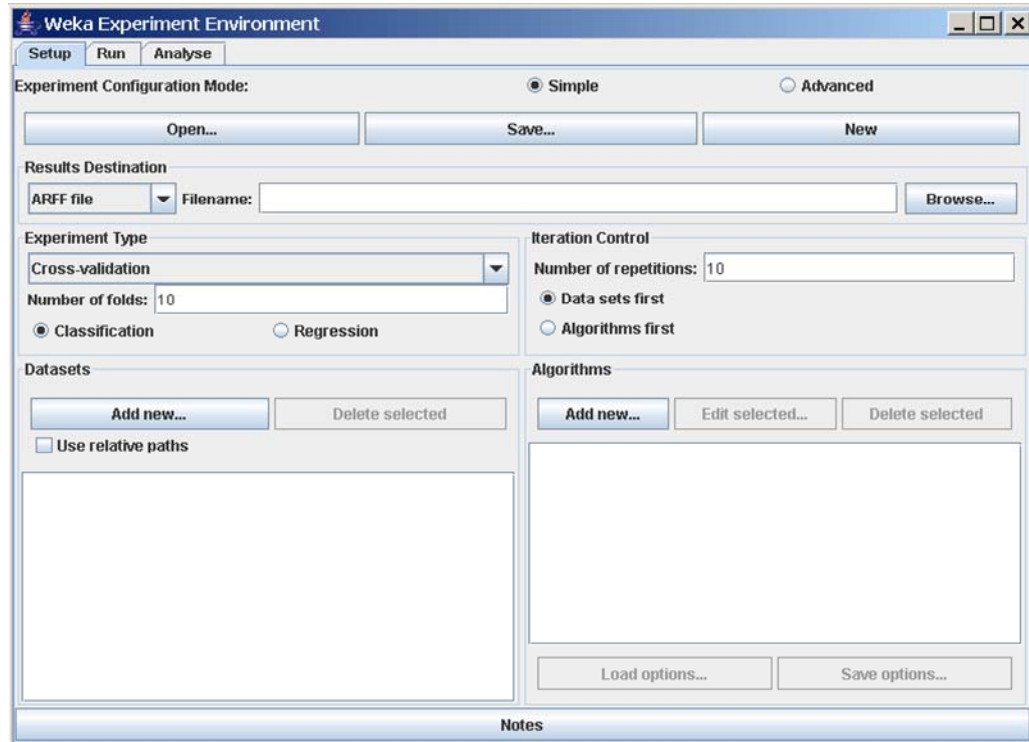


Figure 4.2.1: A simple WEKA experiment window

The file format that WEKA uses for input data mining problem is called ARFF file format. This format simply defines the names of the attributes and their types followed by set of examples. Each example is a vector of the attribute values. In our research, we have used the ReliefF pre-processing filtering algorithm whose source code is freely available with the software itself. However we have tuned the algorithm for the particular data mining problem that we are working on. The three algorithms that we propose are also written and tested in java where we use ECLIPSE as the IDE for development.

4.3 Performance Metrics

The evaluation of the performance of three filtering techniques is based on four performance metrics, True Positive Rate (TPR), True Negative Rate (TNR), Overall Accuracy Rate (Accuracy) and Balanced Error Ratio (BER). After ranking all the attributes using the proposed filtering algorithms and original ReliefF, we select the pre-defined set of most important attributes and build classification model using Naïve Bayes, Random Forest and IB1 classification algorithms. We calculate the four performance parameters from confusion matrices generated by each model. A sample confusion matrix is shown below,

Table 4.3.1: The Confusion Matrix of Classification

Actual Class from Test Data		Predicted Class by Classification Model	
		+ve	-ve
	+ve	True +ve (<i>TP</i>)	False -ve (<i>FN</i>)
	-ve	False +ve (<i>FP</i>)	True -ve (<i>TN</i>)

Legend: +ve = Positive Class (or Minority Class)

-ve = Negative Class (or Majority Class)

The equations to calculate the four parameters are described below,

$$TPR = \frac{TP}{TP + FN} \quad (4.1)$$

$$TNR = \frac{TN}{TN + FP} \quad (4.2)$$

$$Accuracy = \frac{TP + TN}{Total\ No\ of\ Instances\ in\ Test\ Data} \quad (4.3)$$

$$BER = \frac{1}{2} \times \left[\frac{FP}{FP + TP} + \frac{FN}{FN + TN} \right] \quad (4.4)$$

4.4 Analysis of performance on Colon and Central Nervous System Datasets

The four performance parameters of five-times ten-fold cross validation on five popular microarray expression data with imbalanced class distribution using all four filtering methods, Higher Order ReliefF, ReliefF with Differential Minority Repeat, ReliefF with Balanced Minority Repeat and original ReliefF are reported and discussed. We use three classification methods, Naïve Bayes Classifier, Random Forest classifier and IB1 nearest-neighbor classifier. In total we have 60 tables containing the performance data (5 dataset * 4 performance metrics * 3 classifier = 5*4*3 = 60 tables). Each table contains performance data for all four filtering techniques. All the 60 tables containing the performance metrics data are listed in Appendix A.

As mentioned earlier, both Colon and Central Nervous System datasets have almost identical class distribution, with Colon 35.48% minority and 64.52% majority class examples whereas Central Nervous System has 35% minority and 65% majority class examples. So we chose to discuss the performance of the three

proposed filtering techniques on these two datasets in the same section as the results are almost identical for these two datasets.

The following figure (Figure: 4.4.1) shows the True Positive Rates (TPR) for all four filtering techniques using Naïve Bayes classifier on Central Nervous System data.

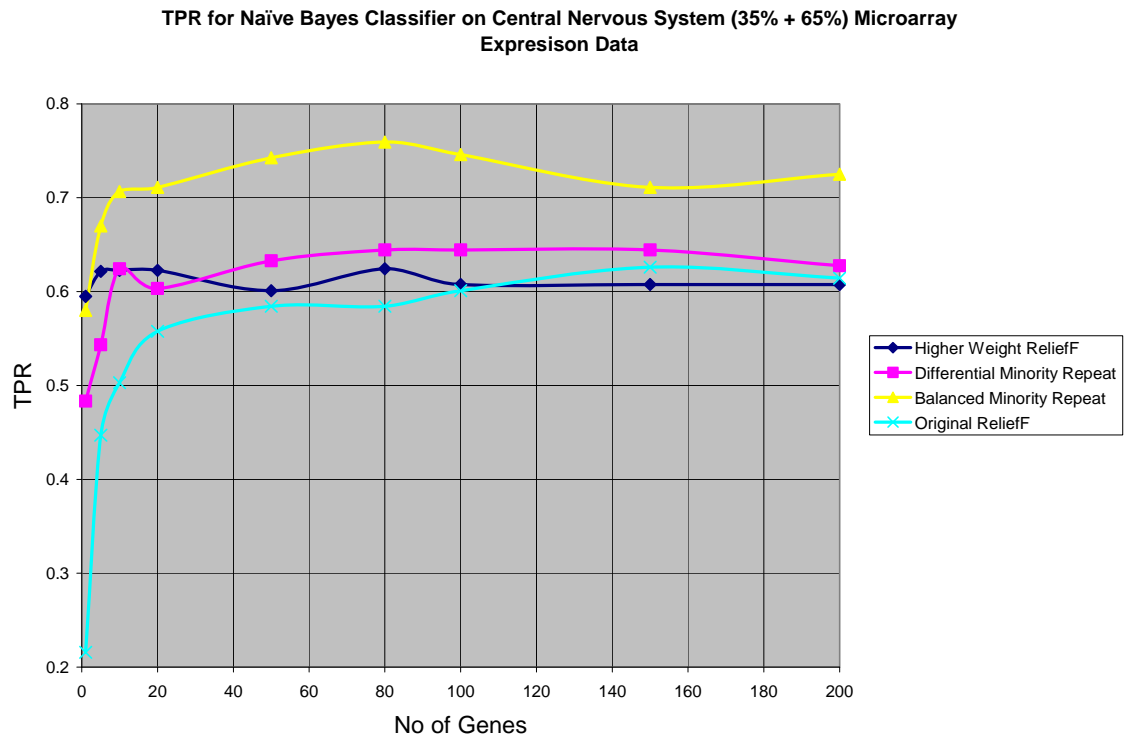


Figure 4.4.1: TPR for Naïve Bayes Classifier on Central Nervous System Data

Next, we change the classification algorithm to Random Forest (RF) and IB1 and plotted the TPR vs. No of Genes on the same dataset. The following two figures (Figure 4.4.2 & 4.4.3) show the performance of the four filtering methods in terms of TPR for RF and IB1 respectively.

TPR for Random Forest Classifier on Central Nervous System (35% + 65%) Microarray Expression Dataset

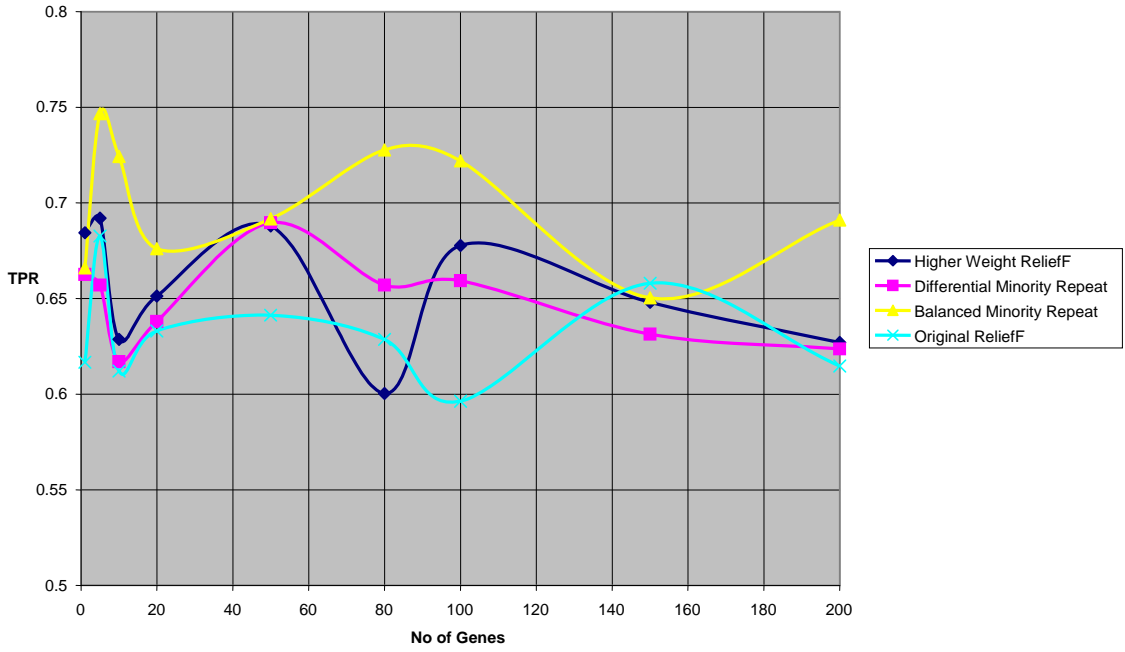


Figure 4.4.2: TPR for RF classifier on Central Nervous Data

TPR for IB1 Classifier on Central Nervous System (35% + 65%) Microarray Expression Dataset

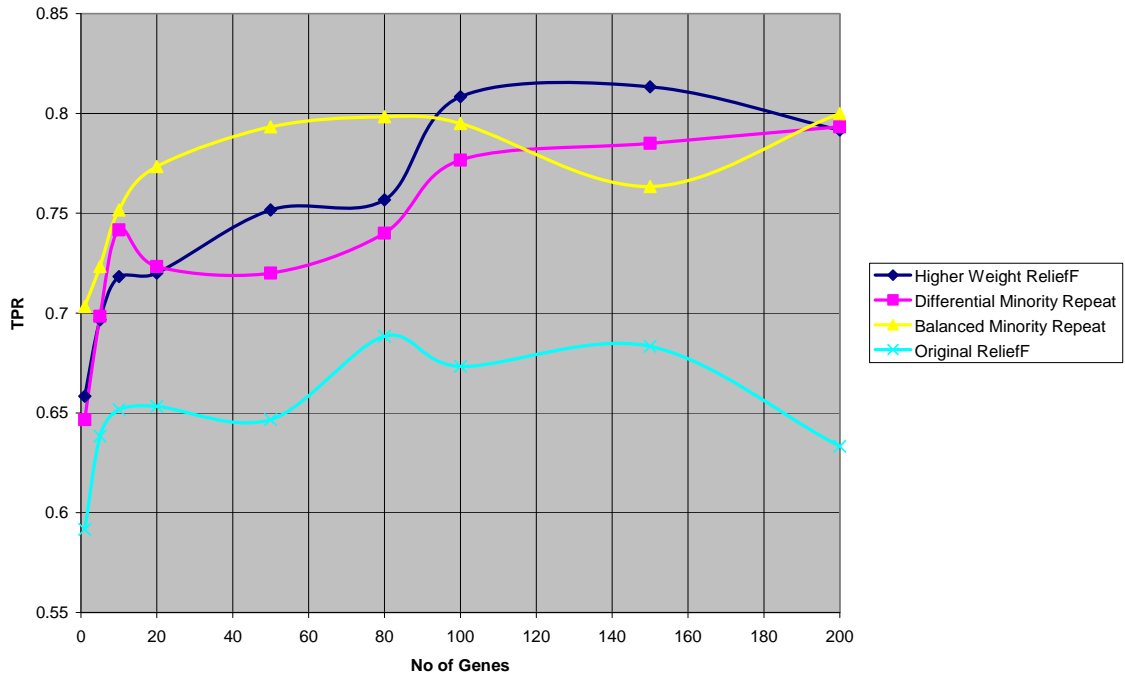


Figure 4.4.3: TPR for IB1 classifier on Central Nervous data

In the next three figures (Figure 4.4.4, 4.4.5 & 4.4.6), we plot Accuracy vs. No of Genes, TNR vs. No of Genes and BER vs. No of Genes for IB1 classifier on Central Nervous System Data. The tables containing data for all these four performance metrics using three classifiers on these two datasets ($4 \times 3 \times 2 = 24$ tables) can be found in Appendix A.

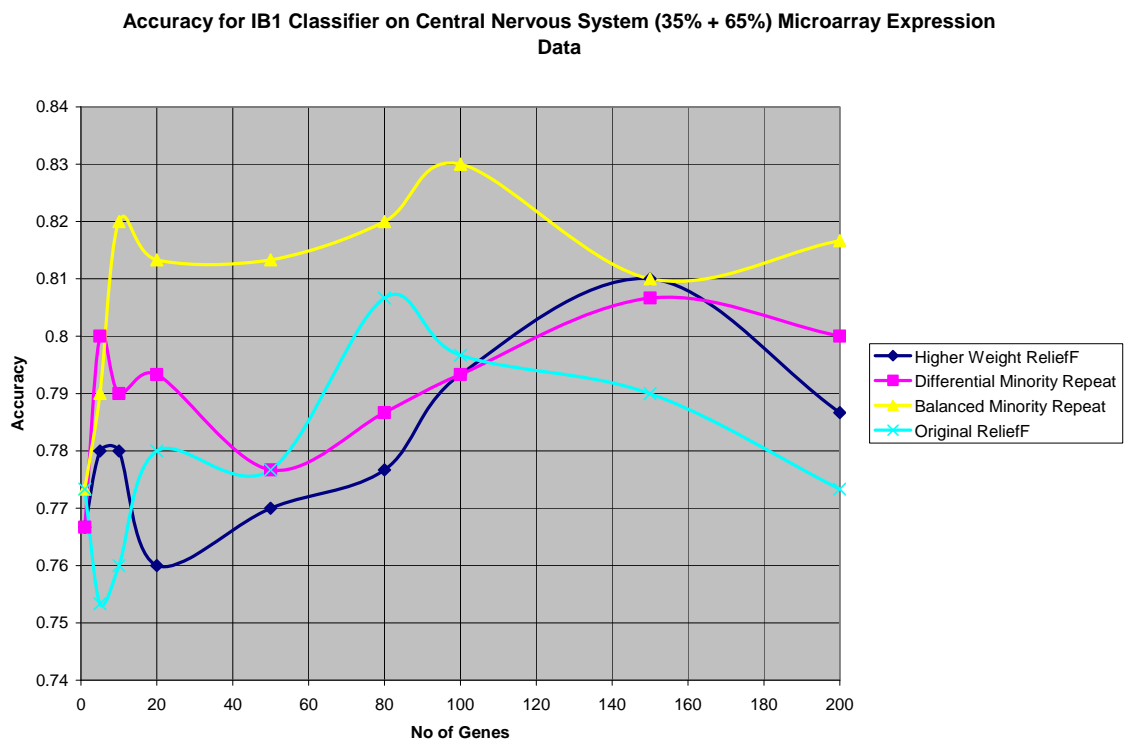


Figure 4.4.4: Accuracy for IB1 classifier on Central Nervous data

TNR for IB1 Classifier on Central Nervous System (35% + 65%) Microarray Expression Data

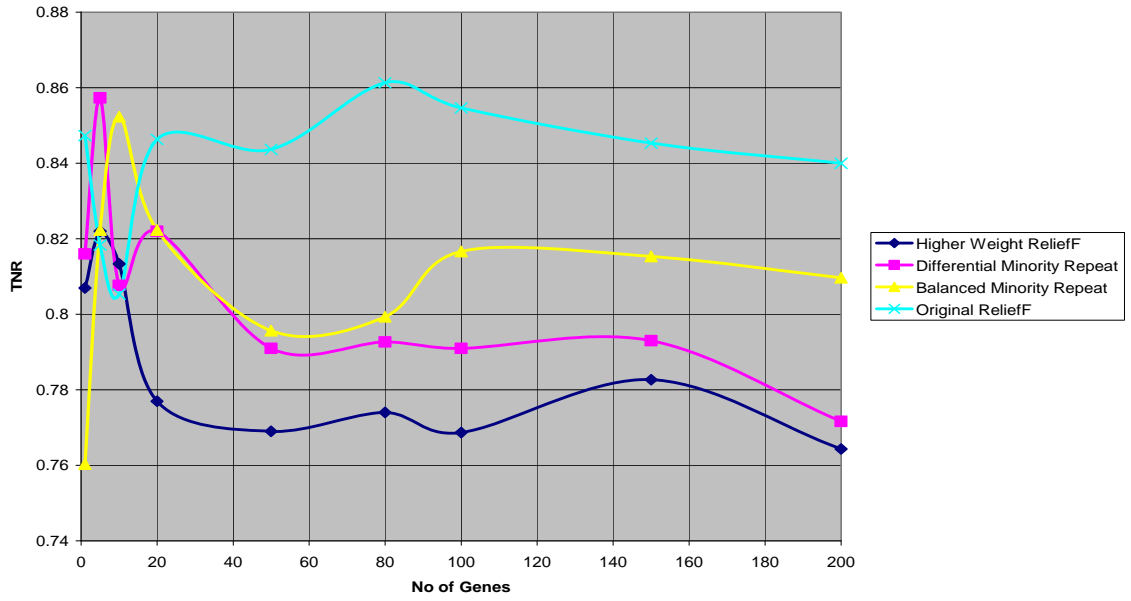


Figure 4.4.5: TNR for IB1 classifier on Central Nervous data

BER for IB1 Classifier on Central Nervous System (35% + 65%) Microarray Expression Data

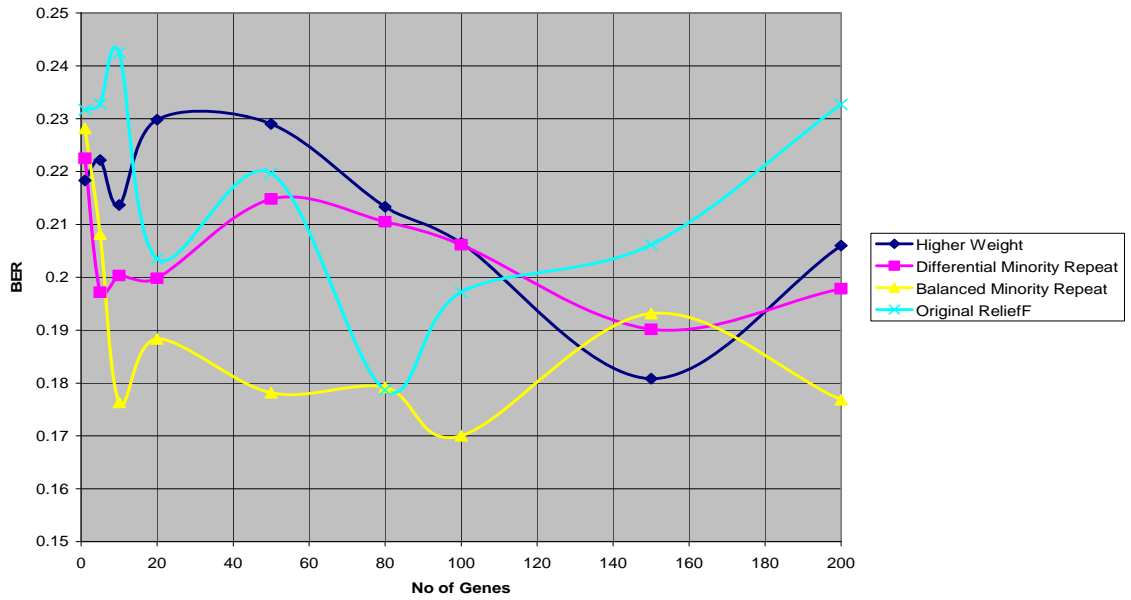


Figure 4.4.6: BER for IB1 classifier on Central Nervous data

4.5 Performance Evaluation on DLBCL Tumor and Lymphoma Microarray Expression Data

The microarray expression dataset DLBCL Tumor has class distribution of 24.68% minority and 75.32% majority class examples and Lymphoma dataset has 23.96% minority and 76.04% majority class examples. Not only both these datasets have identical class distribution but also they have similar performance characteristics. So, we restrict our analysis of performance to only Lymphoma dataset. However, interested readers can consult to the performance results on DLBCL Tumor dataset which, along with all other results, are included in appendix A.

In figure 4.5.1 we plot the TPR for Naïve Bayes classifier on Lymphoma dataset. Next, we plot TPR for Random Forest and IB1 classifier in figures 4.5.2 and 4.5.3 respectively.

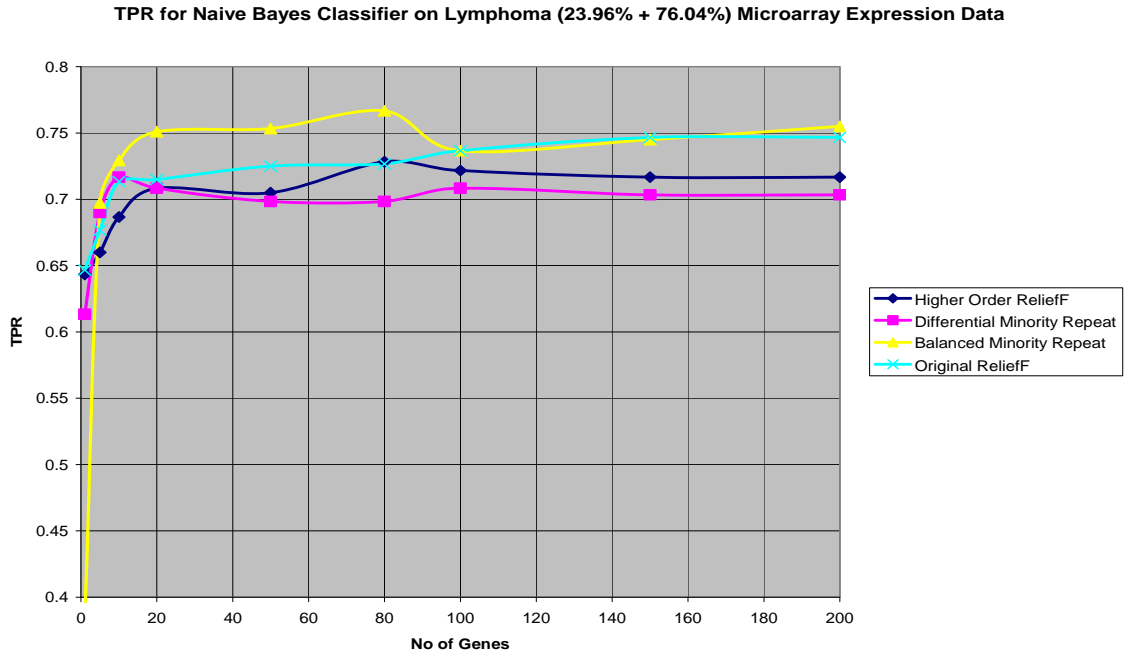


Figure 4.5.1: TPR for Naïve Bayes classifier on Lymphoma dataset

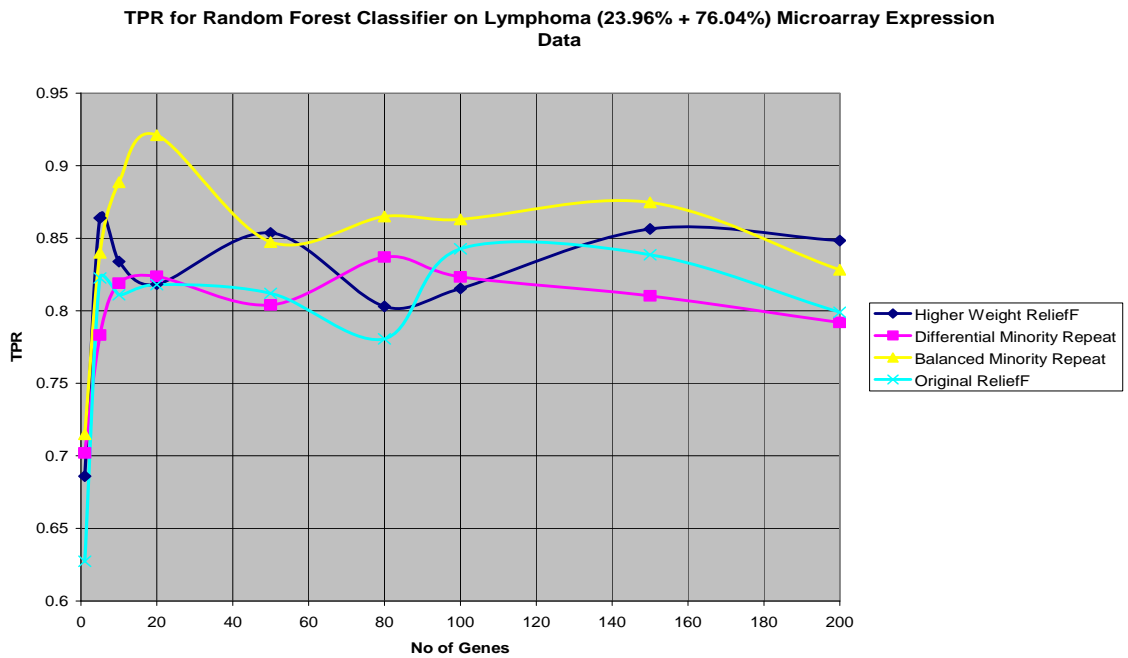


Figure 4.5.2: TPR for Random Forest classifier on Lymphoma data

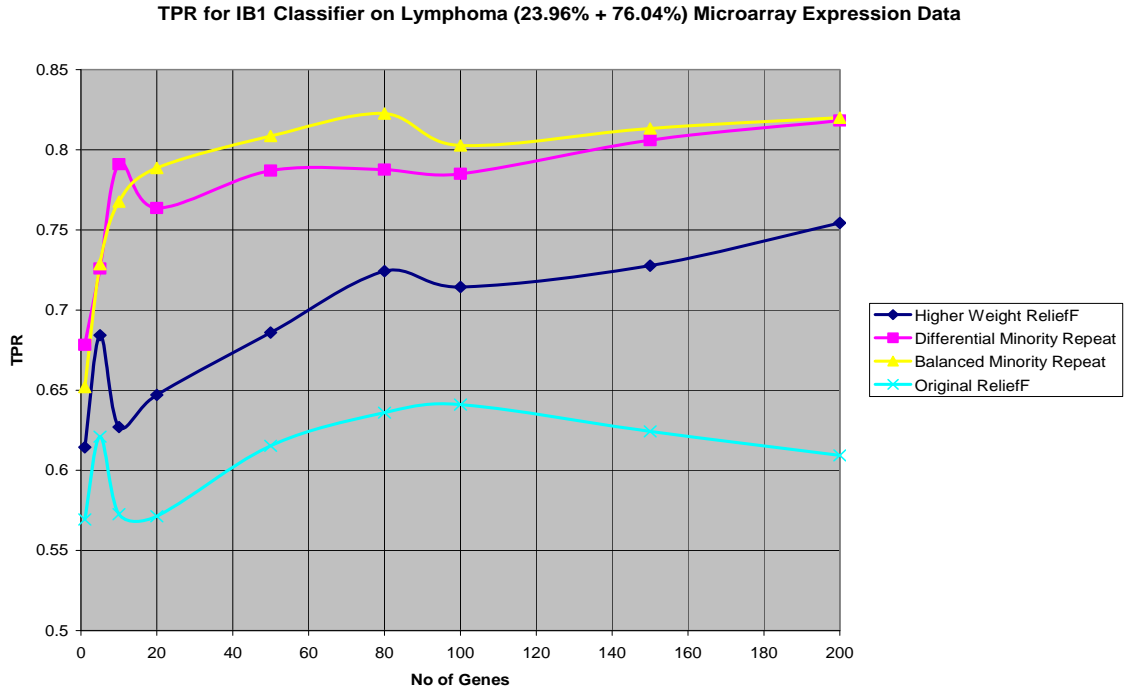


Figure 4.5.3: TPR for IB1 classifier on Lymphoma data

The TPRs using these three classifiers on Lymphoma dataset clearly prove that ReliefF using Balanced Minority Repeat is significantly better than Higher Order ReliefF, ReliefF using Differential Minority Repeat and Original ReliefF. Next we plot the Accuracy, TNR and BER using Naïve Bayes classifier on Lymphoma dataset. The results for these performance metrics using Random Forest and IB1 are listed in Appendix A.

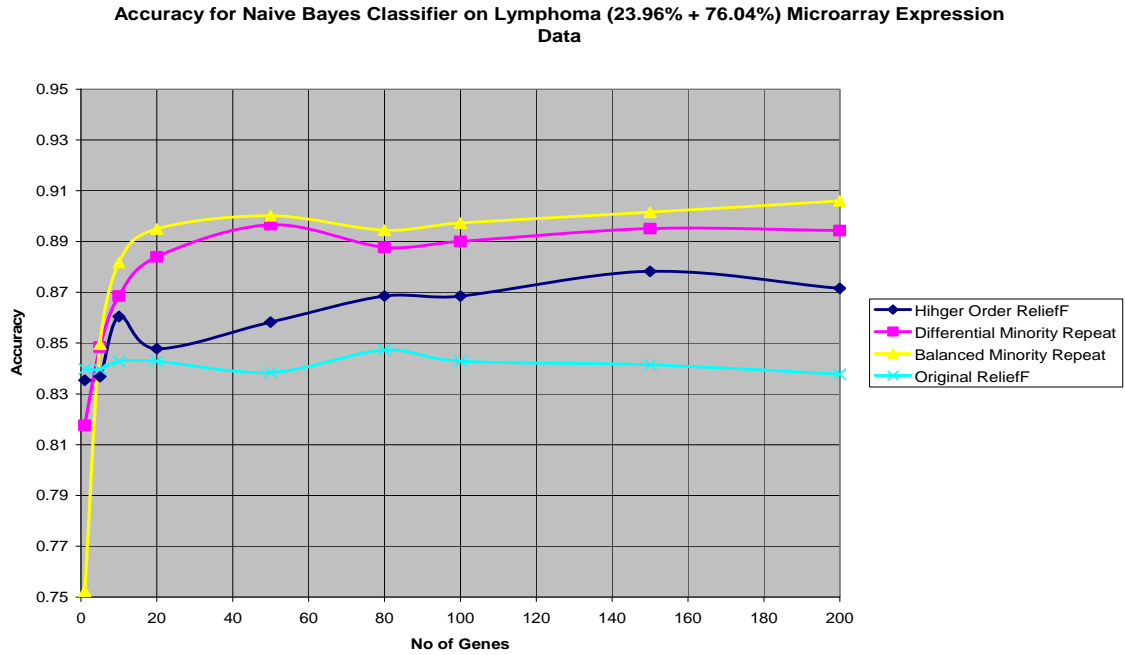


Figure 4.5.4: Accuracy for Naïve Bayes classifier on Lymphoma dataset

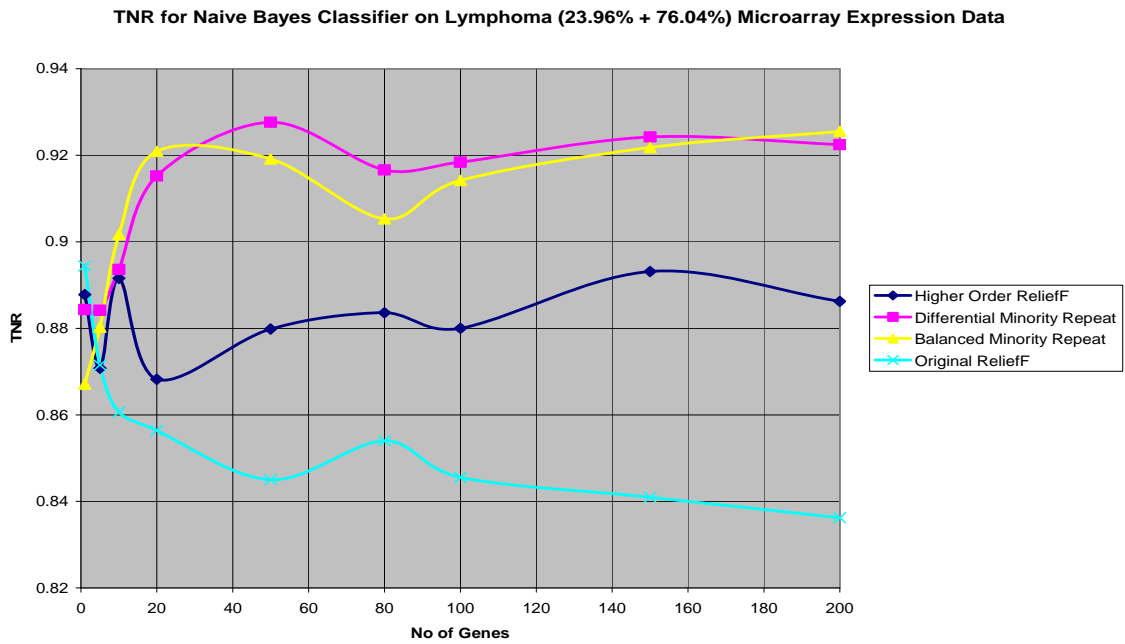


Figure 4.5.5: TNR for Naïve Bayes classifier on Lymphoma dataset

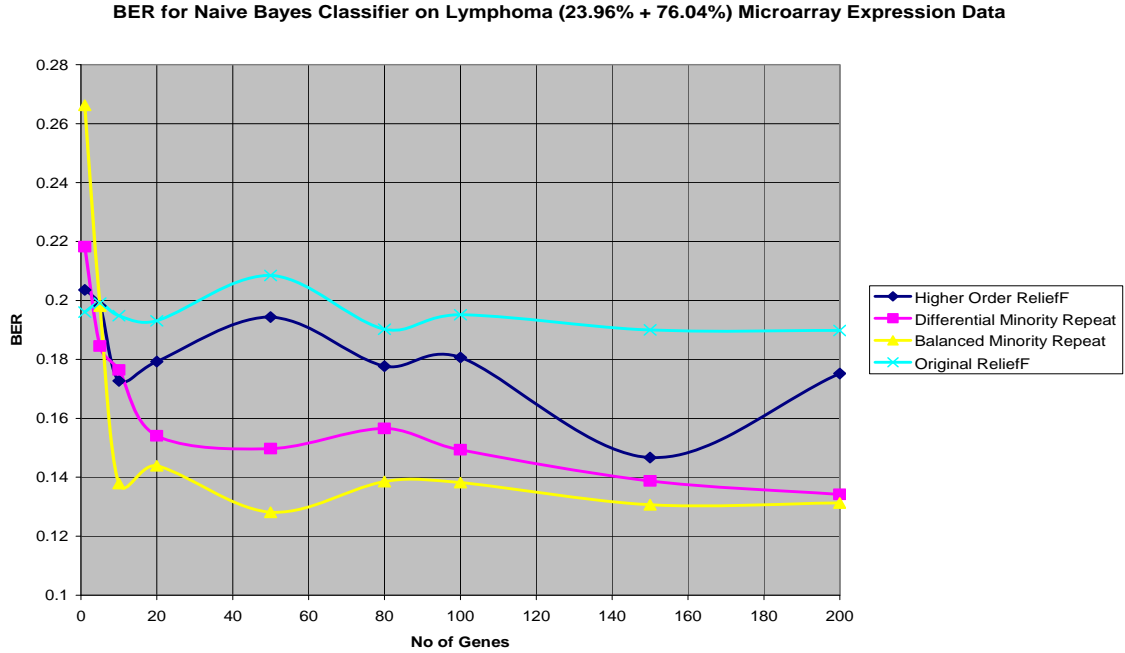


Figure 4.5.6: BER for Naïve Bayes classifier on Lymphoma data

4.6 Performance Evaluation on Pancreas Dataset

Finally, we evaluate the proposed filtering techniques on Pancreas microarray expression dataset which is the most unbalanced dataset out of five datasets. This dataset has a minority class distribution of 8.89% and majority class distribution of 91.11% with 27,679 reported genes in the dataset. This extremely high dimensionality of this dataset along with severe imbalance poses considerable challenge for any filtering techniques. However, the filtering techniques that we propose show substantial improvement with regard to original ReliefF.

At first, we illustrate the performance of all the four algorithms in terms of TPR using the three classification model on this dataset.

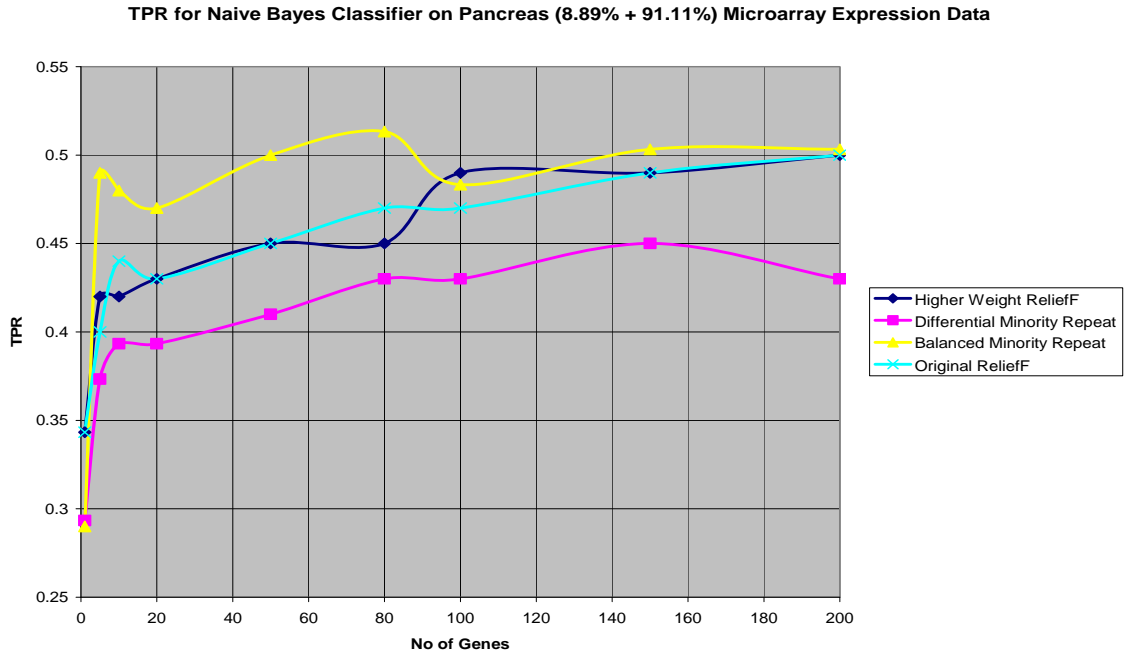


Figure 4.6.1: TPR for Naïve Bayes classifier on Pancreas data

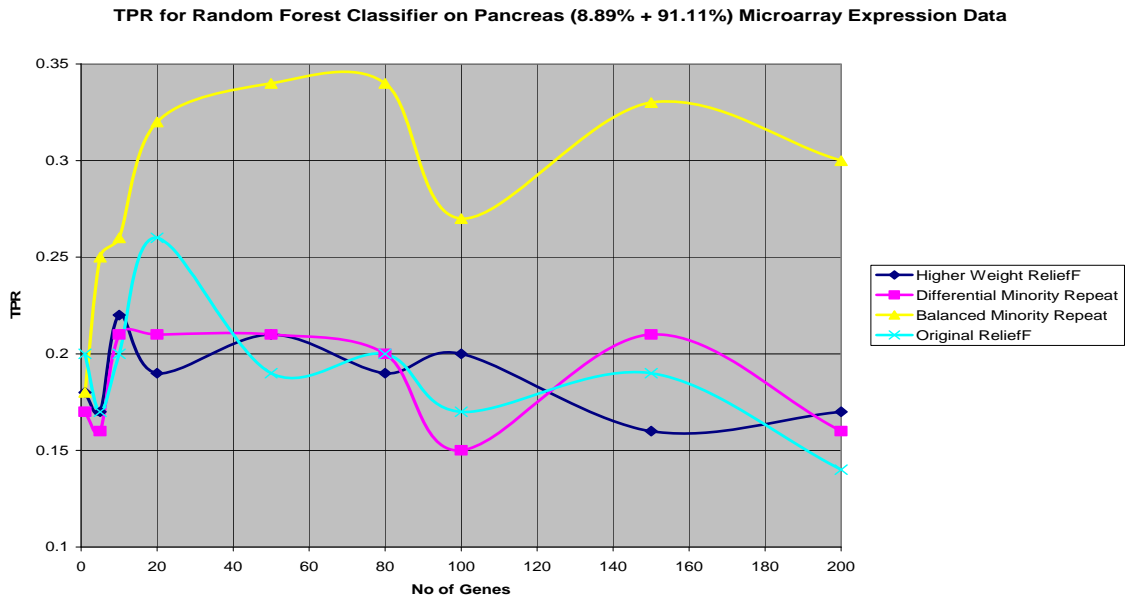


Figure 4.6.2: TPR for Random Forest classifier on Pancreas data

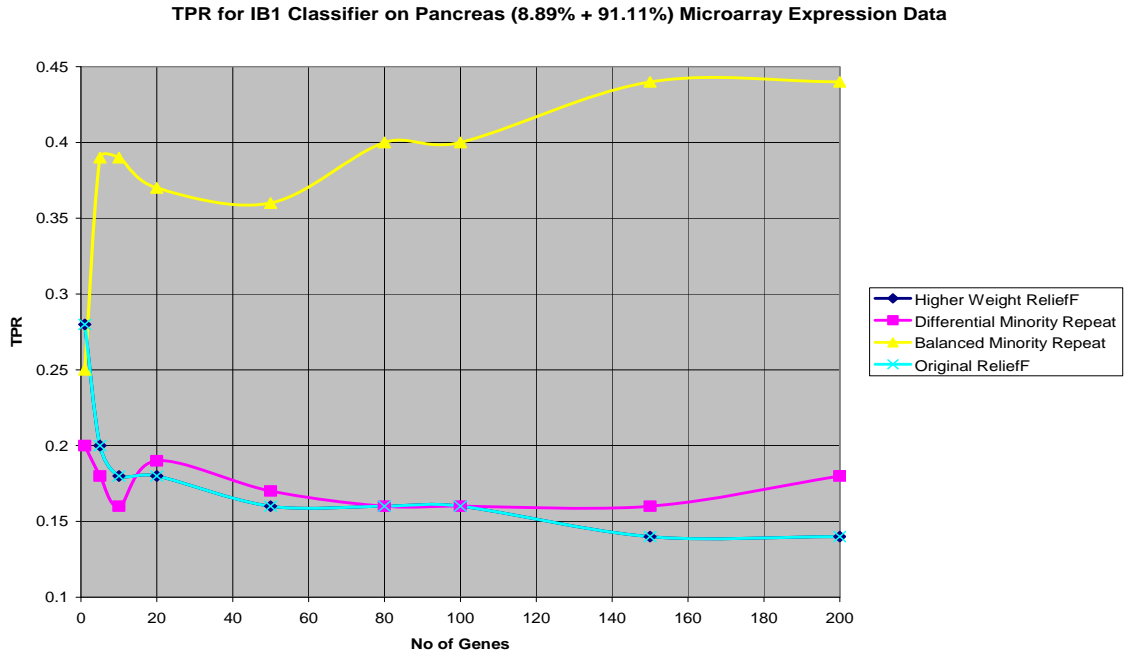


Figure 4.6.3: TPR for IB1 classifier on Pancreas data

The TPRs using all of these three different classifiers conform that ReliefF with Balanced Minority Repeat filtering approach is better than the two other proposed filtering algorithms and the original ReliefF filtering approach. Here we would like to include the fifth-run of 10-fold cross-validation data using IB1 classifier on Pancreas dataset. Interestingly, we observe that only Balanced Minority Repeat filtering approach is able to extract relevant and meaningful genes over the set of pre-selected 1, 5, 10, 20, 50, 80, 100, 150 and 200 most important genes.

Relation name: ECML				
Pancreas				
No of attr is 27680				
No of instances is 90				
Higher Weight ReliefF:				
Attr	Acc	TPR	TNR	BER
1	0.955556	0.3	0.9875	0.041667
5	0.955556	0.2	1	0.022222

10	0.944444	0.1	1	0.027778
20	0.933333	0	1	0.033333
50	0.933333	0	1	0.033333
80	0.933333	0	1	0.033333
100	0.933333	0	1	0.033333
150	0.933333	0	1	0.033333
200	0.933333	0	1	0.033333
Differential Minority Repeat:				
Attr	Acc	TPR	TNR	BER
1	0.966667	0.3	1	0.016667
5	0.933333	0	1	0.033333
10	0.933333	0	1	0.033333
20	0.933333	0	1	0.033333
50	0.933333	0	1	0.033333
80	0.933333	0	1	0.033333
100	0.933333	0	1	0.033333
150	0.933333	0	1	0.033333
200	0.933333	0	1	0.033333
Balanced Minority Repeat:				
Attr	Acc	TPR	TNR	BER
1	0.866667	0.2	0.911111	0.122222
5	0.944444	0.4	0.965278	0.086111
10	0.966667	0.4	0.988889	0.061111
20	0.944444	0.3	0.975	0.068254
50	0.966667	0.3	1	0.016667
80	0.966667	0.5	0.977778	0.055556
100	0.955556	0.5	0.965278	0.080556
150	0.955556	0.6	0.952778	0.083333
200	0.944444	0.6	0.941667	0.083333
Original ReliefF:				
Attr	Acc	TPR	TNR	BER
1	0.955556	0.3	0.9875	0.041667
5	0.955556	0.2	1	0.022222
10	0.944444	0.1	1	0.027778
20	0.933333	0	1	0.033333
50	0.933333	0	1	0.033333
80	0.933333	0	1	0.033333
100	0.933333	0	1	0.033333
150	0.933333	0	1	0.033333
200	0.933333	0	1	0.033333

Figure 4.6.4: The 5th run of 10-fold CV on Pancreas Data

Higher Weight ReliefF and Original ReliefF filtering approach using IB1 classifier are able to find 30%, 20% and 10% of minority classes from models built using 1, 5 and 10 most significant genes. The classification models built from 20, 50, 80, 100, 150 and 200 most significant genes are unable to detect any minority classes from the test datasets of 10-fold cross validations using these two filtering approach. The performance of Differential Minority Repeat for this fifth-run is worse than all other three which detects only 30% of minority classes from the classification model built using 1 most significant gene. All other models from the rest pre-selected genes fail miserably.

However, the performance of ReliefF with Balanced Minority Repeat is very impressive. This approach is able to detect minority classes from the test datasets of 10-fold cross validation over all the models using the pre-selected 1, 5, 10, 20, 50, 80, 100, 150 and 200 most significant genes. For example, the classification model built from 10 most significant genes using Balanced Minority Repeat ReliefF is able to detect 40% of minority classes from the test datasets.

Now, let's take a look on the Accuracy, TNR and BER for the pancreas dataset. We only report the performance of the filtering methods for these three metrics using Naïve Bayes classifier. The tables for these three performance metrics for other two classification methods, Random Forest and IB1, can be found in Appendix A. Next figure (Figure 4.6.5) shows the obtained accuracy rate on pre-selected genes for Naïve Bayes classifier.

Accuracy Naive Bayes Classifier on Pancreas (8.89% + 91.11%) Microarray Expression Data

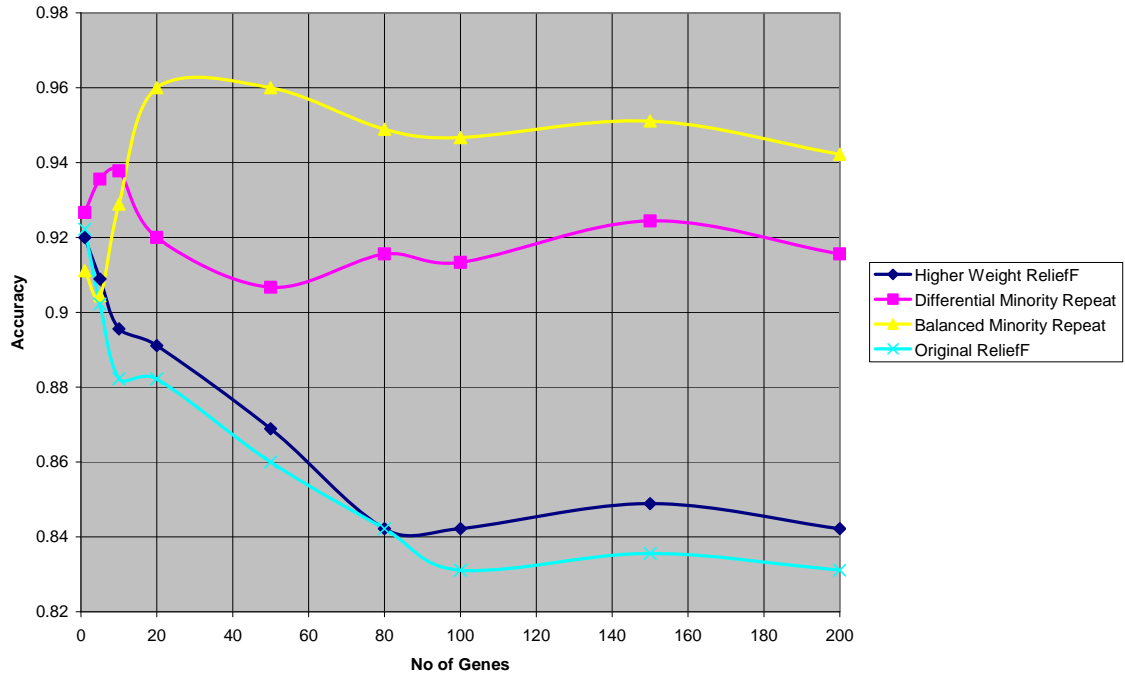


Figure 4.6.5: Accuracy for Naïve Bayes classifier on Pancreas data
 In the following two figures, we plot the TNR and BER for Naïve Bayes classifier on Pancreas dataset.

TNR Naive Bayes Classifier on Pancreas (8.89% + 91.11%) Microarray Expression Data

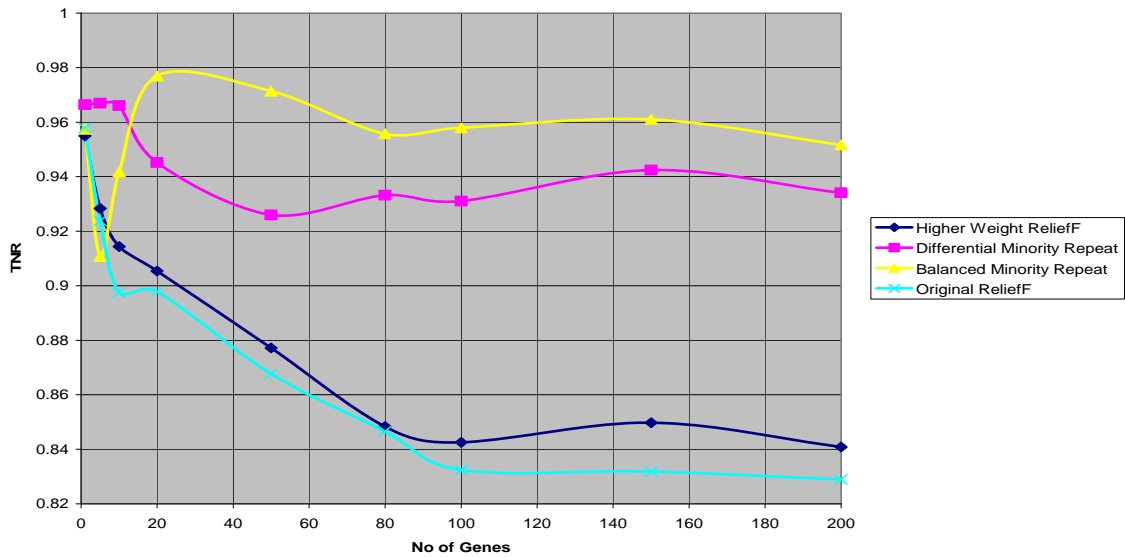


Figure 4.6.6: TNR for Naïve Bayes classifier on Pancreas data

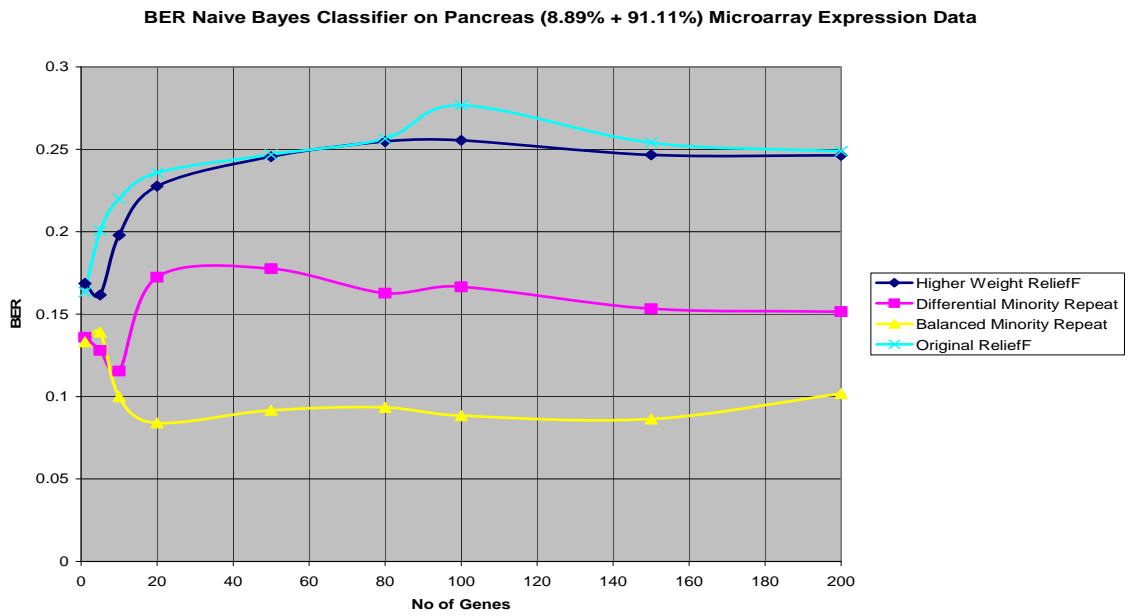


Figure 4.6.7: BER for Naïve Bayes classifier on Pancreas data

4.7 Overall Ranking of Four Filtering Methods

In this section, we summarize the experimental results by ranking all the four filtering methods based on the four performance parameters. Table 4.7.1 illustrates the ranking using TPR in the range of 1 to 4 where higher ranking value means better performance of the classification models. We use Area Under Curve (AUC) to rank the four methods. The actual values of AUC are reported in Appendix B.

Table 4.7.1: Overall Ranking using AUC on TPR performance metric

Dataset Name	Classifier Type	Higher Weight ReliefF	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
Colon	NB	4	3	2	1
	RF	2	1	4	3
	IB1	2	3	4	1
Nevr	NB	4	3	2	1
	RF	3	2	4	1
	IB1	3	2	4	1
Tumor	NB	2	4	3	1
	RF	3	2	4	1
	IB1	2	3	4	1
Lymphoma	NB	2	1	4	3
	RF	3	2	4	1
	IB1	2	3	4	1
Pancreas	NB	3	2	4	3
	RF	2	2	4	3
	IB1	3	2	4	3
Overall Ranking (TPR)		2.66666667	2.33333333	3.66666667	1.66666667

It is clearly evident from above table that ReliefF with Balanced Minority Repeat ranks best (ranking value = 3.67) among the four methods on this very important

performance parameter called TPR. Next comes Higher Weight ReliefF with overall ranking value of 2.67, then ReliefF with Differential Minority Repeat with ranking value of 2.33 and finally original ReliefF with overall ranking of 1.67. In the following three tables, Table 4.7.2, Table 4.7.3 and Table 4.7.4 we report the rankings of these four methods in terms of TNR, Accuracy and BER.

Table 4.7.2: Overall Ranking using AUC on TNR performance metric

Dataset Name	Classifier Type	Higher Weight ReliefF	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
Colon	NB	1	2	4	3
	RF	3	4	1	2
	IB1	1	3	2	4
Nevr	NB	1	2	4	3
	RF	4	3	2	1
	IB1	1	2	3	4
Tumor	NB	3	2	4	1
	RF	2	4	1	3
	IB1	3	4	1	2
Lymphoma	NB	2	4	3	1
	RF	2	3	4	1
	IB1	3	2	1	4
Pancreas	NB	2	3	4	1
	RF	1	4	2	3
	IB1	4	3	1	2
Overall Ranking (TNR)		2.2	3	2.46666667	2.33333333

Although overall ranking using TNR shows that ReliefF with Differential Minority Repeat has better ranking than other three, however, ranking using Accuracy and BER again conforms the superiority of ReliefF with Balanced Minority Repeat to other three filtering methods.

Table 4.7.3: Overall Ranking using AUC on Accuracy performance metric

Dataset Name	Classifier Type	Higher Weight ReliefF	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
Colon	NB	3	2	4	1
	RF	1	4	3	2
	IB1	1	3	4	2
Nevr	NB	2	3	4	1
	RF	3	2	4	1
	IB1	2	3	4	1
Tumor	NB	2	3	4	1
	RF	2	3	4	1
	IB1	2	3	4	1
Lymphoma	NB	2	3	4	1
	RF	3	2	4	1
	IB1	2	4	3	1
Pancreas	NB	2	3	4	1
	RF	1	3	4	2
	IB1	3	2	4	1
Overall Ranking (Accuracy)		2.06666667	2.86666667	3.86666667	1.2

Table 4.7.4: Overall Ranking using AUC on BER performance metric

Dataset Name	Classifier Type	Higher Weight ReliefF	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
Colon	NB	2	1	3	4
	RF	2	4	1	3
	IB1	1	4	2	3
Nevr	NB	2	3	4	1
	RF	3	2	4	1
	IB1	2	3	4	1
Tumor	NB	2	4	3	1
	RF	1	3	4	2
	IB1	2	4	3	1
Lymphoma	NB	2	3	4	1
	RF	3	2	4	1
	IB1	3	4	2	1
Pancreas	NB	2	3	4	1
	RF	1	3	4	2
	IB1	4	3	1	2
Overall Ranking (BER)		2.13333333	3.06666667	3.13333333	1.66666667

Chapter 5

CONCLUSION

In this thesis, we addressed the problem of gene selection from microarray expression data with imbalanced sample distributions. Three gene selection methods, Higher Weight ReliefF, ReliefF with Differential Minority Repeat, and ReliefF with Balanced Minority Repeat are proposed in the thesis. Each method has its own special means of taking the sample distribution into consideration for gene selection. Using four common performance metrics, we concluded that, ReliefF with Balanced Minority Repeat yields the best performance compared to other three methods. For applications which require accurate identifications of the rare events, ReliefF with Balanced Minority Repeat is very effective for building models with compromising the accuracy of the prediction on the rare events or minority class examples. Although we only verified the improved performance of this gene selection technique on microarray expression data, the conclusions drawn in this thesis can be applied to other vast areas of scientific and business applications with imbalanced or biased sample distributions.

Research papers published based on the work reported in this thesis:

- Abu Kamal, Xingquan Zhu, Abhijit Pandya, Sam Hsu, Muhammad Shoaib, The Impact of Gene Selection on Imbalanced Microarray Expression Data, *Proc. Of the 1st International Conference on Bioinformatics and Computational Biology, BICoB*, New Orleans, April, 2009.
- Abu Kamal, Xingquan Zhu, Abhijit Pandya, Sam Hsu, Yong Shi, An Empirical Study of Supervised Learning Methods for Biological Sequence Profiling and Microarray Expression Data Analysis, *Proc. Of the 2008 IEEE International Conference on Information Reuse and Integration (IRI)*, Las Vegas, July, 2008.

Appendix A

Table A.1: TPR for Naïve Bayes on Colon Cancer Data

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.606667	0.606667	0.653333	0.606667
5	0.68	0.66	0.746667	0.68
10	0.745	0.73	0.711667	0.715
20	0.751667	0.761667	0.716667	0.736667
50	0.793333	0.798333	0.751667	0.783333
80	0.798333	0.798333	0.786667	0.798333
100	0.808333	0.803333	0.793333	0.808333
150	0.818333	0.818333	0.8	0.803333
200	0.823333	0.83	0.81	0.818333

Table A.2: TPR for Naïve Bayes on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.595	0.483333	0.58	0.215667
5	0.621667	0.543333	0.67	0.447
10	0.622667	0.624333	0.706667	0.503
20	0.622667	0.603333	0.711	0.557667
50	0.601	0.632667	0.742667	0.584333
80	0.624333	0.644333	0.759333	0.584333
100	0.607667	0.644333	0.746	0.601
150	0.607667	0.644333	0.711	0.626
200	0.607667	0.627667	0.725	0.614333

Table A.3: TPR for Naïve Bayes on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.643333	0.613333	0.384667	0.646667
5	0.66	0.69	0.697333	0.676667
10	0.686667	0.716667	0.729334	0.713333
20	0.708334	0.708334	0.751	0.715
50	0.705	0.698334	0.753334	0.725
80	0.728334	0.698334	0.766667	0.726667
100	0.721667	0.708334	0.736667	0.736667
150	0.716667	0.703334	0.745	0.746667
200	0.716667	0.703334	0.755	0.746667

Table A.4: TPR for Naïve Bayes on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.516667	0.553333	0.473333	0.519167
5	0.713333	0.769167	0.736667	0.690833
10	0.760833	0.814167	0.818333	0.703333
20	0.856667	0.879167	0.853333	0.8
50	0.844167	0.890833	0.873333	0.84
80	0.893333	0.884167	0.893333	0.871667
100	0.906667	0.884167	0.883333	0.86
150	0.886667	0.866667	0.883333	0.896667
200	0.876667	0.856667	0.893333	0.896667

Table A.5: TPR for Naïve Bayes on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.343333	0.293333	0.29	0.343333
5	0.42	0.373333	0.49	0.4
10	0.42	0.393333	0.48	0.44
20	0.43	0.393333	0.47	0.43
50	0.45	0.41	0.5	0.45
80	0.45	0.43	0.513333	0.47
100	0.49	0.43	0.483333	0.47
150	0.49	0.45	0.503333	0.49
200	0.5	0.43	0.503333	0.5

Table A.6: TPR for Random Forest Classifier on Colon Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.64	0.666667	0.703333	0.67
5	0.693333	0.666667	0.776667	0.726667
10	0.736667	0.77	0.816667	0.796667
20	0.766666	0.82	0.783333	0.726667
50	0.796667	0.766667	0.783333	0.743333
80	0.803333	0.76	0.793333	0.82
100	0.833333	0.793333	0.796667	0.833333
150	0.783333	0.83	0.766667	0.776667
200	0.813334	0.78	0.823334	0.776667

Table A.7: TPR for Random Forest Classifier on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.684333	0.662666	0.666	0.616667
5	0.692	0.657	0.746667	0.682667
10	0.628667	0.617	0.724333	0.612
20	0.651333	0.638	0.676	0.633
50	0.688	0.689667	0.691667	0.641333
80	0.600333	0.657	0.727667	0.628667
100	0.677667	0.659333	0.722	0.596333
150	0.648	0.631333	0.650334	0.658
200	0.627	0.623666	0.691	0.614667

Table A.8: TPR for Random Forest Classifier on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.590333	0.578667	0.627	0.568
5	0.662	0.703667	0.765	0.637
10	0.795333	0.722	0.773333	0.755333
20	0.758667	0.762	0.792	0.722
50	0.734333	0.736	0.805333	0.757667
80	0.756	0.733667	0.752	0.757667
100	0.74	0.776	0.792	0.738667
150	0.746	0.732	0.812667	0.693667
200	0.749333	0.709333	0.734333	0.692667

Table A.9: TPR for Random Forest Classifier on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.686	0.702	0.715	0.627333
5	0.864	0.783333	0.84	0.822667
10	0.834	0.819	0.888667	0.811
20	0.818333	0.823667	0.921	0.818
50	0.853667	0.804	0.847333	0.812
80	0.803	0.837	0.865	0.780667
100	0.815333	0.823333	0.863	0.842667
150	0.856333	0.810334	0.874667	0.838667
200	0.848333	0.792	0.828333	0.799

Table A.10: TPR for Random Forest Classifier on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.18	0.17	0.18	0.2
5	0.17	0.16	0.25	0.17
10	0.22	0.21	0.26	0.2
20	0.19	0.21	0.32	0.26
50	0.21	0.21	0.34	0.19
80	0.19	0.2	0.34	0.2
100	0.2	0.15	0.27	0.17
150	0.16	0.21	0.33	0.19
200	0.17	0.16	0.3	0.14

Table A.11: TPR for IB1 Classifier on Colon Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.612667	0.624333	0.624333	0.581
5	0.672	0.647667	0.702667	0.663667
10	0.693333	0.731667	0.724334	0.701
20	0.667667	0.705	0.69	0.678333
50	0.665	0.683333	0.736667	0.641667
80	0.665	0.663333	0.756667	0.631667
100	0.685	0.69	0.756667	0.631667
150	0.708333	0.68	0.763333	0.668333
200	0.708333	0.68	0.72	0.678333

Table A.12: TPR for IB1 Classifier on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.658333	0.646667	0.703333	0.591667
5	0.696667	0.698333	0.723333	0.638333
10	0.718333	0.741667	0.751666	0.651666
20	0.72	0.723333	0.773333	0.653333
50	0.751667	0.72	0.793333	0.646667
80	0.756667	0.74	0.798333	0.688333
100	0.808333	0.776667	0.795	0.673333
150	0.813333	0.785	0.763333	0.683333
200	0.791666	0.793333	0.8	0.633333

Table A.13: TPR for IB1 Classifier on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.556667	0.536667	0.593333	0.503333
5	0.765	0.758333	0.786667	0.745
10	0.775	0.831667	0.798333	0.761667
20	0.82	0.866667	0.858333	0.793333
50	0.846667	0.846667	0.853333	0.808333
80	0.833333	0.856667	0.89	0.823333
100	0.833333	0.883333	0.9	0.833333
150	0.863333	0.861667	0.89	0.831667
200	0.863333	0.873333	0.9	0.821667

Table A.14: TPR for IB1 Classifier on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.614333	0.678334	0.652	0.569333
5	0.684333	0.726	0.728667	0.621
10	0.627	0.791	0.767667	0.572667
20	0.647	0.763667	0.788667	0.571333
50	0.686	0.787	0.808667	0.615333
80	0.724333	0.787667	0.822667	0.636
100	0.714333	0.785	0.802667	0.641
150	0.727667	0.806	0.813333	0.624333
200	0.754333	0.818333	0.82	0.609333

Table A.15: TPR for IB1 Classifier on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.28	0.2	0.25	0.28
5	0.2	0.18	0.39	0.2
10	0.18	0.16	0.39	0.18
20	0.18	0.19	0.37	0.18
50	0.16	0.17	0.36	0.16
80	0.16	0.16	0.4	0.16
100	0.16	0.16	0.4	0.16
150	0.14	0.16	0.44	0.14
200	0.14	0.18	0.44	0.14

Table A.16: TNR for Naïve Bayes on Colon Cancer Data

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.883	0.883	0.864667	0.883
5	0.905667	0.895667	0.891333	0.914
10	0.912333	0.909	0.876333	0.913333
20	0.885	0.898333	0.860333	0.89
50	0.836667	0.848667	0.872333	0.831333
80	0.813	0.830333	0.863	0.810667
100	0.809667	0.794667	0.832	0.804667
150	0.762333	0.769667	0.827	0.795667
200	0.776667	0.765	0.807	0.777333

Table A.17: TNR for Naïve Bayes on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.599667	0.671333	0.656	0.777333
5	0.750667	0.762	0.754333	0.819667
10	0.783667	0.776333	0.775333	0.798333
20	0.797	0.811333	0.802667	0.806333
50	0.794	0.806333	0.837667	0.806667
80	0.799333	0.8	0.834667	0.771667
100	0.790333	0.802333	0.826333	0.783667
150	0.790333	0.812667	0.839	0.775667
200	0.808667	0.814333	0.832	0.776333

Table A.18: TNR for Naïve Bayes on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.899889	0.898444	0.925222	0.926556
5	0.908111	0.933444	0.907778	0.898778
10	0.940778	0.941444	0.924667	0.895143
20	0.960111	0.956778	0.951921	0.935285
50	0.945	0.945	0.948809	0.953445
80	0.949445	0.937444	0.959	0.951667
100	0.953445	0.942778	0.951667	0.948333
150	0.939444	0.939444	0.943666	0.939444
200	0.935444	0.935444	0.959	0.939921

Table A.19: TNR for Naïve Bayes on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.887786	0.884334	0.867111	0.894453
5	0.870795	0.88417	0.880296	0.871508
10	0.891567	0.893615	0.901723	0.860645
20	0.868231	0.915217	0.920914	0.856389
50	0.879826	0.927636	0.91912	0.845
80	0.883596	0.916644	0.905304	0.853993
100	0.879985	0.91839	0.914199	0.845556
150	0.893152	0.924247	0.921787	0.840937
200	0.886207	0.922443	0.925509	0.83623

Table A.20: TNR for Naïve Bayes on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.955	0.966389	0.957341	0.957778
5	0.928333	0.966945	0.910754	0.92381
10	0.914365	0.966111	0.941786	0.897421
20	0.905397	0.945199	0.976945	0.897897
50	0.877182	0.926032	0.971389	0.867738
80	0.848294	0.933254	0.955754	0.846548
100	0.84254	0.931032	0.957897	0.83246
150	0.849762	0.942421	0.961032	0.831825
200	0.840873	0.934087	0.951667	0.828928

Table A.21: TNR for Random Forest Classifier on Colon Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.851476	0.851476	0.85181	0.854476
5	0.917	0.917333	0.892	0.928
10	0.912	0.954	0.889333	0.91
20	0.898667	0.935	0.927667	0.911
50	0.938333	0.920333	0.927333	0.914333
80	0.943333	0.947333	0.938333	0.932333
100	0.921667	0.935667	0.918667	0.919333
150	0.923667	0.935333	0.910333	0.924333
200	0.946	0.940667	0.931667	0.922667

Table A.22: TNR for Random Forest Classifier on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.795667	0.807333	0.823666	0.809
5	0.801667	0.793667	0.813333	0.768333
10	0.792333	0.781667	0.809	0.825667
20	0.830333	0.823	0.820667	0.812667
50	0.811333	0.821333	0.810667	0.799
80	0.835667	0.852667	0.801333	0.811333
100	0.871333	0.839667	0.820333	0.803
150	0.822	0.839	0.851333	0.839667
200	0.842667	0.822667	0.848	0.825

Table A.23: TNR for Random Forest Classifier on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.876926	0.89745	0.908117	0.884426
5	0.96361	0.971095	0.939325	0.939325
10	0.966976	0.97931	0.957429	0.974476
20	0.990143	0.981143	0.969619	0.977491
50	0.966467	0.979952	0.981333	0.973476
80	0.976595	0.98531	0.981333	0.976167
100	0.971833	0.989309	0.991	0.989643
150	0.972476	0.998	0.97253	0.994
200	0.9955	0.976619	0.974333	0.978167

Table A.24: TNR for Random Forest Classifier on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.883277	0.851706	0.873119	0.91088
5	0.956833	0.958166	0.960721	0.947651
10	0.958873	0.96754	0.965516	0.95277
20	0.970754	0.971452	0.957151	0.954492
50	0.966706	0.964786	0.953817	0.970484
80	0.961619	0.956682	0.975397	0.965238
100	0.95169	0.967325	0.979563	0.965682
150	0.962627	0.962984	0.965794	0.954111
200	0.961016	0.961294	0.974151	0.936

Table A.25: TNR for Random Forest Classifier on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.978452	0.978452	0.988333	0.980675
5	0.978532	0.992421	0.983889	0.983254
10	0.987698	0.993056	0.986667	0.990833
20	0.990476	0.991111	0.993333	0.993056
50	0.993333	0.995556	0.993056	0.997778
80	0.991111	0.993333	0.995556	0.995556
100	0.993333	0.993333	0.993056	0.993333
150	0.997778	0.997778	0.993056	0.993333
200	0.993333	0.995556	0.995556	0.993333

Table A.26: TNR for IB1 Classifier on Colon Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.860333	0.877667	0.882	0.884667
5	0.919	0.917667	0.921667	0.917333
10	0.905667	0.900667	0.906333	0.892333
20	0.885667	0.900333	0.891333	0.919667
50	0.904667	0.912	0.882667	0.918667
80	0.898	0.908667	0.892333	0.927
100	0.913667	0.913667	0.921333	0.918667
150	0.900667	0.903	0.910333	0.929333
200	0.908667	0.903	0.902333	0.932

Table A.27: TNR for IB1 Classifier on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.807	0.816	0.760334	0.847333
5	0.822	0.857333	0.822333	0.818333
10	0.813333	0.807667	0.852333	0.805333
20	0.777	0.822	0.822333	0.846333
50	0.769	0.791	0.795667	0.843667
80	0.774	0.792667	0.799333	0.861333
100	0.768667	0.791	0.816667	0.854667
150	0.782667	0.793	0.815333	0.845333
200	0.764334	0.771667	0.809667	0.84

Table A.28: TNR for IB1 Classifier on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.87787	0.874234	0.897377	0.87971
5	0.929078	0.94542	0.923459	0.921783
10	0.951086	0.980134	0.947087	0.93042
20	0.967801	0.963277	0.963324	0.944952
50	0.964333	0.961333	0.960333	0.945991
80	0.972128	0.976492	0.947333	0.973
100	0.981667	0.97803	0.965333	0.981
150	0.964674	0.961674	0.954515	0.975128
200	0.964674	0.96234	0.967515	0.974848

Table A.29: TNR for IB1 Classifier on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.920365	0.924786	0.909635	0.913992
5	0.916833	0.924452	0.930516	0.945833
10	0.950508	0.949802	0.890801	0.951571
20	0.953349	0.954127	0.916198	0.973595
50	0.947705	0.947182	0.935032	0.974619
80	0.965738	0.946825	0.955579	0.961048
100	0.962095	0.943325	0.955659	0.968738
150	0.952016	0.94277	0.946174	0.978421
200	0.955151	0.93777	0.946928	0.973063

Table A.30: TNR for IB1 Classifier on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.986111	0.979087	0.9725	0.988333
5	0.995278	0.997778	0.986111	0.988889
10	0.997778	0.9975	0.978611	0.995278
20	1	0.997778	0.989921	1
50	1	0.997778	0.986389	1
80	0.997778	0.997778	0.971865	0.997778
100	0.997778	0.997778	0.966865	0.997778
150	1	0.997778	0.95131	0.997778
200	1	0.997778	0.935476	1

Table A.31: Accuracy for Naïve Bayes on Colon Cancer Data

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.802381	0.802381	0.809048	0.802381
5	0.841429	0.828095	0.854286	0.837619
10	0.867619	0.858095	0.827619	0.854286
20	0.850952	0.867619	0.823809	0.847619
50	0.843333	0.843809	0.840952	0.837619
80	0.830476	0.840476	0.85381	0.827619
100	0.830476	0.817143	0.835238	0.827619
150	0.803333	0.81	0.835238	0.817143
200	0.807143	0.803333	0.828571	0.813333

Table A.32: Accuracy for Naïve Bayes on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.603333	0.61	0.64	0.593334
5	0.713333	0.703334	0.743333	0.706667
10	0.74	0.733333	0.763333	0.703333
20	0.743333	0.753333	0.783333	0.733333
50	0.753333	0.763333	0.81	0.743333
80	0.766667	0.766667	0.823333	0.72
100	0.753333	0.766667	0.813333	0.726667
150	0.753333	0.776667	0.793333	0.74
200	0.763333	0.77	0.8	0.74

Table A.33: Accuracy for Naïve Bayes on DLBCL Tumor Data

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.81011	0.811648	0.817143	0.824176
5	0.866373	0.903736	0.870549	0.85033
10	0.903297	0.910769	0.903516	0.851868
20	0.936484	0.940879	0.930769	0.904835
50	0.922198	0.929451	0.933846	0.929451
80	0.935165	0.920659	0.943956	0.938242
100	0.940879	0.925055	0.935385	0.92945
150	0.923736	0.920879	0.92967	0.929451
200	0.918022	0.915165	0.943956	0.926594

Table A.34: Accuracy for Naïve Bayes on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.835397	0.817619	0.752222	0.839841
5	0.836825	0.848413	0.849524	0.839682
10	0.860476	0.868571	0.881746	0.842857
20	0.847778	0.883968	0.894921	0.842857
50	0.858254	0.896667	0.900159	0.838413
80	0.868571	0.887778	0.894444	0.847301
100	0.868571	0.89	0.897302	0.842857
150	0.878254	0.895079	0.901587	0.841429
200	0.871587	0.894286	0.906032	0.837778

Table A.35: Accuracy for Naïve Bayes on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.92	0.926667	0.911111	0.922222
5	0.908889	0.935555	0.904445	0.902222
10	0.895556	0.937778	0.928889	0.882222
20	0.891111	0.92	0.96	0.882222
50	0.868889	0.906667	0.96	0.86
80	0.842222	0.915555	0.948889	0.842222
100	0.842222	0.913333	0.946666	0.831111
150	0.848889	0.924444	0.951111	0.835555
200	0.842222	0.915556	0.942222	0.831111

Table A.36: Accuracy for Random Forest Classifier on Colon Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.807143	0.817143	0.821905	0.821905
5	0.850476	0.847143	0.867143	0.880476
10	0.860476	0.900476	0.884286	0.887143
20	0.865238	0.897143	0.893809	0.858095
50	0.89381	0.880476	0.891428	0.871905
80	0.897143	0.890952	0.900952	0.897619
100	0.897619	0.901905	0.894286	0.897143
150	0.881429	0.903333	0.871428	0.888571
200	0.913333	0.890952	0.901905	0.874286

Table A.37: Accuracy for Random Forest Classifier on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.77	0.77	0.776667	0.743333
5	0.78	0.753333	0.813334	0.75
10	0.75	0.74	0.796667	0.753333
20	0.77	0.77	0.793333	0.76
50	0.776667	0.783333	0.783333	0.756667
80	0.77	0.806666	0.786667	0.753333
100	0.816667	0.786667	0.81	0.74
150	0.773333	0.786667	0.793333	0.793333
200	0.776667	0.77	0.803333	0.77

Table A.38: Accuracy for Random Forest Classifier on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.823077	0.841758	0.857582	0.825934
5	0.903516	0.919341	0.916484	0.887692
10	0.939341	0.937802	0.930989	0.938022
20	0.949451	0.943736	0.940879	0.927692
50	0.922418	0.929451	0.958022	0.936703
80	0.935385	0.93956	0.936703	0.934066
100	0.936703	0.950989	0.955385	0.942418
150	0.934066	0.945275	0.950769	0.936483
200	0.952308	0.930989	0.923736	0.928132

Table A.39: Accuracy for Random Forest Classifier on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.821905	0.807143	0.827936	0.824286
5	0.92619	0.904762	0.922381	0.912857
10	0.926825	0.924603	0.932698	0.916667
20	0.929048	0.93127	0.940159	0.919365
50	0.933492	0.918095	0.920952	0.927619
80	0.922381	0.924603	0.945397	0.913492
100	0.914286	0.928413	0.941746	0.929047
150	0.928413	0.919524	0.935079	0.921746
200	0.929048	0.916508	0.933651	0.898095

Table A.40: Accuracy for Random Forest Classifier on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.935556	0.933333	0.942222	0.94
5	0.933333	0.944445	0.946667	0.937778
10	0.948889	0.951111	0.951111	0.948889
20	0.946667	0.951111	0.966667	0.96
50	0.953333	0.955556	0.968889	0.953334
80	0.946667	0.951111	0.971111	0.953334
100	0.951111	0.944444	0.96	0.946667
150	0.948889	0.96	0.968889	0.948889
200	0.948889	0.946667	0.966667	0.942222

Table A.41: Accuracy for IB1 Classifier on Colon Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.801905	0.818571	0.824762	0.808095
5	0.854286	0.840952	0.870476	0.850953
10	0.854286	0.870952	0.860476	0.854286
20	0.851905	0.878095	0.848572	0.874762
50	0.860953	0.871429	0.864762	0.858095
80	0.857619	0.864286	0.870952	0.864286
100	0.871429	0.874762	0.890953	0.860953
150	0.864762	0.864762	0.890952	0.877619
200	0.871429	0.864762	0.871429	0.877619

Table A.42: Accuracy for IB1 Classifier on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.766667	0.766667	0.773333	0.773333
5	0.78	0.8	0.79	0.753333
10	0.78	0.79	0.82	0.76
20	0.76	0.793333	0.813333	0.78
50	0.77	0.776667	0.813333	0.776667
80	0.776667	0.786667	0.82	0.806667
100	0.793333	0.793333	0.83	0.796667
150	0.81	0.806667	0.81	0.79
200	0.786667	0.8	0.816667	0.773333

Table A.43: Accuracy for IB1 Classifier on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.818022	0.806374	0.839121	0.807692
5	0.902637	0.907033	0.905494	0.893846
10	0.915605	0.961319	0.92989	0.904176
20	0.947033	0.952747	0.955604	0.922857
50	0.954286	0.951428	0.96	0.927033
80	0.958461	0.963956	0.957143	0.957143
100	0.965714	0.971209	0.974286	0.965714
150	0.959561	0.956703	0.961319	0.958462
200	0.959561	0.962418	0.972747	0.955605

Table A.44: Accuracy for IB1 Classifier on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.872857	0.883968	0.873016	0.853651
5	0.875079	0.894286	0.899524	0.887619
10	0.899524	0.928254	0.878095	0.880952
20	0.901746	0.929841	0.906826	0.894444
50	0.905397	0.928254	0.921587	0.904762
80	0.926825	0.929683	0.939365	0.897302
100	0.922381	0.92746	0.937143	0.901746
150	0.916508	0.928254	0.931905	0.906984
200	0.924603	0.925238	0.936349	0.900317

Table A.45: Accuracy for IB1 Classifier on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.957778	0.94	0.942222	0.96
5	0.957778	0.957778	0.973333	0.951111
10	0.957778	0.955556	0.966667	0.955556
20	0.96	0.96	0.975556	0.96
50	0.957778	0.957778	0.971111	0.957778
80	0.955556	0.955556	0.962222	0.955556
100	0.955556	0.955556	0.957778	0.955556
150	0.955556	0.955556	0.946667	0.953333
200	0.955556	0.957778	0.931111	0.955556

Table A.46: BER for Naïve Bayes on Colon Cancer Data

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.184191	0.184191	0.175691	0.184191
5	0.1515	0.168833	0.134428	0.139262
10	0.112929	0.128333	0.155595	0.124595
20	0.138762	0.117095	0.164095	0.141262
50	0.153262	0.151595	0.144095	0.141262
80	0.169595	0.155429	0.138262	0.149595
100	0.168262	0.179929	0.160262	0.159262
150	0.182262	0.180595	0.172095	0.177595
200	0.185428	0.187929	0.170095	0.179095

Table A.47: BER for Naïve Bayes on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.368667	0.3845	0.329	0.408833
5	0.2865	0.297	0.2315	0.309167
10	0.261	0.268833	0.231667	0.3135
20	0.2765	0.267	0.230167	0.297833
50	0.2725	0.256	0.205833	0.296167
80	0.2495	0.2435	0.192667	0.308333
100	0.261333	0.244	0.205833	0.2915
150	0.2645	0.2445	0.220834	0.2885
200	0.248	0.251167	0.2025	0.297833

Table A.48: BER for Naïve Bayes on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.196768	0.193336	0.195826	0.161074
5	0.149921	0.104559	0.150488	0.178702
10	0.105364	0.099444	0.102644	0.180794
20	0.063023	0.063654	0.081579	0.111564
50	0.083841	0.077167	0.076745	0.071944
80	0.071064	0.089278	0.066214	0.064142
100	0.064397	0.081778	0.072881	0.074127
150	0.085587	0.092254	0.079548	0.082968
200	0.092587	0.099254	0.066214	0.091397

Table A.49: BER for Naïve Bayes on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.203564	0.218246	0.266238	0.196064
5	0.198203	0.184512	0.198136	0.199111
10	0.172742	0.176401	0.137917	0.194861
20	0.17927	0.154091	0.143937	0.193123
50	0.194361	0.149702	0.128118	0.208504
80	0.177695	0.156599	0.138576	0.190234
100	0.180683	0.149365	0.13819	0.195179
150	0.146702	0.138757	0.130674	0.19
200	0.175175	0.134178	0.131297	0.189833

Table A.50: BER for Naïve Bayes on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.168611	0.135972	0.133234	0.163472
5	0.161667	0.127956	0.13906	0.200595
10	0.197917	0.115417	0.099833	0.22
20	0.227611	0.172417	0.084028	0.235845
50	0.245548	0.177552	0.091667	0.246881
80	0.254742	0.162706	0.093373	0.25648
100	0.255416	0.16654	0.088413	0.27675
150	0.246611	0.153373	0.08629	0.25404
200	0.246337	0.151484	0.10179	0.248837

Table A.51: BER for Random Forest Classifier on Colon Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.188833	0.181167	0.183834	0.176833
5	0.124667	0.134834	0.1335	0.095
10	0.124	0.079333	0.1075	0.093667
20	0.108167	0.093333	0.095	0.1325
50	0.091	0.108	0.096762	0.109929
80	0.091167	0.0865	0.079595	0.087167
100	0.085667	0.081333	0.099095	0.085333
150	0.101167	0.085167	0.117167	0.106762
200	0.075333	0.094333	0.084262	0.107

Table A.52: BER for Random Forest Classifier on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.230333	0.225167	0.219167	0.261167
5	0.219167	0.246167	0.193667	0.251333
10	0.2615	0.266	0.2225	0.246333
20	0.219167	0.234833	0.2215	0.2465
50	0.228833	0.207833	0.224167	0.253833
80	0.230333	0.198833	0.223	0.244667
100	0.175333	0.199	0.179	0.267
150	0.226833	0.213167	0.217667	0.201333
200	0.235167	0.234	0.208667	0.235667

Table A.53: BER for Random Forest Classifier on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.213759	0.164168	0.14875	0.198751
5	0.109306	0.096211	0.092223	0.113096
10	0.062091	0.063576	0.087988	0.060964
20	0.042036	0.049778	0.067488	0.066219
50	0.075163	0.068484	0.038345	0.061504
80	0.060238	0.053103	0.049655	0.054798
100	0.052064	0.042326	0.028559	0.045083
150	0.066242	0.034171	0.05025	0.044469
200	0.032873	0.070194	0.063885	0.060107

Table A.54: BER for Random Forest Classifier on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.194735	0.227671	0.195791	0.184028
5	0.067806	0.083917	0.085123	0.083024
10	0.068044	0.071115	0.067794	0.089988
20	0.06131	0.059822	0.063476	0.085325
50	0.0675	0.067833	0.084873	0.068937
80	0.078996	0.074731	0.052397	0.081885
100	0.101111	0.072739	0.053865	0.073941
150	0.072123	0.077862	0.066095	0.08937
200	0.071905	0.077587	0.057516	0.118218

Table A.55: BER for Random Forest Classifier on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.091706	0.102679	0.06875	0.080595
5	0.099663	0.050317	0.079306	0.089484
10	0.056052	0.046528	0.068334	0.061806
20	0.057817	0.060417	0.043889	0.042222
50	0.050417	0.040695	0.037778	0.032639
80	0.062917	0.051667	0.032778	0.041944
100	0.051667	0.055139	0.042361	0.05375
150	0.035	0.029306	0.042778	0.052639
200	0.052917	0.045	0.035	0.056111

Table A.56: BER for IB1 Classifier on Colon Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.188833	0.176333	0.199	0.185333
5	0.153667	0.151667	0.142167	0.1375
10	0.144333	0.145167	0.1445	0.167167
20	0.141167	0.110333	0.1485	0.115666
50	0.131667	0.117333	0.126333	0.139333
80	0.143167	0.128167	0.140333	0.131834
100	0.121833	0.119167	0.114667	0.136833
150	0.130333	0.130833	0.114167	0.107167
200	0.124833	0.130833	0.1285	0.104

Table A.57: BER for IB1 Classifier on Central Nervous System Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.218333	0.2225	0.228167	0.231667
5	0.222167	0.197167	0.208166	0.232833
10	0.213667	0.200333	0.176333	0.2425
20	0.229833	0.199833	0.188333	0.2035
50	0.229	0.214833	0.178167	0.219667
80	0.213333	0.2105	0.179167	0.178667
100	0.2065	0.206167	0.17	0.197167
150	0.180833	0.190167	0.193167	0.206167
200	0.206	0.197833	0.176833	0.232667

Table A.58: BER for IB1 Classifier on DLBCL Tumor Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.203623	0.204219	0.171076	0.209814
5	0.125119	0.107928	0.122595	0.129052
10	0.096262	0.041857	0.082357	0.109786
20	0.063524	0.068833	0.059024	0.098928
50	0.048595	0.051095	0.050691	0.085429
80	0.051691	0.045024	0.048095	0.046357
100	0.035024	0.031595	0.028095	0.038024
150	0.054691	0.054357	0.045595	0.056357
200	0.054691	0.061762	0.034762	0.051357

Table A.59: BER for IB1 Classifier on Lymphoma Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.163334	0.174193	0.164132	0.181555
5	0.166368	0.158872	0.120254	0.124301
10	0.113773	0.083659	0.147437	0.14215
20	0.110205	0.091948	0.123556	0.087031
50	0.098468	0.077572	0.106429	0.084806
80	0.073697	0.084798	0.077941	0.126194
100	0.071657	0.097492	0.077873	0.102861
150	0.094296	0.09631	0.086897	0.082461
200	0.089713	0.099595	0.086996	0.09375

Table A.60: BER for IB1 Classifier on Pancreas Dataset

	Higher Weight	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
1	0.059861	0.080417	0.070972	0.049861
5	0.034167	0.030278	0.051945	0.059167
10	0.030278	0.026389	0.063929	0.040417
20	0.020278	0.029305	0.031567	0.020278
50	0.021389	0.030417	0.042778	0.021389
80	0.031389	0.031389	0.060556	0.031389
100	0.031389	0.031389	0.056389	0.031389
150	0.0225	0.031389	0.075	0.0325
200	0.0225	0.030278	0.075833	0.0225

Appendix B

Table B.1: Area Under Curve (AUC) on TPR performance metric

Dataset Name	Classifier Type	Higher Weight ReliefF	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
Colon	NB	6.12	6.09	6.038	6.037
	RF	6.14	6.13	6.28	6.15
	IB1	5.42	5.45	5.8	5.25
Nevr	NB	4.91	4.89	5.7	4.32
	RF	5.24	5.19	5.62	5.07
	IB1	5.99	5.91	6.15	5.25
Tumor	NB	6.56	6.69	6.62	6.37
	RF	5.86	5.81	6.17	5.7
	IB1	6.45	6.61	6.72	6.26
Lymphoma	NB	5.61	5.58	5.75	5.74
	RF	6.61	6.45	6.87	6.44
	IB1	5.49	6.19	6.27	4.87
Pancreas	NB	3.57	3.24	3.84	3.57
	RF	1.52	1.52	2.35	1.55
	IB1	1.39	1.37	3.1	1.39

Table B.2: Area Under Curve (AUC) on TNR performance metric

Dataset Name	Classifier Type	Higher Weight ReliefF	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
Colon	NB	6.75	6.77	6.86	6.79
	RF	7.35	7.44	7.3	7.33
	IB1	7.21	7.25	7.22	7.33
Nevr	NB	6.21	6.31	6.41	6.34
	RF	6.58	6.57	6.56	6.48
	IB1	6.29	6.45	6.51	6.72
Tumor	NB	7.514	7.513	7.53	7.46
	RF	7.74	7.82	7.73	7.76
	IB1	7.65	7.68	7.59	7.6
Lymphoma	NB	7.05	7.28	7.26	6.84
	RF	7.65	7.66	7.68	7.63
	IB1	7.59	7.54	7.46	7.7
Pancreas	NB	7.06	7.56	7.63	6.99
	RF	7.92	7.94	7.931	7.934
	IB1	7.98	7.973	7.78	7.972

Table B.3: Area Under Curve (AUC) on Accuracy performance metric

Dataset Name	Classifier Type	Higher Weight ReliefF	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
Colon	NB	6.67	6.66	6.69	6.65
	RF	7.01	7.08	7.07	7.03
	IB1	6.85	6.91	6.95	6.88
Nevr	NB	5.91	5.95	6.25	5.74
	RF	6.21	6.2	6.37	6.06
	IB1	6.25	6.33	6.49	6.24
Tumor	NB	7.29	7.31	7.33	7.21
	RF	7.41	7.45	7.48	7.38
	IB1	7.49	7.54	7.55	7.41
Lymphoma	NB	6.87	7.03	7.05	6.73
	RF	7.36	7.31	7.42	7.3
	IB1	7.25	7.37	7.32	7.15
Pancreas	NB	6.98	7.37	7.53	6.91
	RF	7.57	7.6	7.69	7.59
	IB1	7.66	7.65	7.69	7.64

Table B.4: Area Under Curve (AUC) on BER performance metric

Dataset Name	Classifier Type	Higher Weight ReliefF	Differential Minority Repeat	Balanced Minority Repeat	Original ReliefF
Colon	NB	1.26	1.27	1.24	1.21
	RF	0.857	0.81	0.86	0.852
	IB1	1.12	1.06	1.09	1.08
Nevr	NB	2.18	2.14	1.78	2.46
	RF	1.793	1.795	1.695	1.959
	IB1	1.707	1.63	1.5	1.712
Tumor	NB	0.77	0.75	0.76	0.89
	RF	0.59	0.524	0.521	0.576
	IB1	0.6	0.53	0.54	0.69
Lymphoma	NB	1.44	1.29	1.21	1.56
	RF	0.65	0.66	0.6	0.72
	IB1	0.85	0.83	0.87	0.89
Pancreas	NB	1.8	1.22	0.8	1.9
	RF	0.49	0.41	0.4	0.44
	IB1	0.23	0.266	0.46	0.272

References

1. Robnik-Šikonja, M. and Kononenko, I. 2003. Theoretical and Empirical Analysis of ReliefF and RReliefF. *Mach. Learn.* 53, 1-2 (Oct. 2003), 23-69.
2. Hunter, L. 1993. Molecular biology for computer scientists. In *Artificial intelligence and Molecular Biology*, L. Hunter, Ed. American Association for Artificial Intelligence, Menlo Park, CA, 1-46.
3. Wikipedia, the free Encyclopedia. (<http://en.wikipedia.org/wiki/Gene>).
4. Rodnina M.V., Beringer M., Wintermeyer W. (2007). "How ribosomes make peptide bonds". *Trends Biochem. Sci.* 32 (1): 20–6. doi:10.1016/j.tibs.2006.11.007. PMID 17157507.
5. Deonier, Richard C., Tavaré Simon, Waterman, Michael S. (2007). *Computational Genome Analysis, An Introduction*, Chapter 11 (Measuring Expression of Genome Information). ISBN: 978-0-387-98785-9.
6. Cancerquest, May 01, 2003. The Genes of Cancer. (<http://www.cancerquest.org/index.cfm?page=261>).
7. Cancer Cell, July 22, 2003, University of Illinois at Chicago. (<http://www.hopkinsbreastcenter.org/artemis/200308/feature6.html>)
8. C. D. Logsdon, D. M. Simeone, C. Binkley, T. Arumugam, J-K. Greenon, T. J. Giordano, D. E. Misek, and S. Hanash. Molecular profiling of pancreatic adenocarcinoma and chronic pancreatitis identifies multiple genes differentially regulated in pancreatic cancer. *Cancer Research*, 63:2649--2657, 2003.
9. Langley, P. & Sage, S. (1994). Oblivious decision trees and abstract cases. In *Working Notes of the AAI-94 Workshop on Case-Based Reasoning* (pp. 113-117). Seattle, WA: AAI Press.
10. Langley, P. and Sage, S. 1997. Scaling to domains with irrelevant features. In *Computational Learning theory and Natural Learning Systems: Volume 4: Making Learning Systems Practical*, R. Greiner, T. Petsche, and S. J. Hanson, Eds. MIT Press, Cambridge, MA, 51-63.

11. Aha, D. W., Kibler, D., and Albert, M. K. 1991. Instance-Based Learning Algorithms. *Mach. Learn.* 6, 1 (Jan. 1991), 37-66. DOI=<http://dx.doi.org/10.1023/A:1022689900470>.
12. Baglioni, M., Furletti, B., and Turini, F. 2005. DrC4.5: Improving C4.5 by means of prior knowledge. In *Proceedings of the 2005 ACM Symposium on Applied Computing* (Santa Fe, New Mexico, March 13 - 17, 2005). L. M. Liebrock, Ed. SAC '05. ACM, New York, NY, 474-481. DOI=<http://doi.acm.org/10.1145/1066677.1066787>.
13. Kohavi R. and John G. 1996. Wrappers for feature subset selection. *Artificial Intelligence, special issue on relevance*, 97(1-2):273-324.
14. Almuallim, H. and Dietterich, T. G. 1994. Learning Boolean concepts in the presence of many irrelevant features. *Artif. Intell.* 69, 1-2 (Sep. 1994), 279-305. DOI= [http://dx.doi.org/10.1016/0004-3702\(94\)90084-1](http://dx.doi.org/10.1016/0004-3702(94)90084-1).
15. Fountain, T., Almuallim, H., and Dietterich, T. G. 1991 *Learning with Many Irrelevant Features*. Technical Report. UMI Order Number: 91-30-04., Oregon State University.
16. Caruana, R. & Freitag, D. (1994). Greedy attribute selection. In *Proceedings of the 1994 International Conference on Machine Learning*, pp. 28-36. Morgan Kaufmann, CA.
17. Liu, H. and Setiono, R.. A probabilistic approach to feature selection: a filter solution. In *International Conference on Machine Learning (ICML-96)*, pp. 319-327.
18. Modrzejewski, M. 1993. Feature Selection Using Rough Sets Theory. In *Proceedings of the European Conference on Machine Learning* (April 05 - 07, 1993). P. Brazdil, Ed. Lecture Notes In Computer Science, vol. 667. Springer-Verlag, London, 213-226.
19. Pawlak Z., 1991. *Rough Sets, Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht 1991.
20. Liu H. and Sentiono R. 1995. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence*, (November 5-8, 1995), pp. 388-391.
21. Cardie C. (1995). Using decision trees to improve cased-based learning. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*. AAAI Press.

22. Singh, M., Provan, G.M., 1996. Efficient learning of selective Bayesian network classifiers. In *International Conference of Machine Learning*, pp. 453-461.
23. Koller, D. and Sahami M. (1996). Towards optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 284-292). Morgan Kaufmann.
24. Kohavi, R. and John, G. H. 1997. Wrappers for feature subset selection. *Artif. Intell.* 97, 1-2 (Dec. 1997), 273-324. DOI= [http://dx.doi.org/10.1016/S0004-3702\(97\)00043-X](http://dx.doi.org/10.1016/S0004-3702(97)00043-X).
25. Langley, P., & Sage, S. (1994). Induction of Selective Bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence* (pp. 399-406). Seattle, WA: Morgan Kaufmann.
26. Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. 1992 *Numerical Recipes in C (2nd Ed.): the Art of Scientific Computing*. Cambridge University Press.
27. I. Kononenko. On biases in estimating multi-valued attributes, in: *Proc. of the 14th International Joint Conference on Artificial Intelligence*, 1995, pp. 1034-1040.
28. Quinlan, J. R. 1986. Induction of Decision Trees. *Mach. Learn.* 1, 1 (Mar. 1986), 81-106. DOI= <http://dx.doi.org/10.1023/A:1022643204877>.
29. Quinlan, J. R. 1993 *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc.
30. K. Cherkauer, J. Shavlik, Growing simpler decision trees to facilitate knowledge discovery. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 315-318.
31. Langley, P. & Sage, S. (1994). Oblivious decision trees and abstract cases. In *Working Notes of the AAAI-94 Workshop on Case-Based Reasoning* (pp. 113-117). Seattle, WA: AAAI Press.
32. Aha, D. W. & Bankert, R. L. (1994). Feature Selection for Case-Based Classification of Cloud Types: An Empirical Comparison. In *Proceedings of The 1994 AAAI Workshop on Case-Based Reasoning*, pp. 106-112. Seattle, WA: AAAI Press.
33. Domingos, P. 1997. Context-sensitive feature selection for lazy learners. In *Lazy Learning*, D. W. Aha, Ed. Kluwer Academic Publishers, Norwell, MA, 227-253.

34. Dietterich, T. G. (1997). *Machine learning research: Four current directions*. AI Magazine 18(4), 97--136.
35. Kononenko, I., Šimec, E., and Robnik-Šikonja, M. 1997. Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF. *Applied Intelligence* 7, 1 (Jan. 1997), 39-55. DOI= <http://dx.doi.org/10.1023/A:1008280620621>.
36. U. Pompe and I. Kononenko, "Linear space induction in first order logic with RELIEFF," in *Mathematical and Statistical Methods in Artificial Intelligence*, edited by G. Della Riccia, R. Kruse, and R. Viertl, CISM Lecture Notes, Springer-Verlag, 1995.
37. Robnik-Sikonja, M. and Kononenko, I. 1997. An adaptation of Relief for attribute estimation in regression. In *Proceedings of the Fourteenth international Conference on Machine Learning* (July 08 - 12, 1997). D. H. Fisher, Ed. Morgan Kaufmann Publishers, San Francisco, CA, 296-304.
38. Kira, K. and Rendell, L. (1992), "The feature selection problem: Traditional methods and a new algorithm", in: *Proceedings of AAAI-92*, AAAI Press, 129-134.
39. Kononenko, I. 1994. Estimating attributes: analysis and extensions of RELIEF. In *Proceedings of the European Conference on Machine Learning on Machine Learning* (Catania, Italy). F. Bergadano and L. De Raedt, Eds. Springer-Verlag New York, Secaucus, NJ, 171-182.
40. Chen, X. and Wasikowski, M. 2008. FAST: a roc-based feature selection metric for small samples and imbalanced data classification problems. In *Proceeding of the 14th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining* (Las Vegas, Nevada, USA, August 24 - 27, 2008). KDD '08. ACM, New York, NY, 124-132. DOI= <http://doi.acm.org/10.1145/1401890.1401910>.
41. Guo-Zheng Li, Hao-Hua Meng, Jun Ni, 2008. Embedded Gene Selection for Imbalanced Microarray Data Analysis. In *International Multi-Symposiums in Computer and Computational Sciences* (October 18, 2008). IMSCCS'08, Shanghai Jiaotong University, Medical School, Shanghai, China.
42. Liu, X., Wu, J., and Zhou, Z. 2006. Exploratory Under-Sampling for Class-Imbalance Learning. In *Proceedings of the Sixth international Conference on Data Mining* (December 18 - 22, 2006). ICDM. IEEE Computer Society, Washington, DC, 965-969. DOI= <http://dx.doi.org/10.1109/ICDM.2006.68>.
43. He, Y., Tang, Y., Zhang, Y., and Sunderraman, R. 2006. Fuzzy-Granular Gene Selection from Microarray Expression Data. In *Proceedings of the Sixth IEEE*

international Conference on Data Mining - Workshops (December 18 - 22, 2006).
ICDMW. IEEE Computer Society, Washington, DC, 153-157. DOI=
<http://dx.doi.org/10.1109/ICDMW.2006.84>.