Summer 8-2008

Knowledge-Based Analysis of Genomic Expression Data by Using Different Machine Learning Algorithms for the Purpose of Diagnostic, Prognostic or Therapeutic Application

Venkata Jagan Mohan Thodima
University of Southern Mississippi

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

pg. IV.

The University of Southern Mississippi

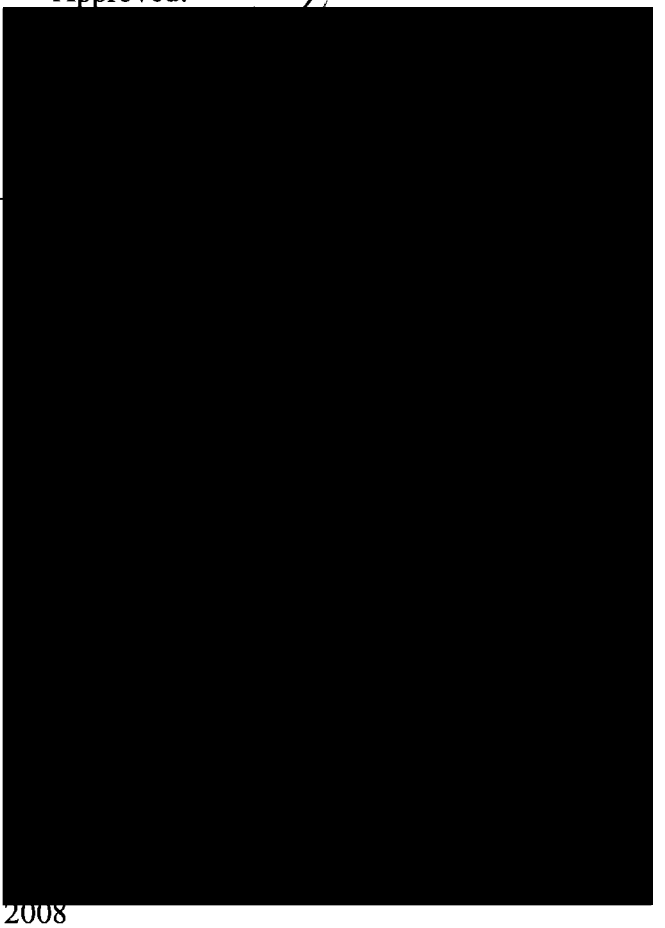
KNOWLEDGE-BASED ANALYSIS OF GENOMIC EXPRESSION DATA BY USING DIFFERENT MACHINE LEARNING ALGORITHMS FOR THE PURPOSE OF DIAGNOSTIC, PROGNOSTIC OR THERAPEUTIC APPLICATION

by

Venkata Jagan Mohan Thodima

A Dissertation
Submitted to the Graduate Studies Office
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Approved:



August 2008

The University of Southern Mississippi

KNOWLEDGE-BASED ANALYSIS OF GENOMIC EXPRESSION DATA BY USING DIFFERENT MACHINE LEARNING ALGORITHMS FOR THE PURPOSE OF DIAGNOSTIC, PROGNOSTIC OR THERAPEUTIC APPLICATION

by

Venkata Jagan Mohan Thodima

A Dissertation
Submitted to the Graduate Studies Office
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

August 2008

ABSTRACT

KNOWLEDGE-BASED ANALYSIS OF GENOMIC EXPRESSION DATA BY USING DIFFERENT MACHINE LEARNING ALGORITHMS FOR THE PURPOSE OF DIAGNOSTIC, PROGNOSTIC OR THERAPEUTIC APPLICATION

by Venkata J. Thodima

August 2008

With more and more biological information generated, the most pressing task of bioinformatics has become to analyze and interpret various types of data, including nucleotide and amino acid sequences, protein structures, gene expression profiling and so on. In this dissertation, we apply the data mining techniques of feature generation, feature selection, and feature integration with learning algorithms to tackle the problems of disease phenotype classification, clinical outcome and patient survival prediction from gene expression profiles.

We analyzed the effect of batch noise in microarray data on the performance of classification. Batchmatch, a batch adjusting algorithm based on double scaling method is advantageous over Combat, another batch correcting algorithm based on the empirical bayes frame work. In order to identify genes associated with disease phenotype classification or patient survival prediction from gene expression data, we compared and analyzed the performance of five feature selection algorithms. Our observations from these studies indicated that Gainratio algorithm performs better and more consistently over the other algorithms studied.

When it comes to performance metric to choose the best classifiers, MCC gives unbiased performance results over accuracy in some endpoints, where class imbalance is more. In the aspect of classification algorithms, no single algorithm is absolutely superior to all others, though SVM achieved fairly good results in most endpoints. Naive bayes algorithm also performed well in some endpoints. Overall, from the total 65 models we reported (5 top models for 13 end points) SVM and SMO (a variant of SVM) dominate mostly, also the linear kernel performed well over RBF in our binary classifications.

DEDICATED TO

MY PARENTS AND WIFE

ACKNOWLEDGEMENTS

First and foremost I would like to acknowledge my supervisor, Dr. Youping Deng, for his encouragement, patience and prompt help. Dr. Deng always provides me complete freedom to explore and work on the research topics that I have interests in. Although it was difficult for me to make quick progress at the beginning, I must appreciate the wisdom of his supervision when I started to think for myself and become relatively independent.

I would like to also thank the members of my dissertation committee, Dr. Mohamed Elasri, Dr. Joe Zhang, Dr. Mac Alford and Dr. Jonathan Sun for their advice and support throughout the duration of this project.

I would like to thank the participating members and working groups of Microarray Quality Control (MAQC-II) consortium for their discussions and advice during this collaborative project. I would especially like to thank Dr. Leming Shi, principal investigator in NCTR, FDA and also the coordinator of this MAQC project, for giving me the opportunity to participate in this project. I would also thank to the Mississippi Functional Genomics Network (MFGN) for their support during this project.

I would like to thank to my colleagues Arun Rawat and Satish Pasula for their efforts in reading the draft copy and suggesting clarifications, and my other lab members, Dr. Mehdi Pirooznia, Tanwir Habib, Dr. Puneet Bandi, Kuan Yang and Ying Li for their help during this project.

I could not have finished my dissertation work without the strong support from my family. Special thanks must go to my parents and brother; together with them is my

wife, Harini, who is always there to provide me her support through both the highs and lows of my time.

TABLE OF CONTENTS

ABSTRACT....ii

DEDICATION....v

ACKNOWLEDGEMENTS....vi

LIST OF ILLUSTRATIONS....x

LIST OF TABLES....xii

CHAPTER

I. INTRODUCTION AND BACKGROUND....1

Motivation....1

Microarray Quality Control (MAQC) Consortium....4

Overview of the Project
MAQC-I Findings
Objectives of MAQC-II
Design of MAQC-II

Supervised Learning and Classification Algorithms....12

Gene Expression Data Representation for Classification
Results Evaluation and Error Estimation
Error Estimation Methods
Classification Algorithms

Feature Selection Algorithm....27

Categorization of Feature Selection Algorithms
Feature Selection Algorithms

Chapter Summary....34

II. MATERIALS AND METHODS....36

Methods....36

Identification of Outlier Samples
Data Normalization

Quality Control Check
Checking the Batch Effect
Dimensionality Reduction / Feature Selection
Classification
Error Estimation

Materials....49

Datasets

Hamner Lung Tumor Dataset
Iconix Liver Cancer Dataset
NIEHS Liver Cancer Dataset
MDACC Breast Cancer Dataset
Multiple Myeloma Dataset
Neuroblastoma Dataset

Chapter Summary....54

III. RESULTS AND DISCUSSION....55

Results....55

Outlier Identification
Preprocessing and Normalization
Batch Effect and Correction
Dimensionality Reduction
Feature Selection / Evaluation
Classification / Error Estimation

Discussion....105

Future Work....107

APPENDICES

A: Summary of MAQC-II Datasets....108
B: Summary of outlier-voting results on the Hamner lung tumor data set...111
C: Summary of outlier-voting results on the MDACC-BR cancer dataset..112
D: USM Data Analysis Plan (DAP)....113
E: Summary of the candidate models for 13 endpoints and the gene lists used for each model....117
F: MAQC - II Participating group list....130

REFERENCES....131

LIST OF ILLUSTRATIONS

Figure

1. Schematic representations of the two major types of applications of 7

2. Confusion matrix for two-class classification problem.....15

3. A sample ROC curve. The dotted line on the 45 degree diagonal is.....18

4. A Graphical depiction of 10-fold cross validation.....20

5. Graphical representation of Support Vector Machines concept.....22

6. Graphical depiction of two feature selection (Filter and Wrapper).....29

7. A small schematic depiction of dimensionality reduction of gene expression.....46

8. The experimental design of the Hamner lung tumor dataset.....49

9. The box plot distribution of RMA normalized 70 array samples of Hamner.....57

10. The box plot distribution of RMA normalised values for 178 array samples.....61

11. Summarized view of the array outlier voting from different analysis groups.....64

12. Correlation heat map of the Hamner lung tumor dataset with 70 arrays.....66

13. Correlation heat map of the Hamner lung tumor dataset with 70 arrays after.....67

14. Principal component analysis (PCA) of the Hamner lung tumor dataset.....68

15. Q-Q plots for the Hamner dataset before correction of batch (top) and after.....69

16. The above correlation heat maps for Hamner lung tumor dataset shows.....70

17. The two-way ANOVA (LT_NLT and Batch label *i.e.* Year) results for Hamner.71

18. The correlation heat maps for Iconix liver cancer dataset.....72

19. Principal component analysis (PCA) of Iconix liver cancer dataset.....72

20. The two-way ANOVA (Class and Batch *i.e.* Year) results for Iconix.....73

21. 512 differentially expressed genes (FC > 2 and P-value < 0.05) in NIEHS 76

22. 106 differentially expressed genes (FC>2 and P-value < 0.05) in MDACC.....77

23. 197 differentially expressed genes (FC > 2 and P-value < 0.05) in MDACC.....78

24. The schematic depiction of nested crossvalidation.....85

25. The schematic diagram of the data analysis plan we studied.....89

LIST OF TABLES

Table

1. This table shows an example of gene expression data.....14

2. dChip analysis results of the Hamner dataset which.....57

3. dChip analysis results of the MDACC breast cancer.....60

4. Consensus array outliers which are excluded from the further analysis.....63

5. SVM and Naïve Bayes (NB) classification performance.....74

6. The differentially expressed genes passed the fold change.....75

7. The five feature selection algorithms classification performance with six.....81

8. Classification performance of three class (LT, NLT and Ctr) prediction.....83

9. Classification performance of only two class (LT and NLT+Ctr) prediction.....83

10. Classification performance of chemical compound based (multi class) 84

11. The classification performance and the best classifiers using nested cross.....86

12. The classification performance and the best classifiers using nested.....87

13. The table shows the top five models for the NT_NLT class (A).....90

14. The table shows the top five models for the Class (B) of the Iconix.....91

15. The table shows the top five models for the Class (C) of the NIEHS.....92

16. The table shows the top five models for the class pCR (D) of the MDACC.....93

17. The table shows the top five models for the class erpos (E) of the MDACC.....94

18. The table shows the top five models for the class OS_MO (F) of the MM.....95

19. The table shows the top five models for the class EFS_MO (G) of the MM.....96

20. The table shows the top five models for the class CPS1 (H) of the MM 97

21. The table shows the top five models for the class CPR1 (I) of the MM.....98

22. The table shows the top five models for the class OS_MO (J) of the NB.....99

23. The table shows the top five models for the class EFS_MO (K) of the NB.....100

24. The table shows the top five models for the class NEP_S (L) of the NB.....101

25. The table shows the top five models for the class NEP_R (M) of the NB.....102

CHAPTER I

INTRODUCTION AND BACKGROUND

The past two decades witnessed an explosive growth in biological information generated by the scientific community. This was caused by major breakthrough advances in the field of molecular biology, coupled with advances in genomic technologies. In turn, the huge amount of genomic data not only leads to a demand on the computer science community to help store, organize and index the data, but also leads to a demand for specialized computational tools to view and analyze the data.[1]

"Biological science in the 21st century is being transformed from a purely lab-based science to an information science as well".[1]

As a result of this transformation, a new field of science was born, in which biology, computer science, and information technology merge to form a single discipline called *bioinformatics*.[1]

Motivation

Two decades ago, the main role of bioinformatics was to create and maintain databases to store biological information, such as nucleotide and amino acid sequences. With more and more data generated, nowadays, the most pressing task of bioinformatics

---

[1] http://www.ncbi.nlm.nih.gov/About/primer/bioinformatics.html

has moved to analysis and interpretation of various types of data, including nucleotide and amino acid sequences, protein domains, protein structures and so on. To meet the new requirements arising from the new tasks, researchers in the field of bioinformatics are working on the development of new algorithms (mathematical formulas, statistical methods, etc.) and software tools which are designed for assessing relationships among large data sets, such as methods to locate a gene within a sequence, predict protein structure and/or function and understand diseases at gene expression level.

In recent years, the rapid development of DNA microarray technology has made it possible for scientists to measure the expression levels of thousands of genes in a single experiment (Schena *et al.* 1995, Lockhart *et al.* 1996). Thus, DNA microarray technology has found many applications in biomedical research. There are many active research applications of this technology in clinical cancer research; it is being used to better understand the biological mechanisms underlying oncogenesis (Butte 2002), in cancer classification (predictors of good outcome versus poor outcome) (Golub *et al.* 1999; Petricoin *et al.* 2002; van 't Veer *et al.* 2002), clinical diagnosis (Yeang *et al.* 2001) and in drug discovery studies. One of the main challenging tasks in this clinical cancer research is the prediction of outcome, i.e., the potentiality of cancer regression and for severe status (metastasis). The need for sensitive and reliable predictors of clinical outcomes is crucial for early discovery of cancer patients. Identification of these clinical outcomes has direct effect on the choice of optimal therapy for each individual (Perez *et al.* 2004; Pusztai *et al.* 2005; Simon 2005).

Currently, there are two approaches to the computational analysis of gene expression data for clinical classification purpose. The two approaches are discrimination (supervised learning) and clustering (unsupervised learning). In unsupervised learning, the classes are unknown and need to be discovered from the data (Brown *et al.* 2000). This involves estimating the number of classes or clusters by using a clustering algorithm such as hierarchical clustering (Eisen *et al.* 1998; Spellman *et al.* 1998) or self-organizing maps (Tamayo *et al.* 1999) and assigning objects to these classes. In supervised learning (also known as classification, supervised pattern recognition and class prediction), the classes are predefined and the goal is to understand the basis for the classification from a set of labeled data, also known as the learning set. This learned information is then used to build a classifier or model, which will be used to predict the class or label of the future unlabeled (blind) data, also known as external validation dataset (Dudoit *et al.* 2002).

Recently, significant research effort has been directed to the prediction of clinical outcomes for several kinds of cancer on the basis of microarray data, which reported a considerable success in this class prediction results (Bair *et al.* 2004; Beer *et al.* 2002; Bhattacharjee *et al.* 2001; Khan *et al.* 2001; Ramaswamy *et al.* 2003; Rosenwald *et al.* 2002; Yeoh *et al.* 2002). But still there are two problems in this approach, the first is when one analysis group's class model or predictor was tested on another group's same type of cancer data, the success rate decreased significantly, and the second is comparison of the marker gene lists used to predict a model by different groups revealed very small overlap (Ein-Dor *et al.* 2006).

The probable explanation for these problems may be due to several variables like patient's age, race, sex, etc. and in the case of toxicological data like the amount of dose, time, etc. Also, the platform of microarray technology used and the different methods of data analysis play a significant role in these discrepancies (Ein-Dor *et al.* 2006; Michiels *et al.* 2005), which we are studying extensively as one analysis group through participating in the Microarray Quality Control Phase II (MAQC-II) project initiated by the Federal Drug Administration (FDA).

Microarray Quality Control (MAQC) Project

Overview of the Project

On March 16, 2004, the US Food and Drug Administration (FDA) released a report on "*Innovation/Stagnation: Challenge and Opportunity on the Critical Path to New Medical Products*", addressing the recent slowdown in innovative medical products submitted to the FDA for approval. The report described the urgent need to modernize the medical product development process – the Critical Path from bench to bed side, and they released the Critical Path Opportunities list that provided a concrete focus for public and private efforts in new research development and tools. Among the 76 opportunities in fields such as genomics, proteomics and bioinformatics, "*Biomarker qualification*" and "*Standards for microarray and proteomics-based identification of biomarkers*" were cited as the top two opportunities.

Microarray technology was identified by the FDA's Critical Path Initiative[2] as a key tool that holds "vast potential" for personalized medicine through the identification of biomarkers. In response to the FDA's CPI, scientists at the FDA's National Center for Toxicological Research (NCTR) formally launched the MicroArray Quality Control (MAQC) project[3] in order to address reliability concerns as well as other performance, standards, quality and data analysis issues (Shi *et al.* 2006).

Microarray gene expression profiling is being used for a variety of applications, two of which are (1) understanding general expression differences in various biological populations, classes, states, or conditions, which typically leads to the identification of lists of differentially expressed genes (DEGs) that distinguish populations and classes, and (2) the development of predictive models or classifiers that accurately predict outcomes of an *individual* based on a gene expression profile. These two types of applications have important ramifications and distinctions. In the first, information about a population or differences between populations is inferred. In the second, something about an individual member of a population is inferred or predicted. Although signatures can be used to classify individuals (e.g., assign or associate the individual with a subtype of a particular disease), MAQC-II is primarily focused on prediction of health outcomes based on microarray measurement of biological samples. These can putatively be used to predict response to treatment regimens, patient prognosis, recurrence of disease, survival, etc.

---

[2] http://www.fda.gov/oc/initiatives/criticalpath/

[3] http://www.fda.gov/nctr/science/centers/toxicoinformatics/maqc/

MAQC - I Findings: Microarrays Are Reproducible and Reliable

One important goal of the MAQC Phase I was to assess the best performance achievable with microarray technology under consistent experimental conditions so that future end users will have a benchmark to judge whether the quality of their microarray data is comparable. A major challenge to the microarray user is the existence of numerous options for analyzing the same data set, this is creating the reproducibility problem (Eisenstein 2006). Even though, the reproducibility has seldom been, but in the future should be used as a critical criterion to judge the performance of data analysis procedures.

The MAQC-I analyses (Shi *et al.* 2006) demonstrated that the apparent lack of reproducibility reported in previous studies (Marshall 2004; Tan *et al.* 2003) using microarray assays was likely caused, at least in part, by the common practice of ranking genes solely by a statistical significance measure, for example, *P*-values derived from simple *t*-tests, and selecting differentially expressed genes with a stringent significance threshold, a result that is consistent with a previous report. The gene lists in the MAQC study were much more concordant when fold change was used as the ranking criterion. In addition, widely used statistical methods such as ranking based on false discovery rate (FDR) values, t-test using SAM (significance analysis of microarray) did not appear to improve inter-laboratory or inter-platform reproducibility compared to fold change ranking. Importantly, non-reproducible gene lists could lead to inconsistent biological interpretations, for example, in terms of enriched GO (Gene Ontology) terms and pathways. Fold change ranking combined with a less stringent *P*-value cutoff was found

to yield more reproducible signature gene lists. The effect of various data normalization methods on the stability of lists of differentially expressed gene is greatly reduced when fold change is used for gene selection.

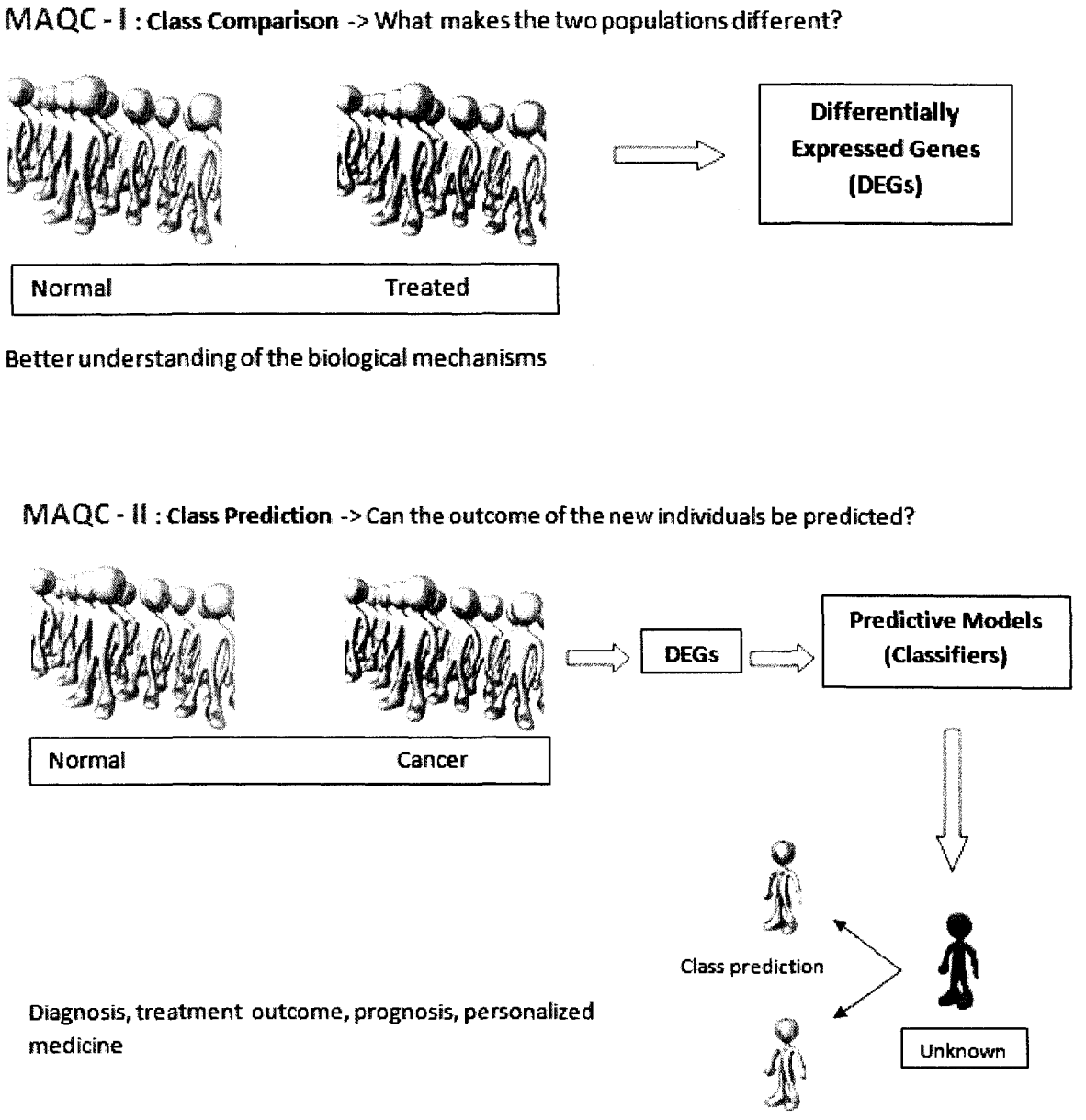


Figure 1: Schematic representations of the two major types of applications of microarray technology are being addressed in Phase I (top image) and Phase II (bottom image) of the MAQC project, *i.e.*, MAQC-I and MAQC-II, respectively.

Major findings of the first phase of the MAQC project were published in six research papers on the September 8, 2006 issue of *Nature Biotechnology* (Canales *et al.* 2006; Guo *et al.* 2006; Patterson *et al.* 2006; Shi *et al.* 2006; Shippy *et al.* 2006; Tong *et al.* 2006).

From MAQC-I to MAQC-II: To investigate the capabilities and limitations of microarrays in clinical applications such as disease diagnosis, prognosis, treatment outcome and personalized medicine, the MAQC Phase II (MAQC-II) has been launched to address technical and scientific issues involved in the development and validation of predictive models or classifiers (Figure 1). Multiple datasets were collected and distributed for independent analyses to the participating organizations, in which the University of Southern Mississippi (USM) group is also actively participating. The results will normally be evaluated at three different levels: within a single dataset via cross-validation, validation across one or more independent datasets from studies with the same (or similar) study objectives, and validation with blinded "prospective" samples.

Objectives of MAQC-II

The overall goal of MAQC-II is to comprehensively evaluate different approaches for the development and validation of predictive models or classifiers in clinical and preclinical (toxicogenomics) applications by applying the same set of approaches to a variety of datasets with diverse endpoints on which predictions are being developed. All predictions pertain to an individual patient endpoint.

Clinical Applications:

1. Understand the behavior of various prediction rules and gene selection methods that may be applied to microarray data sets to produce clinical outcome predictors: (a) Examine the influence of the number of variables (probes or probe sets) on prediction accuracy and robustness of the prediction result (in cross-validation and in independent and "prospective" validation); (b) Examine the influence of prediction rules (algorithms) on prediction accuracy and the robustness of prediction results (in cross-validation and in independent validation); and (c) Examine robustness of prediction results in the face of increasing experimental and artificial noise.

2. Identify and characterize the sources of variability in multi-gene prediction results including (a) Inter- and intra-laboratory variation in prediction results (in replicate experiments on the same platform); and (b) Cross-platform performance of prediction results (in replicate experiments on different platforms). Only NIEHS (National Institute of Environmental Health Sciences) is providing the datasets in two platforms (Affymetrix and Agilent) generated using the same experimental setup.

Preclinical (toxicogenomics) Applications:

The primary goal is to assess the reliability of models for the prediction of toxicity of new chemicals based on the microarray gene expression profiling. The entity to be predicted is the toxicological endpoint (e.g., the presence or absence of liver toxicity) for a chemical, and usually not for an individual animal. An important note is that in clinical applications, the entity to be predicted is usually an outcome of a subject (patient).

Design of MAQC – II

As part of the MAQC – II project, the FDA collected multiple datasets from academic and industrial organizations. These datasets were distributed to the participating organizations for independent analyses with available methodologies. We received these datasets as part of the participating analysis groups after signing the Confidential Information Disclosure and Transfer Agreements (CIDTA) from the USM contracts office with the corresponding data providers.

The project is divided into four working groups for better coordination and simplification for the participating organizations.

1. The Clinical Working Group (CWG) focuses on the datasets related to clinical applications. The USM has been part of this CWG from the initial stages.

2. The Toxicogenomics Working Group (TGxWG) focuses on the datasets related to toxicogenomics applications. The USM group has been part of this working group from the beginning of this group.

3. The Titrations Working Group (TitrationWG) focuses on the datasets from MAQC titration samples (including the MAQC-I Pilot data from 13 titration mixtures). The USM group is not a participant in this working group.

4. The Regulatory Biostatistics Working Group (RBWG) provides recommendations to the MAQC-II CWG and TGxWG on the process and criteria for evaluating the performance of predictive models and classifiers. This working group evaluates and ranks the data analysis plans (DAPs) of the participating analysis groups

whether they are within the acceptable statistical framework or not. Before May 2007, the USM group along with other groups first proposed the Standard Operating Procedures (SOPs) for each dataset separately as per the working groups teleconference discussions. But after discussions and comments from the RBWG on the loopholes in this approach (to have a separate plan for each dataset would be biased) in the face-to-face meeting in SAS, Cary, NC in May 2007. The statisticians' part of the RBWG unanimously recommended to the participating analysis groups to prepare a single comprehensive Data Analysis Plans (DAPs) for all the datasets from each group instead of separate plan for each dataset.

Datasets for Clinical and Toxicogenomics:

Datasets were identified for the purpose of evaluating

a) The performance of predictive models and classifiers (predictive signatures) and

b) The performance of different approaches and methodologies for algorithms commonly used in the development of predictive models and classifiers.

Datasets for Clinical Working Group:

Three diseases, namely breast cancer (BR) from the M.D. Anderson Cancer Center (MDACC), multiple myeloma (MM) from the University of Arkansas for Medical Sciences (UAMS) and neuroblastoma (NB) from the University of Cologne, Germany, were considered for more detailed examination for predictive modeling using microarray data. I will explain in more detail about these datasets in the Materials and Methods chapter.

Datasets for Toxicogenomics Working Group:

The goal of the TGxWG is to develop and compare methods for deriving genomic signatures from gene expression data that diagnose or predict toxicity of compounds in animal models. It should be noted that the individual entities that will be predicted or classified are individual chemicals, not individual animals.

Three datasets are selected to study under this working group. They are Lung Tumor in rats from Hamner Institute, Hepatocarcinogenicity in rats from Iconix and Overall necrosis score in mouse from NIEHS. These six datasets are explained in detail in the Materials part of the chapter III of this dissertation.

Prediction and Classification Algorithms

Numerous algorithms have been reported in the literature for developing prediction models and classifiers based on microarray gene expression data. The Regulatory Biostatistics WG (RBWG) suggested more commonly (and possibly appropriate) used methods to be evaluated with the MAQC-II datasets.

Supervised Learning and Classification Algorithms

Data mining is to extract implicit, previously unknown and potentially useful information from data (Witten *et al.* 2000). It is a learning process, achieved by building computer programs to seek regularities or patterns from data automatically. Machine learning provides the technical basis of data mining. One major type of learning we address in this dissertation is called classification learning, which is a generalization of concept learning. The task of concept learning is to acquire the definition of a general

category given a set of positive class and negative class training instances of the category (Mitchell *et al.* 1986). Thus, it infers a Boolean-valued function from the training instances. As a more general format of the concept learning, classification learning can deal with more than two class instances.

In practice, the learning process of classification is to find models that can separate instances in the different classes using the information provided by training instances. Thus, the models generated can be applied to classify a new unknown (blind) instance to one of those classes. Stating it in simpler words, given some instances of the positive class and some instances of the negative class, can we use them as a basis to decide if a new unknown instance is positive or negative (Mitchell *et al.* 1986)?. This kind of learning is a process from general to specific and is supervised because the class labels of training instances are clearly known.

In contrast to supervised learning is unsupervised learning, where there are no pre-defined classes or labels for training instances. The main goal of unsupervised learning is to decide which instances should be grouped together, or in other words, to form the classes. Sometimes, these two kinds of learning methods are used sequentially; supervised learning makes use of class information derived from unsupervised learning. This two-step strategy has achieved some success in the gene expression data analysis field (Alizadeh *et al.* 2000; Golub *et al.* 1999), where the unsupervised clustering methods were first used to discover classes (for example, subtypes of leukemia) so that supervised learning algorithms could be employed to establish classification models and assign a clinical outcome or phenotype to a newly coming instance.

Gene Expression Data Representation for Classification

In a typical classification task, data are represented as a table of samples (also known as instances). Each sample is described by a fixed number of features (also known as attributes, in our case, these were genes) and a label that indicates its class (Hall, 1998). For example, in studies of clinical outcome classification of cancer samples, gene expression data of m genes for n cancer samples is often summarized by an n x (m+1) table (X,Y) = (x_{ij}, y_i), where x_{ij} denotes the expression level of gene j in sample i, and y_i is the class or label (e.g., erpos in breast cancer) to which sample i belongs (i = 1,2,3,..., n and j = 1,2,..., m). The table (Table 1) below shows a sample dataset with three breast cancer samples.

Sample#	geneA	geneB	geneC	geneD	geneE	Class (label)
Sample1	2.004906	-1.20118	0.528131	1.05386	1.544994	erpos
Sample2	1.445852	-1.0127	1.116866	-0.90285	1.854096	erneg
Sample3	1.429597	0.567925	-0.19037	3.039217	-0.09884	erpos

Table 1: This table shows an example of gene expression data. There are three samples, each of which is described by 5 genes. The class label in the last column indicates the clinical endpoint of the sample.

Results Evaluation or Error Estimation

Evaluation is the key to making real progress in supervised classification (Witten *et al.* 2000). To evaluate the performance of classification algorithms, one way is to split samples into two sets, training samples and test samples. Training samples are used to build a learning model while test samples or external independent dataset (blind dataset) are used to evaluate the accuracy of the model. During validation, test samples or blind

dataset are supplied to the model, having their class labels "hidden", and then their predicted class labels assigned by the model are compared with their corresponding original class labels to calculate prediction accuracy. If two labels (actual and predicted) of a test sample are same, then the prediction for this sample is counted as a *success*; otherwise, it is an *error* (Witten *et al.* 2000). An often used performance evaluation term is *error rate*, which is defined as the proportion of errors made over a whole set of test samples. In some cases, we simply use the number of errors to indicate the performance. Note that, although the error rate on test samples is often more meaningful to evaluate a model, the error rate on the training samples is nevertheless useful to know as well since the model is derived from them.

	Predicted		
	Neg	Pos	
Neg	TN	FP	
Pos	FN	TP	

Figure 2: Confusion matrix for two-class classification problem

Consider the confusion matrix illustrated in the above figure (Figure 2) of a two-class ('Pos' and 'Neg') problem. The *true positive* (*TP*) and *true negative* (*TN*) are correct classifications in samples of each class, respectively. A *false positive* (*FP*) is when a 'Neg' class sample is incorrectly predicted as a 'Pos' class. A *false negative* (FN) is when a 'Pos' class sample is incorrectly predicted as a 'Neg' class. Then each element of a confusion matrix shows the number of test samples for which the actual class is the row and the predicted class is the column. Thus, the error rate is just the number of false positives and false negatives divided by the total number of test samples (*i.e.*, error rate = (FP+FN)/(TN+TP+FP+FN)).

Error rate is a measurement of overall performance of a classification algorithm (also known as a classifier); however, a lower error rate does not necessarily imply better performance on a target task. For example, there are 10 samples in class 'Pos' and 90 samples in class 'Neg'. Suppose, if TP = 5 and TN = 85, then FP = 5, FN = 5 and the error rate is 10%. However, only 50% of the samples are correctly classified in class 'Pos'. So, this is not a perfect evaluation metric in all cases. To more impartially evaluate the classification results, some other evaluation metrics are constructed.

1. True positive rate (TP rate) = TP/(TP+FN), also known as *recall* or *sensitivity*, measures the proportion of samples in class 'Pos' that are correctly classified as class 'Pos'.

2. True negative rate (TN rate) = TN/(FP+TN), also known as *specificity*, measures the proportion of samples in class 'Neg' that are correctly classified as class 'Neg'.

3. False positive rate (FP rate) = FP/(FP+TN) = 1-*specificity*.

4. False negative rate (FN rate) = FN/(TP+FN) = 1-sensitivity.

5. Another evaluation metric in the classification studies is Matthews Correlation Coefficient (MCC). We used this as our priority metric in determining the candidate model for each end point as per the RBWG recommendation due to more unbalanced classes for each endpoints in our study.

MCC takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very

different sizes (significantly unbalanced). It returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 an average random prediction and -1 the worst possible prediction (Baldi *et al.* 2000; Matthews 1975). Other measures, such as the proportion of correct predictions, are not useful when the two classes are of very different sizes. For example, assigning every object to the larger set achieves a high proportion of correct predictions, but is not generally a useful classification.[4]

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

In classification, it is a normal situation that along with a higher TP rate, there comes a higher FP rate and same to the TN rate and FN rate. Thus, the receiver operating characteristic (ROC) curve was invented to characterize the tradeoff between TP rate and FP rate (Zweig *et al.* 1993). The ROC curve plots TP rate on the vertical axis against FP rate on the horizontal axis. With an ROC curve of a classifier, the evaluation metric will be the area under the ROC curve. The larger the area under the curve (AUC) (the more closely the curve follows the left-hand border and the top border of the ROC space), the more accurate the test. Thus, the ROC curve for a perfect classifier has an area of 1. The expected curve for a classifier making random predictions will be a line on the 45 degree diagonal and its expected area is 0.5. Please refer to Figure 3 (figure slightly modified from the courtesy image by Indon, 2007)[5] for a sample ROC curve.

---

[4] http://en.wikipedia.org/wiki/Matthews_Correlation_Coefficient

[5] http://en.wikipedia.org/wiki/Image:ROC_space.png

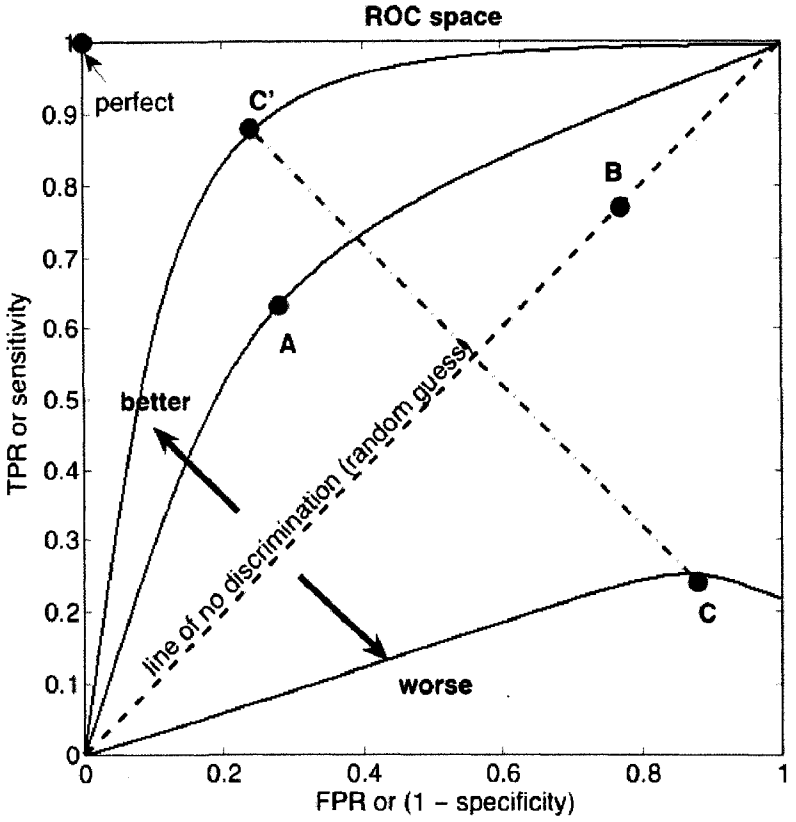


Figure 3: A sample ROC curve. The dotted line on the 45 degree diagonal is the expected curve for a classifier making random predictions.

Error estimation methods

If the number of samples for training and testing is limited, a standard way of predicting the error rate of a learning technique is to use stratified *k*-fold cross validation (*k*-fold CV). In *k*-fold cross validation, first, a full data set is divided randomly into *k* disjoint subsets of approximately equal size, in each of which the class is represented in approximately the sample proportions as in the full dataset (Witten *et al.* 2000). Then the above process of training and testing will be repeated *k* times on the *k* data subsets. In each iteration, (1) one of the subsets is held out in turn, (2) the classifier is trained on the

remaining k-1 subsets to build classification model, (3) the classification error of this iteration is calculated by testing the classification model on the holdout set (Figure 4). Finally, the k numbers of errors are added up to yield an overall error estimate. Obviously, at the end of cross validation, every sample has been used exactly once for testing.

A widely used selection for k is 10. Why 10? "Extensive tests on numerous different data sets, with different learning techniques, have shown that ten is about the right number of folds to get the best estimate of error, and there is also some theoretical evidence that backs this up"(Witten *et al.* 2000). Although 10-fold cross validation has become the standard method in practical terms, a single 10-fold cross validation might not be enough to get reliable error estimate (Witten *et al.* 2000). The reason is that, if the seed of the random function that is used to divide data into subsets is changed, the cross validation with the sample classifier and data set will often produce different results. Thus, for a more accurate error estimate, it is suggested to repeat the 10-fold cross validation process ten times and average the error rates. This is called 10-fold cross validation with ten iterations and naturally, it is a computation-intensive undertaking. First we used the 10-fold CV, but based on the recommendations from RBWG in 8th face-to-face MAQC meeting, we choose to perform 5-fold CV with ten iterations because the Hamner dataset is small and not strong data to use 10-fold with ten iterations. This avoids the over fitting problem.

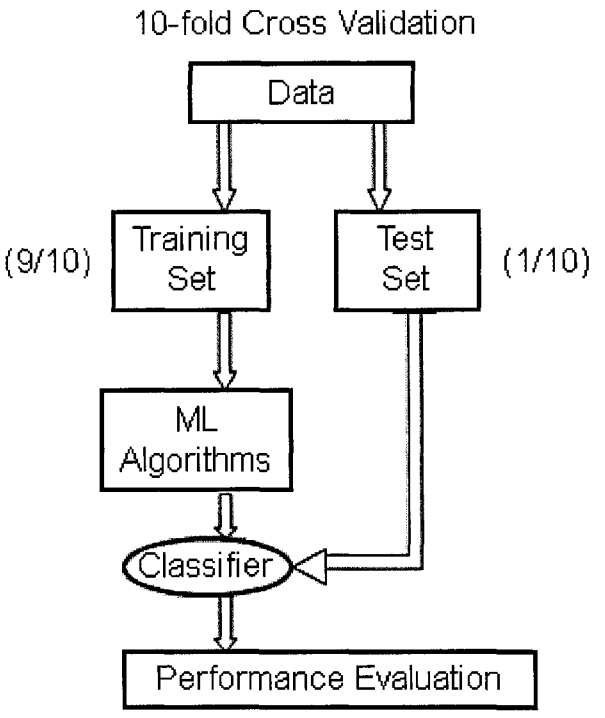


Figure 4: A Graphical depiction of 10-fold cross validation

Instead of running cross validation ten times, another approach for a reliable results is called *leave-one-out* cross validation (LOOCV). LOOCV is simply n-fold cross validation, where n is the number of samples in the full data set. In LOOCV, each sample in turn is left out and the classifier is trained on all the remaining n-1 samples. Classification error for each iteration is judged on the class prediction for the holdout sample, success or failure. Different from k-fold ($k < n$) cross validation, LOOCV makes use of the greatest possible amount of samples for training in each iteration and involves no random shuffling of samples.

Classification Algorithms

There are various ways to find models that separate two or more data classes, *i.e.* to do classification. Models derived from the same sample data can be very different from one classification algorithm to another. As a result, different models represent the knowledge learned in different formats as well. For example, decision trees represent the knowledge in a tree structure; instance-based algorithms, such as nearest neighbor, use the instances themselves to represent what is learned; Naïve Bayes method represents knowledge in the form of probabilistic summaries. In this section, we will describe a number of classification algorithms that have been used in this project, including Naïve Bayes, Support Vector Machines (SVM) and Voted Perceptron methods.

Support Vector Machines (SVM)

Support vector machines (SVM) is a kind of a blend of linear modeling and instance-based learning (Witten *et al.* 2000), which uses linear models to implement nonlinear class boundaries. It originates from research in statistical learning theory (Vapnik, 1995). An SVM selects a small number of critical boundary samples from each class of training data and builds a linear discriminant function (also called maximum margin hyperplane) that separates them as widely as possible. The selected samples that are closest to the maximum margin hyperplane are called support vectors. Then the $f(T)$ discriminant function

for a test sample T is a linear combination of the support vectors and it is constructed as:

$$f(T) = \sum_i \alpha_i y_i (X_i . T) + b$$

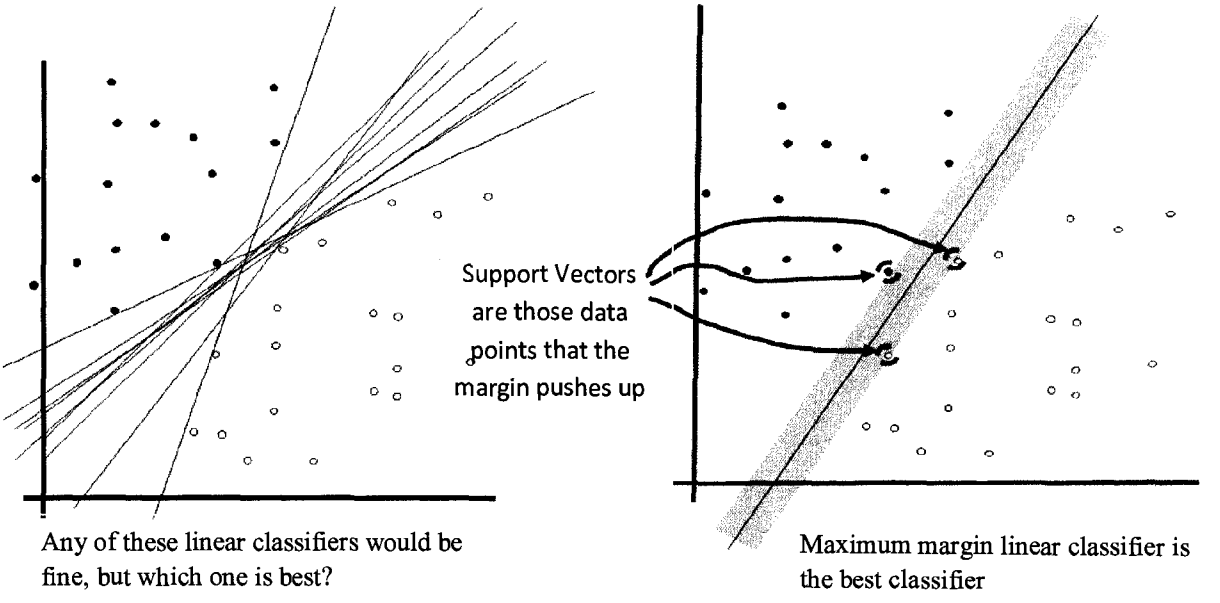


Figure 5: Graphical representation of Support Vector Machines concept

where the vectors X_i are the support vectors, y_i are the class labels (which are assumed to have been mapped to 1 or -1) of X_i, vector T represents a test sample. $(X_i . T)$ is the dot product of the test sample T with one of the support vectors X_i. α_i and b are numeric parameters (like weights) to be determined by the learning algorithm.

In the case that no linear separation is possible, the training data will be mapped into a higher-dimensional space $\hat{H}$ and an optimal hyperplane will be constructed there. The mapping is performed by a kernel function $K(.,.)$ which defines an inner product in $\hat{H}$. Different mappings construct different SVMs (Figure 5). When there is a mapping, the discriminant function is given like below which is a representation of a linear SVM.

$$f(T) = \sum_i \alpha_i y_i K(X_i, T) + b$$

An SVM is largely characterized by the choice of its kernel function. There are two types of widely used kernel functions; *polynomial* kernel and Gaussian *radial basis function* (RBF) kernel (Burges, 1998).

1. A polynomial kernel is $K(X_1, X_2) = (X_1 . X_2 + 1)^d$, the value of power d is called degree and generally is set as 1, 2 and 3. Particularly, the kernel becomes a linear function if $d = 1$. It is suggested to choose the value of degree starting with 1 and increment it until the estimated error ceases to improve. However, it has been observed that the degree of a polynomial kernel plays a minor role in the final results (Santos *et al.* 2002) and sometimes, linear function performs better than quadratic and cubic kernels due to over-fitting of the latter kernels.

2. An RBF kernel has the form $K(X_1, X_2) = exp(-\frac{\|x_1 - x_2\|^2}{2\sigma^2})$, where σ is the width of the Gaussian. The selection of parameter σ can be conducted via cross validation or some other manners. When using SVM with RBF kernel on gene expression data analysis, Brown group (Brown *et al.* 2000) set σ equal to the median of the Euclidean distances from each positive samples (sample with class label as 1) to the nearest negative sample (sample with class label as -1).

Besides polynomial kernel and Gaussian RBF kernel, other kernel functions include sigmoid kernel (Schölkopf *et al.* 2002), locality-improved kernel (Zien *et al.* 2000) and so on.

In order to determine parameters α and b in $f(T) = \sum_i \alpha_i y_i K(X_i, T) + b$, the construction of the discriminant function finally turns out to be a constrained quadratic problem on maximizing the Lagrangian dual objective function (Weston *et al.* 2001).

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(X_i, X_j)$$

under constraints

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \ \alpha_i \geq 0, \qquad (i = 1,2, \ldots, n)$$

where n is the number of samples in training data. However, the quadratic programming (QP) problem in the above equation cannot be solved easily via standard techniques since it involves a matrix that has a number of elements equal to the square of the number of training samples.

Sequential Minimal Optimization (SMO)

To efficiently find the solution of the above QP program, Platt developed the *sequential minimal optimization* (SMO) algorithm (Platt *et al.* 1998); one of the fastest SVM training methods. Like other SVM training algorithms, SMO breaks the large QP problem into a series of smaller possible QP problems. Unlike other algorithms, SMO tackles these small QP problems analytically, which avoids using a time-consuming numerical QP optimization as an inner loop. The amount of memory required by SMO is linear with number of training samples (Platt *et al.* 1998). Therefore it is good for large

datasets, as in our case, to take advantage of computationally inexpensive aspect. SMO has been implemented into Weka, a data mining software package, which we used in this study (Witten *et al.* 2000).

SVMs have shown to perform well in multiple areas of biological analysis, such as detecting remote protein homologies, recognizing translation initiation sites (Liu *et al.* 2003; Zeng *et al.* 2002; Zien *et al.* 2000), and prediction of molecular bioactivity in drug design (Weston *et al.* 2003). Recently, more and more bioinformaticians employ SVMs in their research on evaluating and analyzing microarray expression data (Brown *et al.* 2000; Furey *et al.* 2000; Yeoh *et al.* 2002). SVMs have many mathematical features that make them attractive for gene expression analysis, including their flexibility in choosing a similarity function, sparseness of solution when dealing with large data sets, the ability to handle large feature spaces, and the ability to identify outliers (Brown *et al.* 2000).

In many practical data mining applications, success is measured more subjectively in terms of how acceptable the learned description rules, decision trees, or whatever are to a human user (Witten *et al.* 2000) . This measurement is especially important to biomedical applications such as cancer studies where comprehensive and correct rules are crucial to help biologists and doctors understand the diseases (Huiqing, 2004).

Naïve Bayes

In machine learning, we are interested in determining the best hypothesis $h(x)$ from space H, based on the observed training data x. Best hypothesis is almost equal to *most probable* hypothesis, given the data x with any initial knowledge about the prior probabilities of the various hypothesis in H (Jaynes 2003, Richard *et al.* 2001).

Bayes theorem provides a way to calculate,

(i) the probability of a hypothesis based on its prior probability Pr(*h(x)*)

(ii) the probabilities of the observing various data given the hypothesis Pr(*x*|h)

(iii) the probabilities of the observed data Pr(*x*)

We can calculate the posterior probability *h(x)* given the observed data *x*, Pr(*h(x)*|*x*)

using *Bayes theorem.*

$$\Pr(h(x)|x) = \frac{\Pr(x|h(x))\Pr(h(x))}{\Pr(x)}$$

Naïve Bayes (NB) is a classification model obtained by applying a relatively simple method to a training dataset (Mitchell *et al.* 1986). A NB classifier calculates the probability that a given instance (example) belongs to a certain class. It makes the simplifying assumption that the features constituting the instance are conditionally independent, given the class.

Given an example X, described by its $(x_1, \ldots, x_n)$ feature vector we are looking for a class *C* that maximizes the likelihood: $P(X|C) = P(x_1, \ldots, x_n|C)$

The (naïve) assumption of conditional independence among the features, given the class, allows us to express this conditional probability $P(X|C)$ as a product of simpler probabilities: $P(X|C) = \prod_{i=1}^{n} P(x_i|C)$. We used the Weka program to train the NB classifier.

Voted Perceptron

The voted perceptron algorithm proposed by Freund *et al.* (1999) is based on the well known perceptron algorithm of Rosenblatt (1958, 1962) and a transformation of

online learning algorithms to batch learning algorithms developed by Helmbold and Warmuth (1995). Moreover, they used the kernel functions proposed by Aizerman, Braverman and Rozonoer (1964), to run their algorithm efficiently in very high dimensional spaces. This algorithm and its analysis involve little more than combining these three known methods.

Their studies indicate that the use of kernel functions with the perceptron algorithm yields a dramatic improvement in performance, both in test accuracy and in computation time. In addition, they found that, when training time is limited, the voted-perceptron algorithm performs better than the traditional perceptron algorithm.
I discussed about the algorithms which I used for my final classification analysis in this project. I have ignored the other algorithms in this discussion which we studied initially for preliminary studies like KNN, Random Forest, J48 etc.

Feature Selection Algorithms

A well known problem in classification (in general machine learning) is to find ways to reduce the dimensionality of the feature space to overcome the risk of over-fitting especially when we are dealing with gene expression data. Data over-fitting happens when the number of features (genes) is large ("curse of dimensionality") and the number of training samples is comparatively small ("curse of data set sparsity"). In such a situation, a decision function can perform very well on classifying training data, but does poorly on test samples. Feature selection is concerned with the issue of distinguishing signal from noise in data analysis.

Categorization of feature selection algorithms

Feature selection techniques can be categorized according to a number of criteria (Hall *et al.* 2003). One popular categorization is based on whether the target classification algorithm will be used during the process of feature evaluation. A feature selection method, that makes an independent assessment only based on general characteristics of the data, is named "filter" (Witten *et al.* 2000); while, on the other hand, if a method evaluates features based on accuracy estimates provided by certain classification learning algorithm which will ultimately be employed for classification, it will be named as "wrapper" (Kohavi *et al.* 1997, Witten *et al.* 2000). With wrapper methods, the performance of a feature subset is measured in terms of the learning algorithm's classification performance using just those features (see Figure 6 below).

The classification performance is estimated using the normal procedure of cross validation, or the bootstrap estimator (Witten *et al.* 2000). Thus, the entire feature selection process is rather computation intensive. For example, if each evaluation involves a 10-fold cross validation, the classification procedure will be executed 10 times. For this reason, wrappers do not scale well to data sets containing many features (Hall *et al.* 2003). Besides, wrappers have to be re-run when switching from one classification algorithm to another.

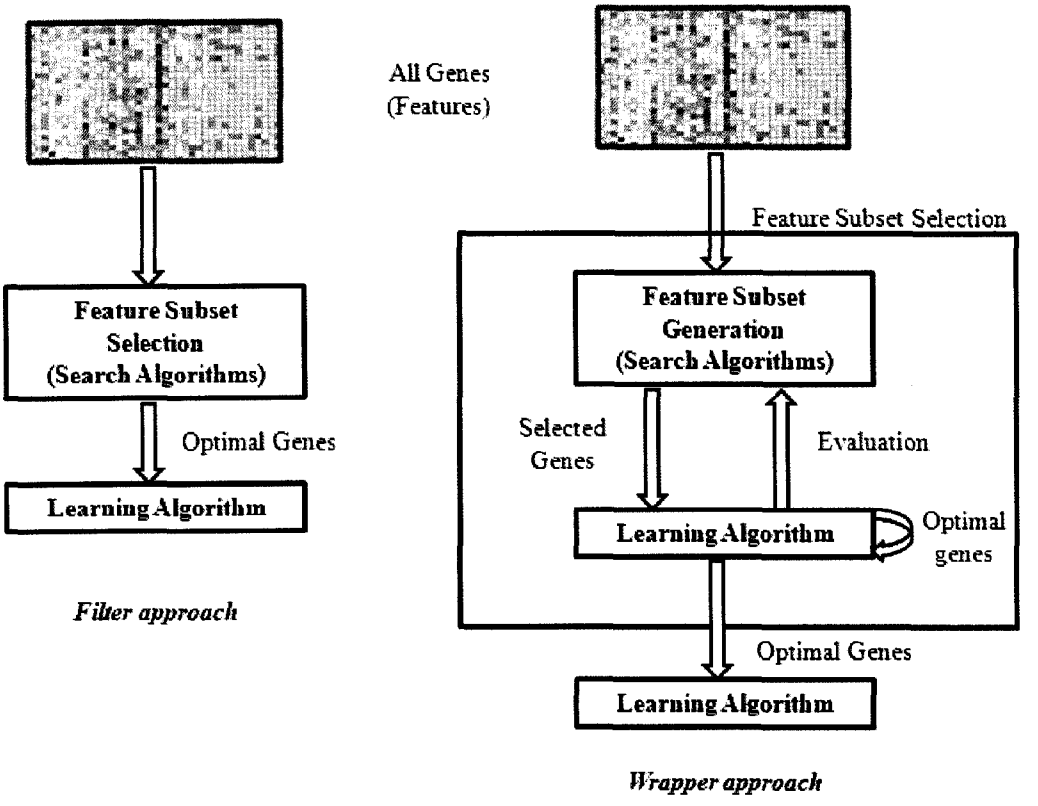


Figure 6: Graphical depiction of two feature selection (Filter and Wrapper) approaches.

In contrast to wrapper methods, filters operate independently of any learning algorithm and the features selected can be applied to any learning algorithm at the classification stage. Filters have been proven to be much faster than wrappers and hence, can be applied to data sets with many features (Hall *et al.* 2003). Since the biological data sets discussed in the later chapters of this dissertation often contain a huge number of features (e.g., gene expression profiles), we not only concentrate wrapper but also filter methods.

Another taxonomy of feature selection techniques is to separate algorithms evaluating the worth or merit of a subset features from those of individual features. There

are some other dimensions to categorize feature selection methods. For example, some algorithms can handle regression problem, that is, the class label is numeric rather than a discrete valued variable; and some algorithms evaluate and rank features independently from class, i.e., unsupervised feature selection (Witten *et al.* 2000). We will restrict our study to the data sets with discrete class label since this is the case of the biological problems analyzed in later chapters of this dissertation, though some algorithms presented can be applied to numeric class label as well.

Feature selection algorithms

There are various ways and algorithms to conduct feature selection. We studied five feature selection methods in this project; they are T-test, χ^2 statistical measure, gain ratio, information gain and Relief-F.

T-test

Highly consistent with the well-known ANOVA principle, a basic concept for identifying a relevant feature from an irrelevant one is the following: if the values of a feature in samples of class '*A*' are significantly different from the values of the same feature in samples of class '*B*', then the feature is likely to be more relevant than a feature that has similar values in '*A*' and '*B*'. More specifically, in order for a feature *f* to be relevant, its mean value in '*A*' should be significantly different from its mean value in '*B*'(Golub *et al.* 1999).

The classical t-statistic is constructed to test the difference between means of two groups of independent samples. So, if samples in different classes are independent, the t-

statistic can be used to find features that have big difference in mean level between the two classes. These features can be then considered to have ability to separate samples between different classes (Nguyen *et al.* 2002). We tested this method in the initial stages but did not perform well with over the other methods studied.

χ^2 - statistical measure

χ^2 measure evaluates features individually by measuring the χ^2 - statistic with respect to the class. Different from the preceding methods, χ^2 measure can only handle features with discrete values, χ^2 measure of a feature *f* with *w* discrete values is defined as,

$$\chi^2(f) = \sum_{i=1}^{w} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

where *k* is the number of classes, A_{ij} is the number of samples with *i*th value of *f* in *j*th class. E_{ij} is the expected frequency of A_{ij} and

$E_{ij} = R_i * C_j / n$

R_i is the number of samples having *i*th value of *f*, C_j is the number of samples in the *j*th class and *n* is the total number of samples.

We consider a feature f_j to be more relevant than a feature f_l $(l \neq j)$ if $\chi^2(f_j) > \chi^2(f_l)$.

Obviously, the worst χ^2 value is 0 if the feature has only one value. The degree of freedom of the χ^2 – statistic measure is *(w-1) * (k – 1)* (Liu *et al.* 1995). With

the degree of freedom known, the critical value for certain significant level can be found from the appendix tables provided in most statistics books.

To apply χ^2 measure to numeric features, a discretization preprocessing has to be taken. The most popular technique in this area is the state-of-art supervised discretization algorithm developed by Fayyad and Irani (Fayyad *et al.* 1993) based on the idea of entropy. At the same time, feature selection can be also conducted as a by-product of discretization.

Information gain and Information gain ratio

Information gain is simply the expected reduction in entropy by partitioning the samples according to this feature that it is the amount of information gained by looking at the value of this feature. More precisely, the information gain *Gain(f,S)* of a feature *f*, relatively to a set of samples *S*, defined as,

$$Gain(f,S) = Ent(S) - Ent(f,T_f,S)$$

where *Ent(S)* can be calculated from

$$Ent(S) = \sum_{i=1}^{k} -p_i * log_2 p_i$$

and *Ent(f, T_f, S)* is the class entropy of the feature (for a numeric feature *f*, T_f is the best partition to *f* 's value range under certain criteria, such as MDL principle in discretization). Since *Ent(S)* is a constant once *S* is given, the information gain and entropy measures are equivalent when evaluating the relevance of a feature. In contrast to the rule " the smaller the class entropy value, the more important the feature is" that is

used in entropy measure, we consider a feature f_j to more relevant than a feature f_l $(l \neq j)$ if $Gain(f_j, S) > Gain(f_l, S)$ (Xing *et al.* 2001, Quinlan 1986).

However, there is natural bias in the information gain measure - it favors features with many values over those with few values. An extreme example is a feature having different values in different samples. Although the feature perfectly separates the current samples, it is a poor predictor on subsequent samples. One refinement measure that has been used successfully is called *information gain ratio*. The gain ratio measure penalizes features that with many values by incorporating amount of split information, which is sensitive to how broadly and uniformly the feature splits the data (Mitchell *et al.* 1986).

$$Ent(S) = \sum_{i=1}^{w} \frac{|S_i|}{|S|} * log_2 \frac{|S_i|}{|S|}$$

where S_1 through S_w are the w subsets of samples resulting from partitioning of S by w-values discrete or w-value-interval numeric feature f. Then, the gain ratio measures is defined in terms of the earlier information gain measure and this split information, as follows:

$$GainRatio\,(f,S) = \frac{Gain(f,S)}{Split\ Information\,(f,S)}$$

Note that split information is actually the entropy of S with respect to the values of feature f and it discourages the selection of features with many values (Mitchell *et al.* 1986). For example, if there is total number of n samples in S, the split information of a feature f_1, which has different values in different samples, is $log_2 n$. In contrast, a Boolean feature f_2 that splits the same n samples exactly in half will have split

information of 1. If these two features produce the equivalent information gain, then clearly feature f_2 will have a higher gain ratio measure. Generally, a feature f_j is considered to be more significant than a feature f_l ($l \neq j$) if *GainRatio (f, S)* > *GainRatio(f_l, S)*. When using gain ratio measure (or information gain measure) to select features, we sort the values of gain ratio (information gain) in the descending order and consider those features with highest values.

ReliefF

The key idea of ReliefF is to estimate attributes according to how well their values distinguish among the instances that are near to each other. For that purpose, given an instance, ReliefF searches for its two nearest neighbors: one from the same class (called *nearest hit*) and the other from a different class (called *nearest miss*). The original algorithm of ReliefF (Kira *et al.* 1992) randomly selects *n* training instances, where *n* is the user-defined parameter.

Chapter summary

In this chapter, I introduced the concept of classification in data mining as well as the ways to evaluate the classification performance. I presented in detail some of classification algorithms — putting the emphasis on several methods used in the final analysis like SVMs, SMO and Naïve Bayes. We also used KNN, Random forest, J48 algorithms to compare and contrast with the above algorithms in our preliminary studies which are addressed in later chapter about these studies.

Also the basic concepts of feature selection algorithms and the differences between filter and wrapper approaches were discussed. Also, the details about the feature selection algorithms we studied like t-test, χ^2 statistic measure, Information gain, Gainratio and ReliefF were explained.

CHAPTER II

MATERIALS AND METHODS

Methods

One of the important recent breakthroughs in experimental molecular biology is microarray technology. This novel technology allows the monitoring of expression levels in cells for thousands of genes simultaneously and has been increasingly used in cancer research (Alizadeh *et al.* 2000; Alon *et al.* 1999; Golub *et al.* 1999) to understand more of the molecular variations among tumors so that a more reliable classification becomes possible.

There are two main types of microarray systems: the cDNA microarrays developed in the Brown and Botstein Laboratory at Stanford (DeRisi *et al.* 1997) and the high-density oligonucleotide chips from the Affymetrix company (Lockhart *et al.* 1996). The cDNA microarrays (two-color) are also known as 'spotted' arrays, popularly called as 'agilent' prepared from Agilent company (Miller *et al.* 2002), where the probes are mechanically deposited onto modified glass microscope slides using a robotic array machine. Oligonucleotide chips are synthesized *in silico* (e.g., via photolithographic synthesis as in Affymetrix GeneChip arrays) are also popularly called as 'single channel' arrays. For a more detailed introduction and comparison of the biology and technology of the two systems, please refer to Harrington *et al.* (2000).

Gene expression data from DNA microarrays are characterized by many measured variables (genes or features) on only a few observations (experiments or

samples), although both the number of experiments and genes per experiment are growing rapidly (Nguyen *et al.* 2002). The number of genes on a single array is usually in the thousands while the number of experiments is only a few tens or hundreds. There are two different ways to view data: (1) data points as genes, and (2) data points as samples (e.g., patients). In the way (1), the data are presented by expression levels across different samples, thus there will be a large number of features and a small number of samples. In the way (2), the data is represented by expression levels of different genes, thus the case will be a large number of samples with a few attributes. In this dissertation, all the discussions and studies on gene expression profiles are based on the format of data presentation that is data points as genes or features.

Microarray experiments raise many statistical questions in many diversified research fields, such as image analysis, experimental design, cluster and discriminant analysis, and multiple hypothesis testing. The main objectives of most microarray studies can be broadly classified into one of the following categories: class comparison, class discovery, or class prediction (Miller *et al.* 2002).

Class comparison is to establish whether expression profiles differ between classes. If they do, which genes are differentially expressed between the classes, i.e. gene identification. For example, which genes are useful to distinguish tumor sample from non-tumor ones. This is the typical microarray analysis we will perform every day.

Class discovery is to establish subclusters or structure among specimens or among genes. For example, to define previously unrecognized tumor subtypes.

Class prediction is to predict a phenotype using information from a gene expression profile (Miller *et al.* 2002). This includes assignment of malignancies into known classes (tumor or non-tumor) or tumor samples into already discovered subtypes, prediction of patients outcome such as which patients are likely to experience severe drug toxicity versus who will have none, or which breast cancer patients will relapse within five years of treatment versus who will remain disease free.

In this dissertation, we will focus on the class comparison and class prediction. For these two tasks, supervised analysis methods that use known class information are most effective (Miller *et al.* 2002). In practice, feature selection techniques are used to identify discriminatory genes while classification algorithms are employed to build models on training samples and predict the phenotype of blind test cases.

Preprocessing of Expression Data

Despite optimal techniques to ensure RNA quality, some amount of non-biology-related variation remains; thus, preprocessing of the microarray data is essential before analysis can be initiated. Several critical preprocessing techniques have been developed to enhance the validity of microarray analyses. Based on the characteristics of the experimental data, the normal preprocessing steps include identification of outlier arrays, scale transformation, data normalization, missing value management, batch effect correction, replicate handling and so on (Herrero *et al.* 2003).

Identification of Outlier Samples

Array outliers are due to excessive chip-to-chip variation and may be the result of improper hybridization errors that create smudges, scratches or cross-hybridization to the

microarray (Han *et al.* 2004). These arrays may have disproportionately high or low intensities for individual probe sets due to non-specific or mismatch mRNA binding. These changes can sometimes be observed on the DAT file (the image file, not the CEL file) of the microarray during visual inspection. It is a better quality control procedure to remove the samples from the analysis when their intensities do not match the overall tendencies for the same probe sets in the group.

We used dChip[6] analysis to find out the array outliers. It does the probe summarization and high-level analysis of gene expression data through model based approach by applying Model Based Expression Index (MBEI) algorithm (Li *et al.* 2001). This model is based on a balanced hybridization of all probe sets, in $\log_2$ format. When the uniformity of hybridization exceeds the model limits, the microarray is identified as an outlier. For this algorithm if the standard error for a probe set is more than three times larger than the other probe sets on the microarray it is identified as an array outlier. If greater than 5% of the probe sets on an individual microarray chips are identified as outliers, dChip flags the entire microarray as a potential outlier (Li *et al.* 2001). MAQC asked all the analysis groups to vote each array in the dataset either as outlier (1), moderate outlier (0.5) and non-outlier (0). We gave our voting on each array of the datasets using this method and also by visualizing the box plot and PCA distribution of arrays in the dataset.

[6] www.dchip.org

Data Normalization

The purpose of normalization is to adjust the effects which arise from variations in the microarray technology rather than from biological differences between the RNA samples or between the printed probes, so that data from different chips can be directly compared. No step in the microarray hybridization process can be perfectly controlled. The quantity of RNA in a sample varies slightly from chip to chip. Even if the exact same sample is used on each of several chips, there will be chip to chip differences in the overall distribution of probe intensity values (Bolstad *et al.* 2003; Irizarry *et al.* 2006). In microarray analysis, the normalization methods vary depending on the technology of the arrays we used. In this project, we have total six datasets, among these five datasets are Affymetrix single color technology arrays and the Neuroblastoma dataset is Agilent two colored customized array.

We used MAS5 (Microarray Suite Ver. 5) probe-set summarization and normalization algorithm from Affymetrix (Affymetrix, 2002) for affymetrix datasets (exception to Iconix data, we used Median scale 1000 normalization, because the array is customized not the standard array supplied by Affymetrix). There is one specific reason for selecting this normalization for this study is that we can accommodate the external or blind validation dataset in to the classification system without altering the developed model using training dataset. I mean, the normalization step should be independent and should not affect the coming external validation dataset in the future. It can be possible when the normalization method is done within array instead of across the arrays. MAS5 algorithm works within array and the other normalization methods like RMA, MBEI

(dChip) and PLIER does the normalization between the arrays. So, MAS5 algorithm fits well to our study and also MAQC recommendes this normalization for the same reason.

MAS5 algorithm

The MAS5 algorithm uses the TukeyBiweight algorithm, which reweights the differences depending on how far the expression values are from the median, and discards any differences which are more than five times the median absolute distance from the median (Affymetrix, 2002). The MAS5 algorithm also replaces the MM (mismatch) value (MAS4 considers MM values) with a value that is always less than the PM (perfect match) value, calculating what is called an ideal mismatch (IM) in this situation. Ideal mismatch (IM) intensity value calculated from MM value subtracted from PM value (IM is never bigger than PM).

If MM < PM then IM = MM;

If MM > PM then IM = PM – correction value;

Robust mean of probe set values are taken using TukeyBiweight algorithm. In this algorithm the mean is calculated to identify center of data. Distance of each data point from the mean is calculated (Affymetrix, 2002). This distance determines how each value is weighted in the average i.e. outliers far from the median contribute less to the average.

And the signal is calculated using;

$$Signal = TukeyBiweight\{log_2(PM_j - IM_j)\}$$

We implemented this normalization method on the datasets using affy packages in Bioconductor version 2.1[7] in R 2.5.0[8] frame work.

In the case of Neuroblastoma dataset, which is an Agilent platform (two color), we used mean scale normalization, a simple scaling normalization method. In the case of Iconix dataset, the normalization is median scale 1000 even though it is single color data but the platform provider is GE Healthcare, not as usual from Affymetrix.

Data Transformation

Missing value transformation: One of the characteristics of the gene expression profile is the presence of missing values in the data set. There are diverse reasons that cause missing values, including insufficient resolution, image corruption, or simply due to dust or scratches on the slide (Troyanskaya *et al.* 2001). In practice, missing data also occur systematically as a result of the robotic methods used to create them. Unfortunately, many data analysis algorithms require a complete matrix of gene array values as input (Troyanskaya *et al.* 2001). For example, standard hierarchical clustering methods and K-means clustering are not robust to the excess of missing values since the calculations in these algorithms are based on a distance matrix. Even with a few missing values, they may lose effectiveness. More strictly, some methods like principal components analysis (PCA) can not deal with missing values at all. Therefore, methods for imputing missing data are needed, not only to minimize the effect of incomplete data

[7] www.bioconductor.org

[8] www.cran.org

on further analyses, but also to increase the range of data sets to apply learning algorithms.

There are some general solutions to impute missing values. Here, we list five commonly used strategies: (1) filling blanks with zeros; (2) replacing with the gene's average expression levels over all experiments; (3) replacing with the median of the gene's expression levels over all experiments; (4) singular value decomposition (SVD); (5) using weighted KNN imputation method. The KNN based method is to use the *k*-nearest neighbours (KNN) to estimate the missing values, where a user is defined parameter.

Both weighted KNN and SVD-based techniques surpass the commonly used simple average method (Troyanskaya *et al.* 2001). This conclusion is very natural since the winning methods take advantage of the correlation structure of the data to estimate missing expression values. In these two methods, we used KNN based missing value imputation method. For this purpose we used ArrayAssist (Stratagene Inc.) package installed in the linux computer in the lab.

Log transformation

We transformed the normalized raw expression data into $\log_2$ transformation for better graphical presentation of the data and to continue further analysis with this transformed data.

Quality Control Check

Filtered genes based on flag values: We filtered the bad quality spots or probe sets based on the flag values Present (P) and Marginal (M) calls, we excluded the Absent (A)

calls. We also excluded the genes which have either P or M calls in less than half of the samples in the dataset. The remaining genes after filtering based on flag values are considered for further analysis.

Quality check with box-plots: We checked the quality of the normalized data using the box plot distribution.

Checking Batch Effect

Batch effects are observed when the overall intensity of a batch of microarrays more closely resembles the batch than the rest of the group, and this tendency may add enough noise in the analysis that errors are elevated. Due to the technical limitation that all samples cannot be processed simultaneously and must be run in batches, batch effects can be a potential confounding factor. This happens when the number of microarrays in a study makes it impossible to hybridize each sample to a microarray at the same time, by the same technologist, at the same location or with the same lot number of reagents or equipment. In our case especially with toxicogenomic datasets, which were designed to study the effect of chemical compounds on animal models over the period of years. This increases the chance of batch effect in the dataset either by time or the technology used. This type of errors ultimately generates false interpretations at the end of the analysis of large amount of information.

We checked for the batch effect in the dataset using unsupervised hierarchical clustering between the samples and Principal Component Analysis (PCA). We used two types of methods to correct these batch effects. One is Combat, a new function in the

bioconductor works based on empirical Bayes framework (Johnson *et al.* 2007), but it requires the class label information to correct the batch. The other method is BatchMatch developed by SystemsAnalytics Inc., which works same as the above method but has the option to choose to include class label information or not. This is important especially when we are doing classification performance with external validation dataset, that does not have class label information. We compared both methods on the correction of batch in these datasets.

Dimensionality Reduction / Feature Selection

The microarray expression data contain thousands of genes; there may be actually a small number of underlying variables that account for most of the variation in the data (West *et al.* 2001). For example, a few linear combinations of genes may explain most of the response variation. So, dimension reduction is a necessary and crucial part of multivariate analysis of high-throughput assay data such as gene expression data. Class prediction problem is a multivariate regression problem where the number of variables (genes) far exceeds the number of samples. This affects the performance of classification algorithm studying to a bottom level and also it is a computationally expensive procedure. One way to achieve dimension reduction is to transform the large number of original variables (genes) to a new set of variables (gene components) or differentially expressed variables (genes), which are uncorrelated and ordered so that the few genes account for most of the variation in the data (Figure 7).

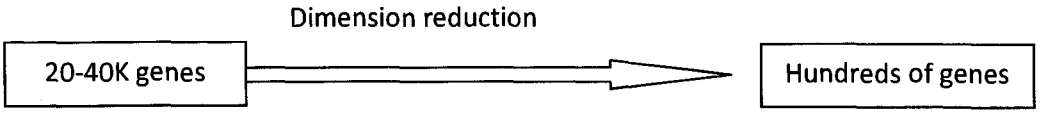


Figure 7: A small schematic depiction of dimensionality reduction of gene expression data.

There are few ways to reduce the dimensionality like Principal Component Analysis (PCA), Sliced Inversion Regression (SIR), Partial Least Squares (PLS) and Fold Change methods (Li *et al.* 2007).

Our approach of dimensionality reduction or feature selection or gene selection is in two levels. In the first level we reduced most of unnecessary genes using fold change and p-value combination. In the second level we applied feature selection algorithms.

We combined both fold change and p-value (<0.05) to filter out differentially expressed genes. We varied the fold change level from 1.2 to 2 based on the dataset, but we fixed the p-value to 0.05. We separated these differentially expressed genes using volcano plot, which plots fold change on x-axis and p-value on y-axis, both in log scales. So, we can easily separate the genes which fall between certain fold change and p-value combination.

After filtering these differentially expressed genes for all 13 end points in six datasets using the corresponding class labels, we checked the box plot distribution for those.

Feature Selection

After filtering differentially expressed genes for each end point, we further reduced the number of genes or features using information gain, gain ratio, χ^2 – statistic, relief-F and Correlation based feature selection algorithms. These feature selection algorithms also ranks genes based on the ranker search algorithm. We discussed about feature selection algorithms earlier in this dissertation in feature selections chapter.

We initially compared and contrasted the above feature selection algorithms to find which algorithm performs better to use further in our studies. We ran these with 10-fold cross validation with 10 iterations with classification algorithms to know which feature selection algorithms generate better classification performance. After this, we separated the subsets of genes (like 10, 20, 30, 40, 60, 100 genes) based on the ranks in each algorithm for further classification studies.

Classification algorithms

I explained about the details of the classification algorithms we studied in this project in chapter 4 of this dissertation. We used Sequential Minimal Optimization (or SMO) with linear kernel by keeping exponential value (E or *d*) to 1 in polynomial kernel of SMO, explained in page number 23. The remaining parameters were kept as default as it is in Weka program. In LibSVM, we studied both linear kernel and Radial Basis Function (RBF) kernel with c =10 and $\gamma = 0.01$ and the type of SVM is C-SVC. We also studied the voted perceptron and Naïve Bayes classification algorithms with default options in Weka machine learning framework. The error estimation is performed using internal cross validation (CV) with 5-fold CV with 10 iterations. Initially we did 10-fold CV with 10 iterations on all datasets, which is a standard and well established. But after

heated discussion in the MAQC 8th face-to-face meeting in Washington, DC., analysts feared to use 10-fold CV on the Hamner dataset, which is small in sample size and weak data with strong batch effect, that can introduce the over-fitting in the classification performance. So, analysis groups decided to use 5-fold to all datasets instead of 10-fold CV. Before this analysis, we tested the implementation of nested cross validation using different SVM classification algorithms to avoid introducing over-fitting. After that, work flow is designed to fit only stratified cross validation only, due to the difficulty in using other classification algorithms in nested cross validation and also stratified CV is recommended by the RBWG.

Error Estimation

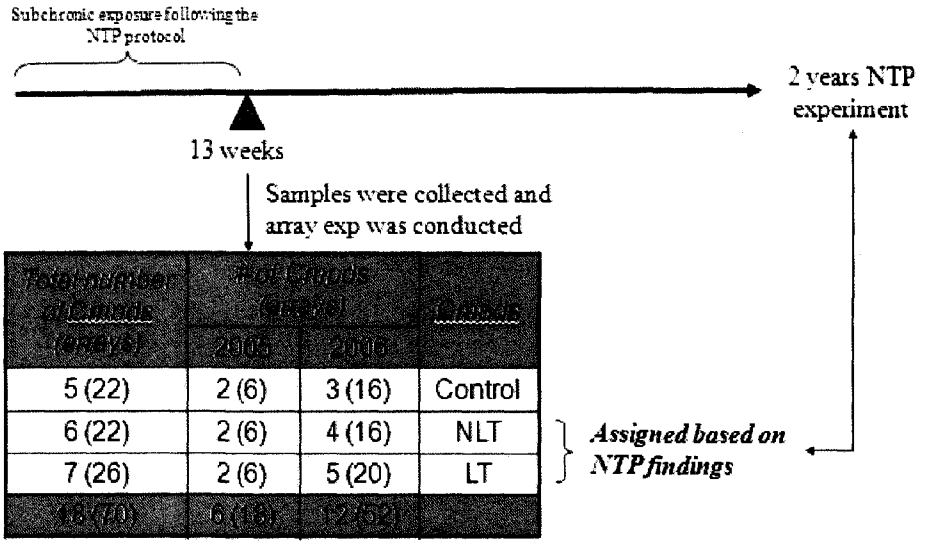
After running these classification algorithms, we tested the internal cross validation error estimation using several classification performance metrics as described in page number 18 of this dissertation. We reported Matthews Correlation Coefficient (MCC), accuracy, sensitivity, specificity, area under ROC curve (AUC) and root mean square error (RMSE) performance metric with the standard deviations from the 10 iterated models for each classification using the confusing matrix generated by classification algorithms. Among these we preferred MCC over other performance metrics in selecting candidate models for each end point due to heavy imbalanced class label datasets we studied. MCC overcomes the bias generated by the imbalanced class label datasets by taking of all four elements of the two class confusion matrix into consideration.

Materials

Datasets used in this project

We already mentioned very briefly about the six datasets available to study in this project in chapter 2 of this dissertation. Here we will go through in detail about the datasets and its experimental designs and the clinical endpoints studied in these experiments to get a better understanding about the endpoints we are predicting.

Hamner Lung Tumor dataset



5 (22)	2 (6)	3 (16)	Control
6 (22)	2 (6)	4 (16)	NLT
7 (26)	2 (6)	5 (20)	LT

Figure 8: The experimental design of the Hamner lung tumor dataset from mice

This is one of the dataset among the three toxicogenomics datasets studied and provided by the Hamner Institute (Thomas *et al.* 2007). The objective of this experiment is whether the gene expression at 13 weeks can predict tumor observed at 2 yrs in mice, which are exposed to toxic compounds. If the classifier developed from this study is valid, then the above hypothesis is true and saves millions of money and time spent for National Toxicology Program (NTP), which exposes the animals to toxic compounds over 2-3 years.

The endpoint we are studying in this experiment is lung tumor formation in two year rodent cancer bioassay. For this experiment total 150 female B6C3F1 mice were used by exposing to 13 chemicals over a period of 90 days (13 weeks) by Rusty Thomas group in the Hamner Institute, NC. Among these chemicals seven were positive for an increased incidence of primary alveolar/bronchiolar adenomas or carcinomas and six chemicals were negative. Animal treatment was initiated at 5 weeks of age. Mice were housed 5 per cage in the same environmental and physical conditions. Animal exposures for each chemical were performed via the route and dose recommended by NTP. Refer to Thomas *et al.* 2007for more information about this experimental design.

Microarray analysis was performed on 3 to 4 animals per treatment group except for the corn oil and feed control groups. The microarray platform used for this study is Affymetrix mouse 430 version 2. They generated total 70 samples of expression data from 22 non-carcinogen (NLT) exposure, 26 carcinogen exposure (LT) and remaining 22 were treated with controls (Figure 8). These 70 samples were generated in two experimental batches (18 in year 2005 and 52 in year 2006). For our analysis purpose, we labeled the phenotypic classes as Lung Tumor (LT) and Non-Lung Tumor (NLT) and our endpoint code for this class is 'A' with positives to negatives ratio 0.59. We treated the control samples also into NLT class as per MAQC recommendation.

Iconix dataset

This is another toxicogenomics dataset provided by the Iconix Inc. The experimental study is explained in their publication in (Fielden *et al.* 2007). They studied the hepato-carcinogenicity (liver cancer) in rats by exposing them to several chemical

compounds. They were exposed to 22 chemical compounds with 2-3 doses per compound and 4-5 time points. They treated 3 rats per dose-time combination. The exposure is multiple routes depending on the compound. Totally they provided 216 samples of expression data on single colored Codelink RU1 platform from GE Healthcare Inc. In this, 73 samples are phenotypically classified as liver carcinogenous and 143 samples are non-liver carcinogenous and the class is labeled as Class (B) with a positives to negative ratio 0.51. The endpoint we are studying is Liver Carcinogen.

NIEHS dataset

This is third toxicogenomics dataset in the MAQC datasets provided by the National Institute of Environmental Health Sciences (NIEHS), NIH (Lobenhofer *et al.* 2006). The experiment is designed by exposing rats with seven acute hepato-toxicants and one non-toxic control. The experimental design has four doses for each compound and three time points for each compound-dose group and four rats for each dose-time-compound group with a total of 214 samples. The class is labeled as Class (C) and the end point we are studying is the Overall Necrosis Score with positives to negatives ratio of 0.58 (79/135).

MDACC-BR dataset

This breast cancer (BR) dataset, part of the clinical datasets studied in the MAQC was provided by MD Anderson Cancer Center (MDACC) (Hess *et al.* 2006). They studied two clinical end points based on the treatment outcome in this experiment, one is pathologic complete response (pCR) and the other one is estrogen receptor status (erpos). pCR was defined as no residual invasive cancer in the breast or lymph nodes. Residual *in*

situ carcinoma without invasive component was also considered a pCR (Hess *et al.* 2006). The gene expression data was provided using Human Affymetrix U133A slides from total 130 samples. In this the pCR end point has 0.34 (33/97) positives to negatives ratio and erpos end point has 1.6 (80/50) positives to negatives ratio.

Multiple Myeloma (MM)

This multiple myeloma (MM) dataset, studied four clinical end points, provided by the University of Arkansas Medical Sciences (UAMS) (Shaughnessy *et al.* 2007a; Shaughnessy *et al.* 2007b). The UAMS provided the 340 samples of gene expression data in Affymetrix U133 Plus version 2 platform for the analysis. They studied four clinical end points or classes, namely Overall Survival Milestone Outcome (OS_MO), Event-free Survival Milestone Outcome (EFS_MO), Clinical Parameter S1 (CPS1) and Clinical Parameter R1 (CPR1).

- OS milestone outcome (OS_MO) is a coding of a binary clinical outcome (overall survival) related to whether the subject survived up to the milestone (24 months): 1= deceased by 24 months, 0= alive at 24 months. The positive to negatives ratio of this end point is 0.48 (112/228).

- EFS milestone outcome (EFS_MO) is a coding of a binary clinical outcome (event-free means disease relapse or progression) related to whether the subject was event-free up to the milestone (24 months): 1=event occurred < 24 months, 0=no event in first 24 months. The positives to negatives ratio of this end point is 1.1 (179/161).

- Clinical parameter S1 is a coding of binary clinical outcome of parameter S1 is either positive (1) or negative (0). The data providers did not provide the complete description of this parameter based on their confidentiality policy. The positives to negatives ratio of this end point is 1.33 (194/146).

- Clinical parameter R1 is a coding of binary clinical outcome of parameter R1 is either positive (1) or negative (0). The positives to negatives ratio of this end point is 1.43 (200/140).

Neuroblastoma (NB) dataset

This neuroblastoma (NB) clinical dataset is provided by the University of Cologne, Germany with four clinical end points to study (Oberthuer *et al.* 2006). The gene expression data they provided is in customized Agilent NB array, which is two color data. They provided total 492 expression profiles from 246 NB samples along with dye-flipped replicates. The four clinical outcomes in this study are Overall Survival Outcome (OS_MO), Event-free Survival Outcome (EFS_MO), Newly Established clinical outcome Parameter S (NEP_S) and Newly Established clinical outcome Parameter R (NEP_R).

- OS milestone outcome (OS_MO) is a binary coding of overall survival status by the milestone (900 days): 0= alive, 1=deceased. The positives to negatives ratio in this clinical end point is 0.32 (59/187).

- EFS milestone outcome (EFS_MO) is a coding of binary clinical outcome, event-free (event-free means disease relapse or progression) survival status with consideration of the nature of the event by the milestone (900 days): 0=no event

by milestone, 1=event by milestone. The positives to negatives ratio in this end point is 0.65 (97/145).

- NEP_S is the newly established clinical outcome parameter S. The positives to negatives ratio of this end point is 1.44 (145/101).

- NEP_R is the newly established clinical outcome parameter R. The positives to negatives ratio of this end point is 1.44 (145/101).

The summarized details about the datasets are given in the Appendix A.

Chapter summary

In this chapter, I explained about the methods and materials used in this analysis. I explained in detail about the preprocessing steps and quality control measures taken on the datasets studied. Also explained about the design of the analysis, the feature selection algorithms, classification methods studied and the error estimation methods used. This chapter also covered about the datasets studied and the end points (clinical and pre-clinical outcomes) used in the classification prediction studies.

CHAPTER III

RESULTS AND DISCUSSION

The results of our analysis from preprocessing to classification prediction and performance on total 13 clinical and preclinical end points of six different datasets studied in this project are explained and discussed in this chapter.

Outlier identification

As part of the quality assessment of the arrays of the datasets being studied, we did dChip analysis, box plot distribution and PCA (where ever necessary) to identify array outliers as described in the methods section of this dissertation.

For the Hamner dataset, we got the "array summary file" (Table 2) after "Model-based expression" a pre-processing step from the dChip, and looked for unusual median intensity, low P call % and higher array outlier %.

For each array, array outliers measure the percent of probe sets with expression patterns that are inconsistent from the rest of the arrays. In general, dChip gives warnings by showing ' * ' in the warnings column of the result, when either single or array outlier percentages exceed 5% and recommends removal of arrays from further analysis when any of these values is 15% or more. None of the chips were flagged based on these criteria (Table 2). Finally, I observed the box plot distribution (Figure 9) of these arrays using GeneSpring software by selecting RMA pre-processing.

By carefully observing the two results, we voted GSM142182 file as single outlier in these array files based on its unusual 'Median Intensity' value and low 'P call %' and

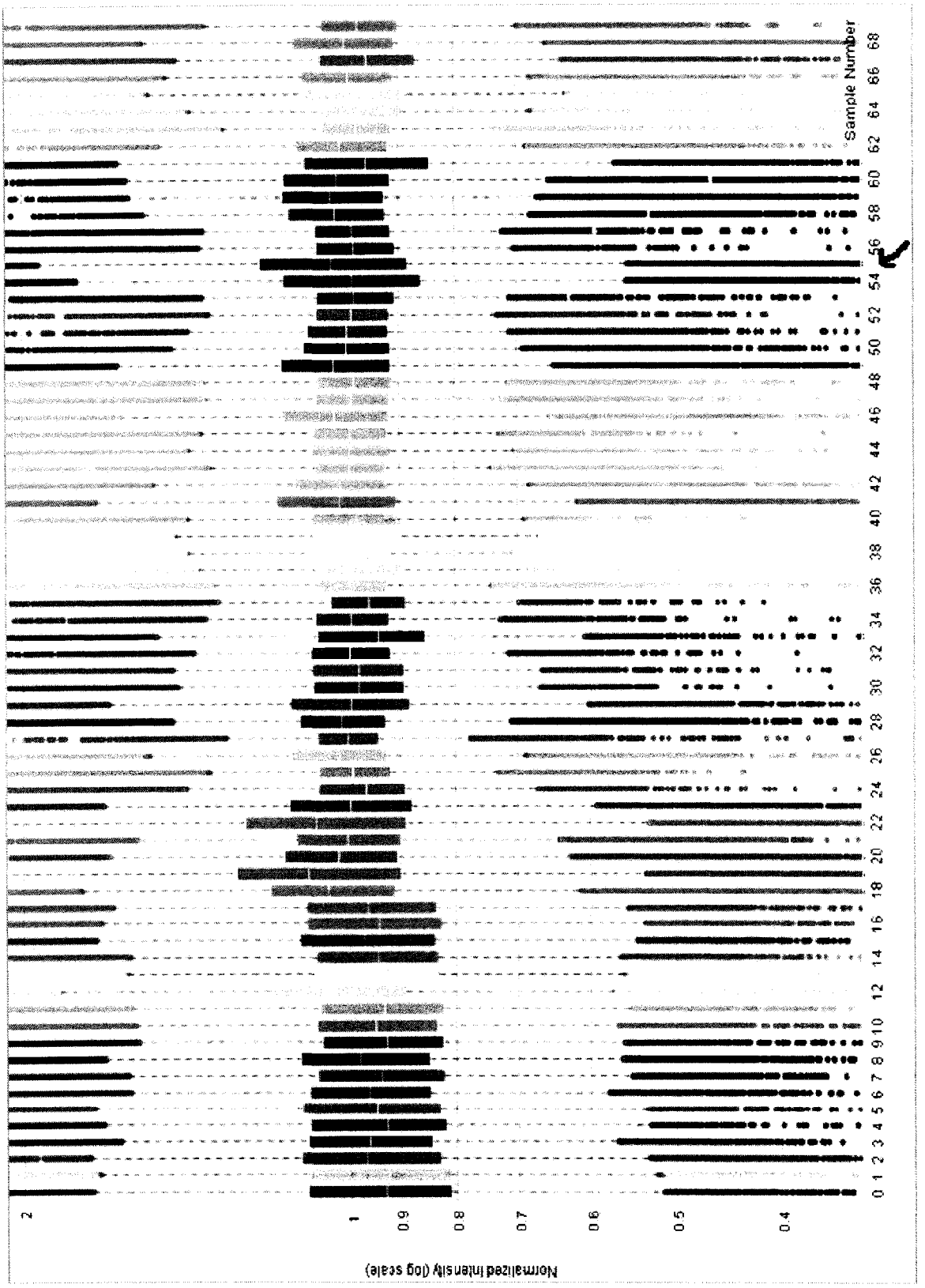
high '%array outlier'. Also we voted nine other array files as marginal outliers based on the above results and rest as non-outliers reported in the voting sheet of MAQC. This information is used to do further meta-analysis on the outlier identification methods.

Number	Array	Median Intensity	P call %	% Array outlier	% Single outlier	Warnings
55	GSM142182	215	55.1	1.049	0.397	
7	GSM142134	137	67.3	0.953	0.37	
68	GSM142195	106	64.7	0.936	0.216	
30	GSM142157	209	56.2	0.933	0.568	
62	GSM142189	172	59.8	0.92	0.307	
13	GSM142140	92	65	0.869	0.284	
69	GSM142196	98	62	0.778	0.304	
16	GSM142143	106	66	0.772	0.265	
4	GSM142131	103	66.6	0.756	0.276	
2	GSM142129	86	69.1	0.752	0.225	
32	GSM142159	60	66.1	0.698	0.27	
56	GSM142183	116	55.7	0.696	0.381	
11	GSM142138	152	68.1	0.694	0.25	
5	GSM142132	101	69.6	0.645	0.228	
3	GSM142130	128	66.6	0.63	0.23	
9	GSM142136	134	65.9	0.619	0.214	
17	GSM142144	82	67.9	0.55	0.238	
24	GSM142151	242	55.4	0.508	0.365	
1	GSM142128	107	69.4	0.506	0.188	
6	GSM142133	95	68.6	0.499	0.206	
20	GSM142147	101	59.8	0.468	0.301	
12	GSM142139	134	68.7	0.421	0.175	
10	GSM142137	136	69.5	0.408	0.167	
57	GSM142184	92	66.6	0.392	0.267	
14	GSM142141	132	68.7	0.386	0.206	
27	GSM142154	91	63.1	0.386	0.246	
18	GSM142145	136	67	0.357	0.164	
31	GSM142158	93	67.2	0.344	0.178	
51	GSM142178	104	65.4	0.344	0.151	
61	GSM142188	95	62	0.326	0.282	
22	GSM142149	111	58.7	0.322	0.309	
53	GSM142180	80	65	0.306	0.235	
23	GSM142150	150	59.2	0.295	0.199	
15	GSM142142	127	68.5	0.29	0.182	
66	GSM142193	118	62	0.29	0.193	
50	GSM142177	94	64.1	0.279	0.158	
21	GSM142148	94	63.5	0.275	0.222	
40	GSM142167	88	67.5	0.275	0.153	

59	GSM142186	122	62.5	0.255	0.242	
67	GSM142194	127	61.1	0.251	0.211	
47	GSM142174	111	61.8	0.248	0.191	
42	GSM142169	86	65.4	0.239	0.165	
37	GSM142164	115	65.9	0.228	0.185	
48	GSM142175	87	66.3	0.228	0.148	
34	GSM142161	83	68.4	0.224	0.148	
54	GSM142181	82	65	0.222	0.18	
25	GSM142152	101	65.7	0.215	0.158	
19	GSM142146	99	62.8	0.206	0.182	
26	GSM142153	98	66.4	0.204	0.134	
58	GSM142185	111	67	0.2	0.124	
39	GSM142166	98	65.9	0.195	0.127	
70	GSM142197	118	65.8	0.193	0.14	
60	GSM142187	104	63.4	0.191	0.196	
49	GSM142176	101	67.4	0.186	0.121	
63	GSM142190	107	61.7	0.184	0.178	
41	GSM142168	86	67.7	0.171	0.13	
8	GSM142135	135	67.7	0.166	0.104	
65	GSM142192	157	62.6	0.166	0.141	
33	GSM142160	83	67.2	0.162	0.1	
52	GSM142179	98	65.3	0.157	0.174	
45	GSM142172	162	66	0.155	0.141	
28	GSM142155	108	65.4	0.14	0.128	
64	GSM142191	114	66	0.133	0.143	
43	GSM142170	132	66.6	0.126	0.109	
46	GSM142173	146	67	0.118	0.126	
29	GSM142156	89	65.8	0.109	0.107	
36	GSM142163	152	68.8	0.098	0.088	
35	GSM142162	120	67.3	0.089	0.126	
38	GSM142165	132	68.5	0.089	0.115	
44	GSM142171	146	68.3	0.084	0.12	

Table 2: dChip analysis results of the Hamner dataset which contains 70 array samples. The analysis results showed no array is an outlier in the total 70 samples (observe no warnings in the warning column in this table). But we voted sample array GSM142182 as an array outlier (shown in orange background above) and marginal outliers (yellow color) based on the box plot distribution.

Figure 9: The figure shown in the next page is the box plot distribution of RMA normalized 70 array samples of Hamner dataset to identify array outliers. The red mark shown in this figure is voted a array outlier based on dChip analysis and this box plot distribution.


Normalized Intensity (log scale)
2
1
0.9
0.8
0.7
0.6
0.5
0.4
Sample Number
0 1 2 3 4 5 6 7 8 9 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68

The outlier identification analysis results for MDACC breast cancer dataset using dChip analysis (Table 3) showing significant array outliers. By carefully observing the two types of analysis (Figure 10) results as I mentioned above, we voted as array outlier for five files in these 176 array files based on its unusual 'Median Intensity' value and low 'P call %' and high '%array outlier' (more than 15%). Also we voted 38 other array files as marginal outliers and rest as non-outliers.

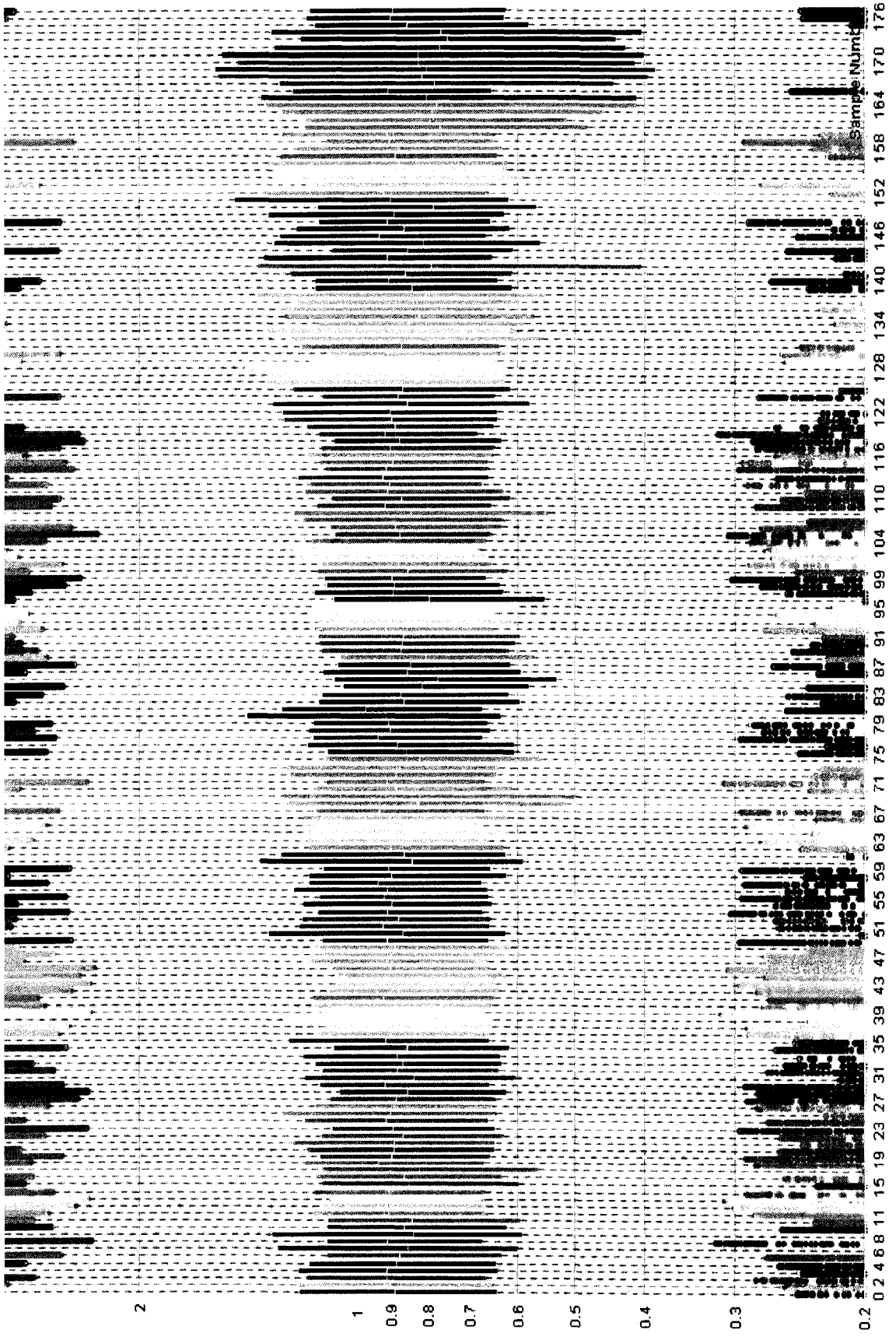
Number	Array	Median Intensity	% P call	% Array outlier	% Single outlier	Warning
143	29539_AB01833733_35649	62	23.3	29.3	2.378	*
168	U133A_FL151_US129_12_08_05	60	27.3	16.654	1.647	*
109	28998_AB02091099_34966	93	32.9	16.08	1.254	*
69	23678_AB01562100_24635	48	41.7	13.45	1.023	*
166	U133A_FL136_US123_11_14_05	211	9.1	12.22	2.353	*
171	U133A_FL175_US147_01_13_06_2	149	19.5	12.05	0.896	*
70	23678_AB01562100_26133	48	37.3	11.619	0.957	*
135	29539_AB01833522_35706	80	43.8	11.152	1.26	*
165	U133A_FL112_US120_10_13_05	113	26.8	10.506	1.066	*
63	23678_AB01542220_24643	76	58.5	9.509	0.551	*
145	29539_AB01833747_35697	64	52.8	9.492	1.049	*
172	U133A_FL32-US2_05_19_05	105	19.9	9.447	0.876	*
67	23678_AB01542241_24647	65	56.2	9.204	0.511	*
173	U133A_FL46-314_07_08_05	90	29.8	9.034	0.945	*
77	23678_AB01562152_24646	75	46.2	9.016	0.557	*
139	29539_AB01833699_35605	97	64.7	8.823	0.25	*
170	U133A_FL161_US125_01_10_06	90	22.1	8.729	0.832	*
71	23678_AB01562113_24644	66	55.2	8.657	0.494	*
175	U133A_FL80_US97_09_01_05	95	27.4	8.616	0.824	*
133	29539_AB01833504_35681	107	65	8.594	0.35	*
162	FL398-PERU53	92	37.2	8.45	0.879	*
136	29539_AB01833526_35614	90	63.7	8.392	0.359	*
126	29539_AB01723039_35684	79	57.5	8.257	0.534	*
161	29539_AB01833935_35648	86	57.2	7.759	0.443	*
65	23678_AB01542230_24645	73	55.7	7.746	0.447	*
151	29539_AB01833769_35700	89	63.6	7.629	0.269	*
146	29539_AB01833749_35607	100	62.9	7.625	0.254	*
169	U133A_FL15_03_17_05	101	19.3	7.616	0.69	*
160	29539_AB01833931_35690	82	53.2	7.387	0.721	*
75	23678_AB01562130_24648	65	47.6	7.337	0.571	*
130	29539_AB01723044_35687	97	55.9	7.311	0.582	*
127	29539_AB01723040_35686	92	58.8	7.198	0.391	*

163	FL412-PERU55	95	39.9	7.064	0.771	*
174	U133A_FL78_US92_09_01_05	84	31.1	6.907	0.612	*
59	23678_AB01542151_24650	107	56.6	6.844	0.33	*
140	29539_AB01833716_35658	85	60.4	6.839	0.423	*
141	29539_AB01833728_35659	96	49.8	6.83	0.721	*
142	29539_AB01833732_35677	96	61.1	6.444	0.323	*
131	29539_AB01723056_35693	90	59.5	6.202	0.488	*
154	29539_AB01833821_35682	85	56.4	6.166	0.483	*
49	23678_AB01233000_24649	69	56.1	6.067	0.361	*
132	29539_AB01833495_35688	96	59.2	6	0.42	*
128	29539_AB01723041_35689	97	61.6	5.951	0.372	*
148	29539_AB01833756_35615	88	62.3	5.91	0.277	*
121	29539_AB01723009_35679	77	60.1	5.803	0.495	*
157	29539_AB01833840_35610	82	61.8	5.641	0.27	*
159	29539_AB01833876_35613	91	59.1	5.556	0.404	*
164	FL454-713	83	35.1	5.533	0.545	*
137	29539_AB01833535_35695	115	58.1	5.475	0.343	*
155	29539_AB01833829_35611	97	64.5	5.466	0.202	*
125	29539_AB01723032_35694	91	57.4	5.439	0.501	*
11	19893_AB01923090_16992	133	48.6	5.412	0.963	*
91	24817_AB02262650_26174	62	56.6	5.313	0.364	*
134	29539_AB01833515_35616	104	61.3	5.287	0.216	*
149	29539_AB01833758_35698	84	55	5.264	0.488	*
98	24817_AB02263405_26175	68	57	5.17	0.401	*
122	29539_AB01723028_35692	94	59	5.026	0.384	*
176	U133A_ROM233_06_04_04	60	49.2	4.999	0.428	
144	29539_AB01833741_35650	88	58.4	4.887	0.315	
10	19893_AB01913300_16991	146	49	4.802	1.028	
85	24817_AB02261485_26161	66	57.7	4.784	0.257	
129	29539_AB01723043_35685	110	60.9	4.748	0.248	
16	19893_AB01983478_17035	58	47	4.604	0.488	
152	29539_AB01833780_35612	99	60.1	4.582	0.206	
158	29539_AB01833841_35702	92	59.6	4.555	0.242	
156	29539_AB01833832_35608	91	62	4.542	0.235	
138	29539_AB01833542_35683	93	63.8	4.506	0.285	
123	29539_AB01723030_35657	86	60.6	4.344	0.24	
167	U133A_FL137_US134_11_14_05	1001	36.8	4.313	0.537	
150	29539_AB01833759_35699	89	61	4.061	0.214	
124	29539_AB01723031_35678	98	64.1	3.99	0.204	
177	U133A_ROM286_06_04_04	61	53.9	3.904	0.309	
96	24817_AB02263399_26158	76	49.2	3.801	0.526	
147	29539_AB01833754_35654	156	53.6	3.801	0.238	

Table 3: dChip analysis results of the MDACC breast cancer dataset which contains 178 array samples. The results shows (only 75 arrays results showed due to space constraint) 57 arrays as an array outliers in the total 178 samples (observe for '*' in warnings

column). But we voted 5 arrays as an array outlier (shown in orange background above) and other 39 as marginal outliers based on the box plot distribution and dChip results.

Figure 10: The figure shown in the next page is the box plot distribution of RMA normalised values for 178 array samples from Iconix. We can observe the most of outliers and moderate outliers are in between array numbers 140 to 175.


Normalized Intensity (log scale)
2
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0 2 4 6 8 11 15 19 23 27 31 35 39 43 47 51 55 59 63 67 71 75 79 83 87 91 95 99 104 110 116 122 128 134 140 146 152 158 164 170 176
Sample Numb

The meta-analysis of the outlier identification methods and its results (Figure 11) from the other analysis groups (Appendix B&C) provided by Leming Shi (MAQC coordinator) are discussed below. The criteria used for calling the final outlier quality based on the consensus score for each array. Consensus score is calculated from,

Consensus score (%) = 100*(Sum of the votes from *n* organizations)/*n*

MAQC proposed the following rules for assigning an array quality as one of the three statuses:

Status	*Consensus score (%)*
Good:	<33.33% (<1/3 votes)
Marginal:	>=33.33% and <66.67% (1/3 - 2/3 votes)
Outlier:	>=66.67% (>=2/3 votes)

Based the analysis and criteria explained above, MAQC identified a consensus final array outliers for all the six datasets used in this project based on the meta-analysis. There are no consensus array outliers for Hamner, Iconix, NIEHS datasets. Consensus array outliers are found in MDACC breast cancer dataset (19 arrays), MM dataset (5 arrays) and NB dataset (5 arrays). I shown the array names (Table 4) which are excluded for the further analysis, but did not show the 38 arrays excluded from MDACC breast cancer dataset.

Multiple Myeloma dataset Consensus array outliers (5)	**Neuroblastoma dataset Consensus array outliers (5)**
P0266-01-B79-U133Plus-2.CEL	US22502540_251271410122_S01_A02.txt
P0748-01-C393-U133Plus-2.CEL	US22502540_251271410124_S01_A02.txt
P0753-01-C413-U133Plus-2.CEL	US22502540_251271410332_S01_A02.txt
P0941-02-C782-U133Plus-2.CEL	US22502540_251271410531_S01_A02.txt
P0984-01-C763-U133Plus-2.CEL	US22502540_251271410646_S01_A02.txt

Table 4: Consensus array outliers which are excluded from the further analysis for MM and NB datasets are shown in the above table (not shown the 38 arrays from MDACC-BR dataset)

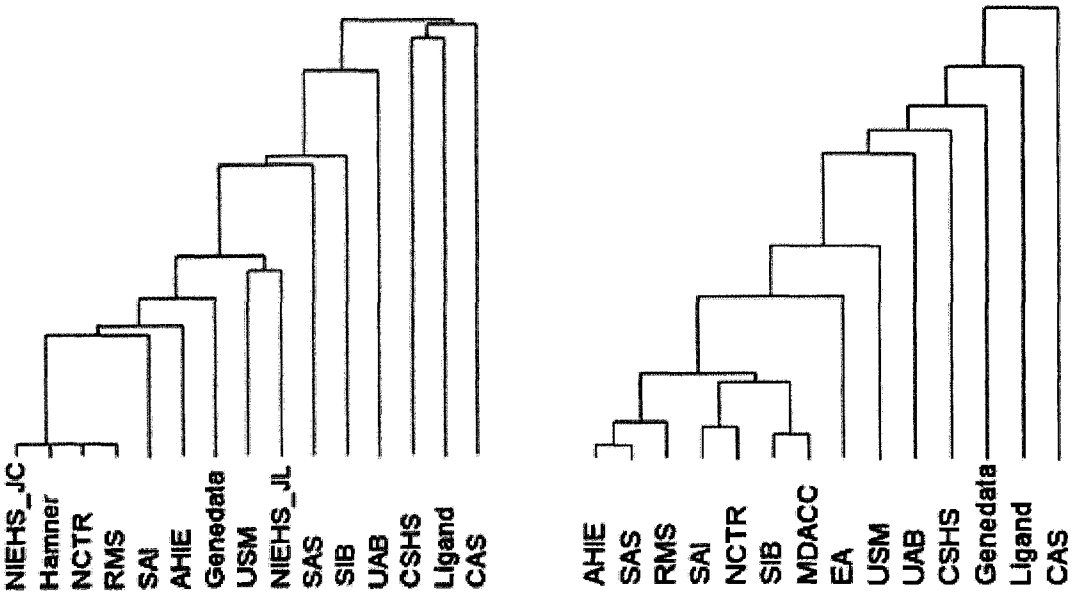


Figure 11: Summarized view of the array outlier voting from different analysis groups shown in cluster diagram. The left side cluster is for the Hamner dataset and the right side cluster is for MDACC breast cancer dataset (data of the matrix shown in the Appendix B&C).

Preprocessing and Normalization

We performed the basic preprocessing low level summarization methods for Affymetrix datasets (Hamner, NIEHS, MDACC-BR and MM). The normalization we performed on these datasets is MAS5 (with a target value 500) to facilitate the incoming external validation datasets without any changes to the model developed with training datasets. For Iconix dataset, we performed median scale 1000 normalization method and mean scale normalization for NB dataset. For NB agilent dataset background correction was applied by FG - meanBG calculation.

In the next step, we filtered the genes based on the P, M and A absolute flag values generated from the MAS5. Also filtered the genes which has low signal values, the threshold cut off used based on the dataset we are studying.

After $\log_2$ transformation of signal values, we used KNN- based missing value imputation algorithm to predict the missing values. Finally, we performed the quality of the arrays with box plot distribution after normalization and compared with before normalization. But here, I did not show the distribution results because of the large number of samples in the datasets, the quality and appearance of the images are not fine for dissertation purpose.

Batch Effect and Correction

Batch effect identification and correction is an important step in the quality assessment procedure, especially when we are working with large number of samples in the datasets. We performed this assessment using correlation heat maps and principal component analysis (PCA) and Q-Q plot visualizations of the datasets. We observed strong and significant batch effect based on the year of samples generated in Hamner and Iconix datasets. There is only slight and insignificant batch effect in the other four datasets studied.

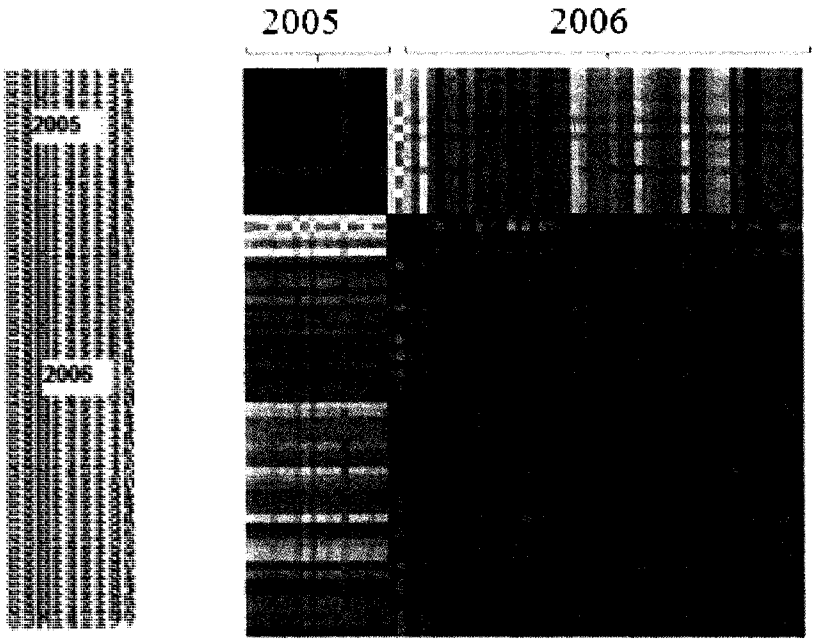


Figure 12: Correlation heat map of the Hamner lung tumor dataset with 70 arrays. This heat map clearly shows the strong batch effect based on the year of array production.

We corrected the batch effect using an R function called 'Combat' in the bioconductor package. This adjusts the data based on parametric and non-parametric empirical Bayes frame work. The complete details about this method are obtained from Johnson *et al.* 2007 paper. We generated the correlation heat maps before (Figure 12) and after correcting the batch effect (Figure 13). Also we generated principal component analysis (PCA) diagrams before (Figure 14a) and after (Figure 14b) the batch correction. These two visualizations clearly show the correction of batch effect present in this dataset using combat algorithm. This algorithm also generates Q-Q plots (Figure 15) to check the normality of the data.

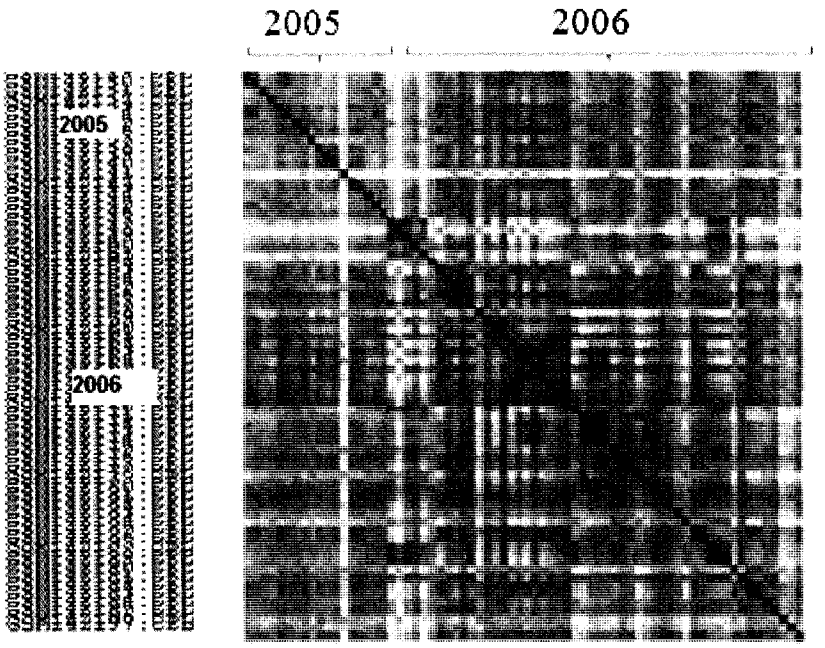


Figure 13: Correlation heat map of the Hamner lung tumor dataset with 70 arrays after batch correction with combat funtion. This heat map clearly shows the correction of the batch effect based on the year of array production.

"Quantile-quantile plots (also called QQ plots) are used to determine if two data sets come from populations with a common distribution. In such a plot, points are formed from the quantiles of the data. If the resulting points lie roughly on a line with slope 1, then the distributions are the same". (http://mathworld.wolfram.com/Quantile-QuantilePlot.html).

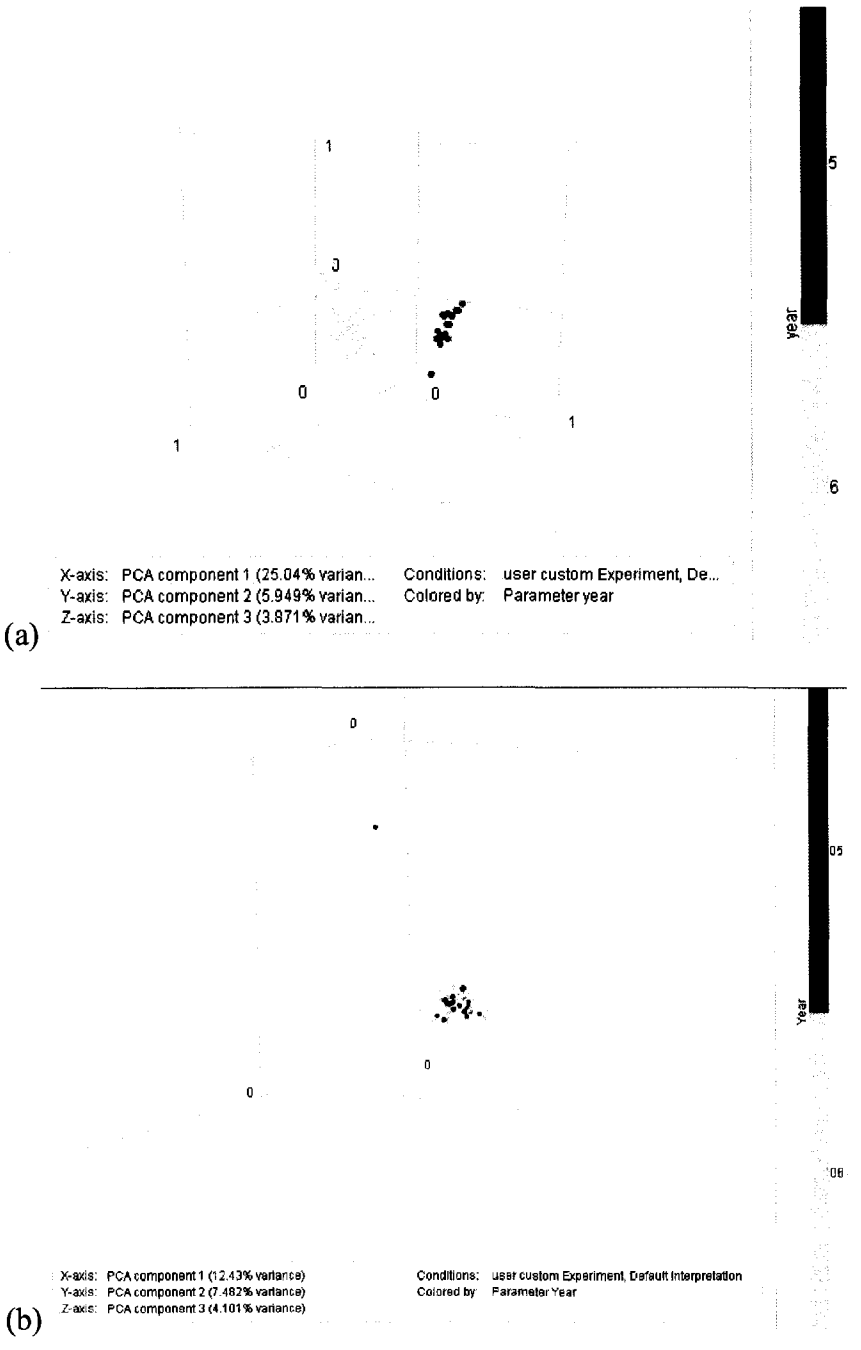


Figure 14: Principal component analysis (PCA) of the Hamner lung tumor dataset with 70 arrays based on the year (samples shown red with year 2005 and yellow with year 2006) (a) before batch correction, samples are separated clearly based on the year (b) we cannot observe the differentiation based on the year after batch correction.

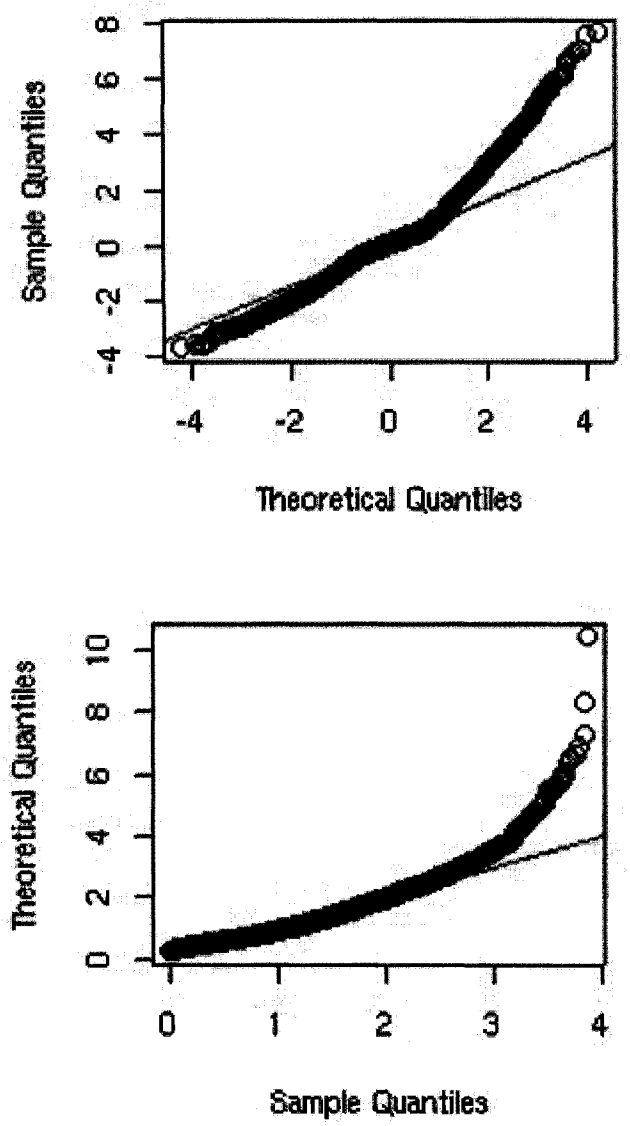


Figure 15: Q-Q plots for the Hamner dataset before correction of batch (top) and after correction (bottom). Q-Q plots shows the normality of the dataset, if the quantiles of theoretical and samples falls straight on the line (empirical bayes normal line) then the dataset is near to normal.

Compare with BatchMatch 1.3

We also compared and contrasted the Batchmatch 1.3 from System Analytics Inc. (www.systemsanalystics.com) on the correction of batch effect in these datasets. Batchmatch works presently on 'double scaling' algorithm. The results generated from this algorithm shows a clear batch effect in both Hamner and Iconix datasets. It generates the visual results in correlation heat maps (Figure 16), PCA, sample cluster diagrams and analysis of variance (ANOVA).

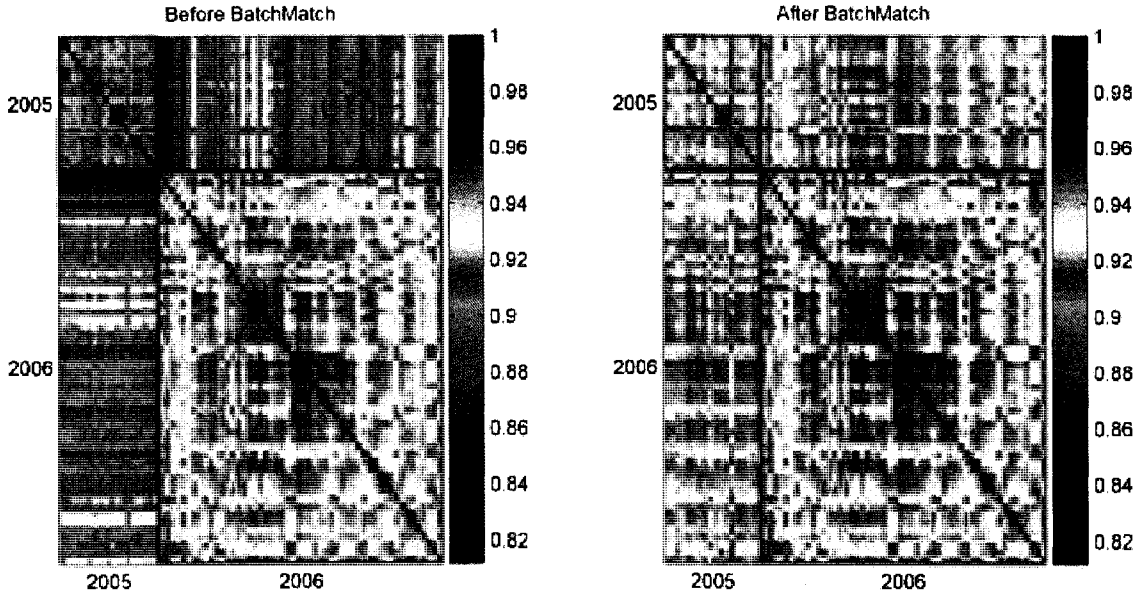


Figure 16: The above correlation heat maps for Hamner lung tumor dataset shows the same results as it was with Combat. The left side image shows before batch correction and right side image shows after batch correction.

In the next page, ANOVA results of Hamner dataset are shown (Figure 17). In Anova analysis, two-factor ANOVA (treatment/biological effect and batch effect) with interaction term is performed before and after batch effect removal. The total variance and the variance percentage of each effect are shown on the pie charts.

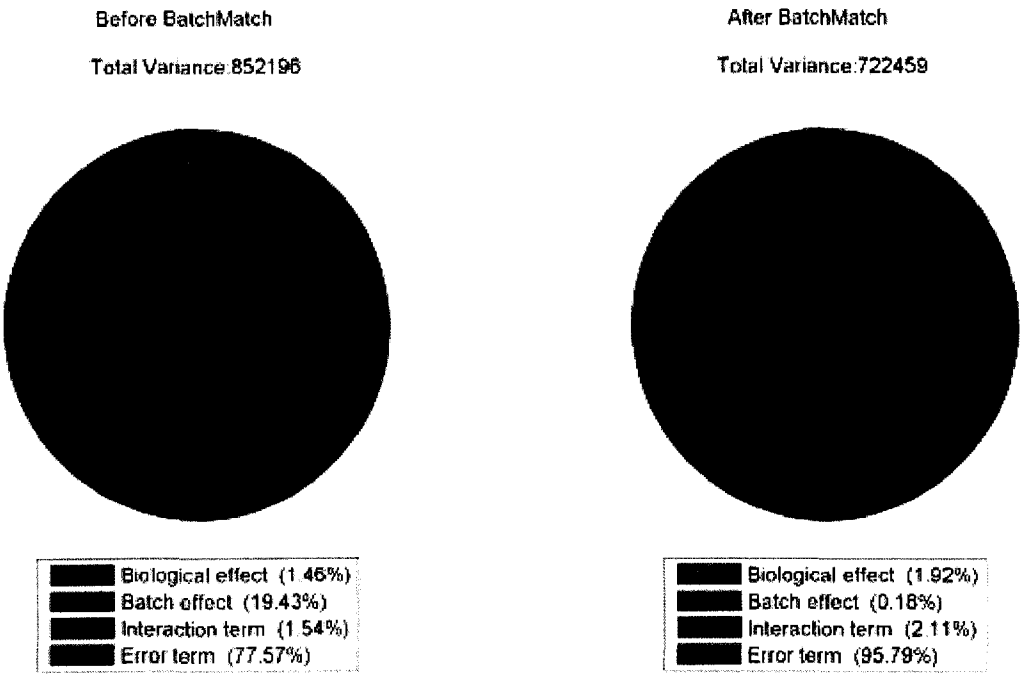


Figure 17: The two-way ANOVA (LT_NLT and Batch label *i.e.* Year) results for Hamner lung tumor dataset shows the adjustment of batch effect. The left side image shows before batch correction and right side image shows after batch correction.

Besides, giving a similar performance in adjusting the batch effect as it was with Combat function, the Batchmatch has several advantages over Combat especially in predicting classifiers using with or without class label information. Other than this, we have the chance of using one, a few, or all batches as reference. Reference batch(s) become necessary when the objective of the study is to construct a predictive model using the current available dataset to predict the labels of future dataset.

We performed the same analysis on Iconix liver cancer dataset, which shows strong batch effect based on the year 2001 and 2002. The correlation heat maps (Figure 18), PCA (Figure 19) and ANOVA (Figure 20) results of this dataset are shown in the next pages.

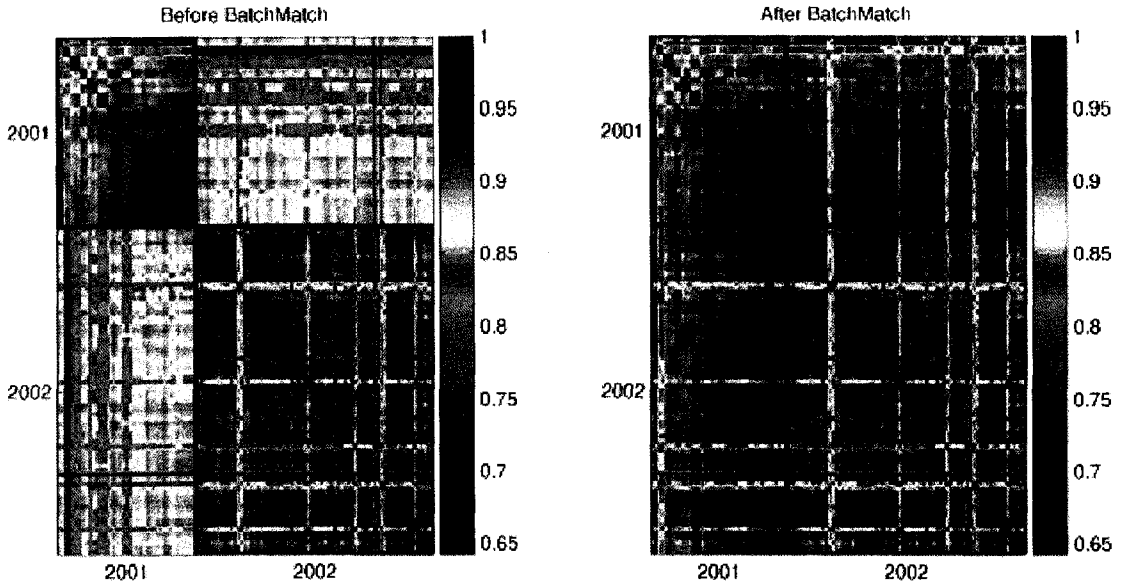


Figure 18: The correlation heat maps for Iconix liver cancer dataset (216 samples) shows the clear batch effect based on the 2001 and 2002 year. The left side image shows before batch correction and right side image shows after batch correction.

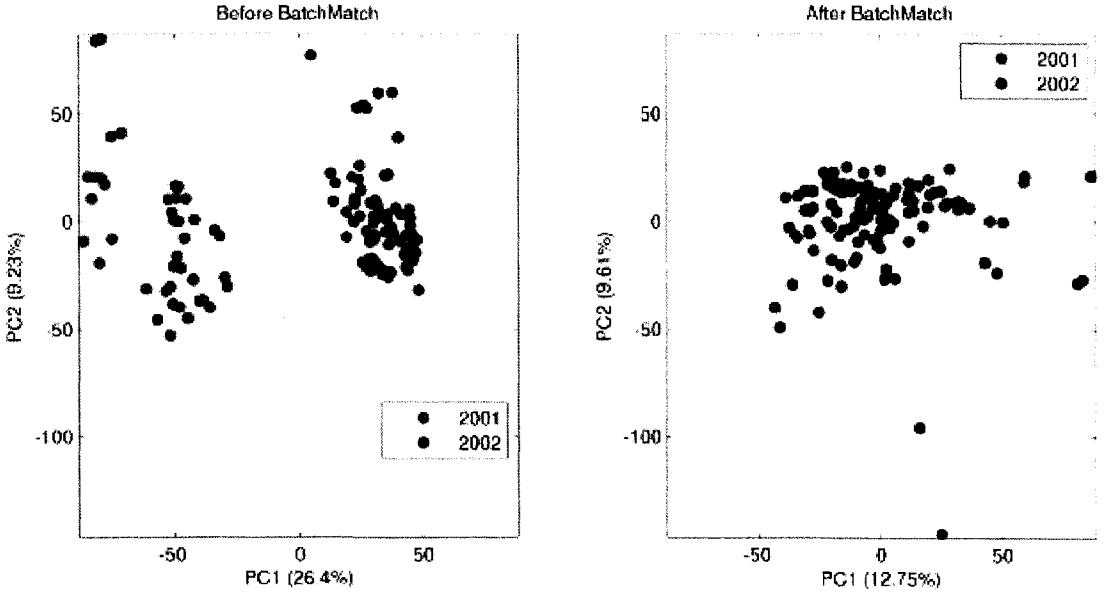


Figure 19: Principal component analysis (PCA) of Iconix liver cancer dataset (216 samples) shows the clear batch effect based on the 2001 and 2002 year. The left side image shows before batch correction and right side image shows after batch correction.

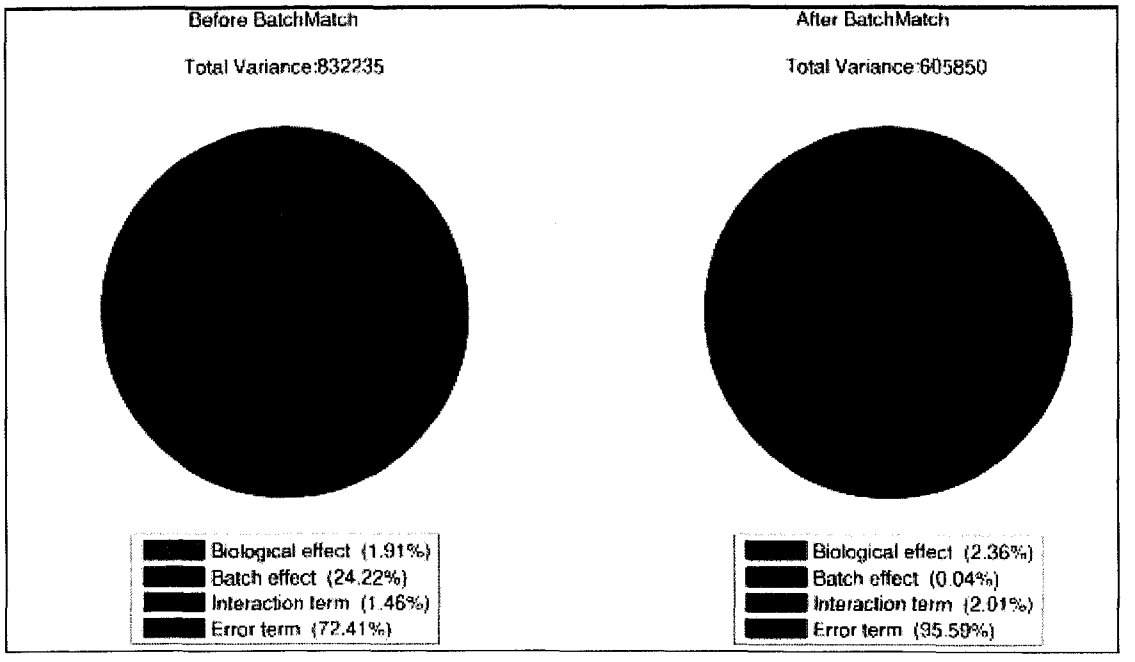
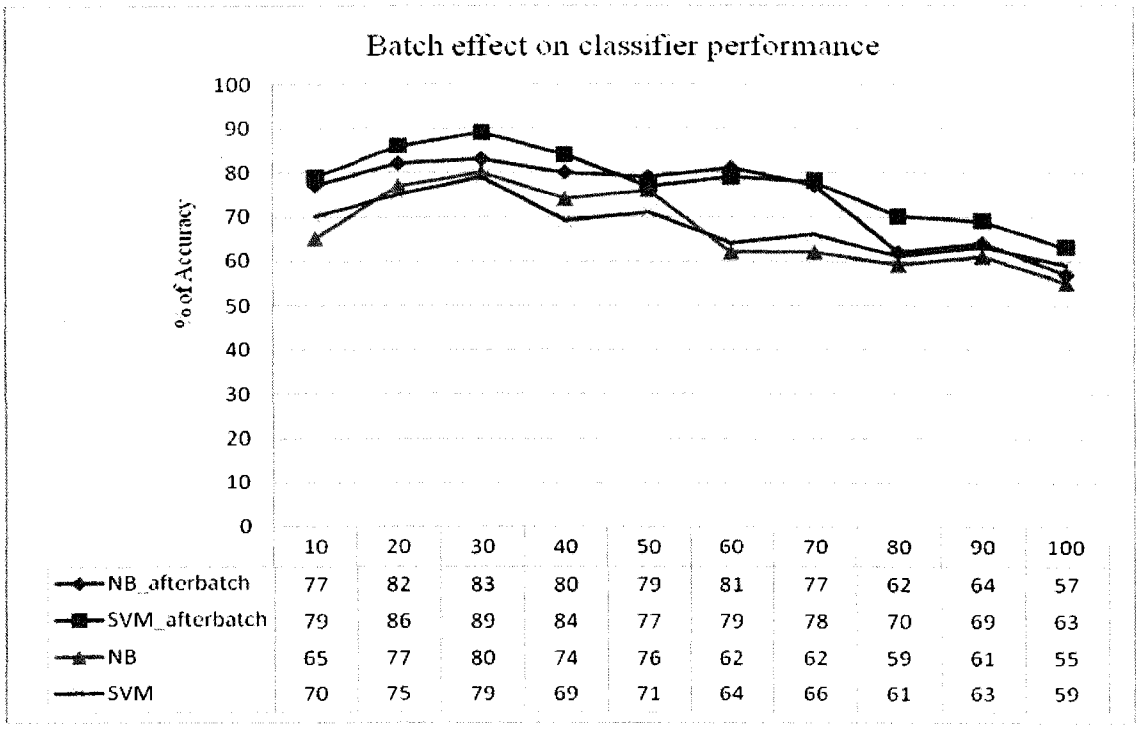


Figure 20: The two-way ANOVA (Class and Batch *i.e.* Year) results for Iconix liver cancer dataset shows the adjustment of batch effect. The left side image shows before batch correction and right side image shows after batch correction.

We also studied the effect of batch in expression data on the performance of class prediction. For this analysis, we used SVM and Naïve Bayes (NB) classification algorithms and the feature selection is based on fold change and p-value and the evaluation by gainratio ranking algorithm on the Hamner lung tumor with and without batch adjustment datasets. The error estimation was done using 10-f cross validation with 10 iterations and percentage of accuracy as performance metric. The features selected in 10, 20, 30, ...100 subsets based on the ranking provided by gainratio algorithm. We observed from these results that the batch effect adjusted dataset performed better than uncorrected dataset (Table 5).



	10	20	30	40	50	60	70	80	90	100
NB_afterbatch	77	82	83	80	79	81	77	62	64	57
SVM_afterbatch	79	86	89	84	77	79	78	70	69	63
NB	65	77	80	74	76	62	62	59	61	55
SVM	70	75	79	69	71	64	66	61	63	59

Table 5: SVM and Naïve Bayes (NB) classification performance based on the % of accuracy on the Hamner with and without batch adjusted datasets. The after batch adjusted data performed better than the without batch corrected dataset.

With the above results, we finished the preprocessing of datasets and quality assessment. We corrected the batch effect observed in the Hamner lung tumor dataset and Iconic liver cancer dataset based on the year they generated the samples. We came to a conclusion that Batchmatch and Combat performs similarly, but Batchmatch has several advantages over Combat in the application of classification without leaking class label information. Also we came to know that batch noise in expression data shows effect on classification performance.

Dimensionality reduction is the next step in our work flow of this analysis after this preprocessing and quality check.

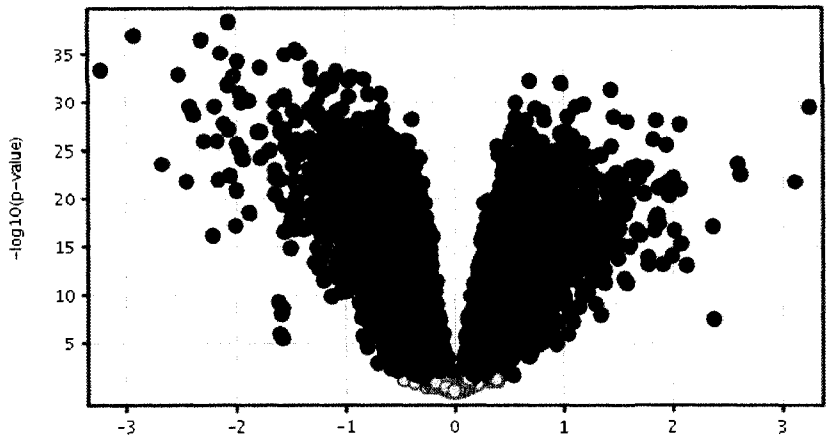
Dimensionality Reduction

Feature selection or dimensionality reduction was done by combining fold change and p-value using volcano plot, as explained in methods part of this dissertation. The fold change used was depending on the number of genes passing the filter for a particular class label and dataset, because in some class labels no genes passed the two fold change expression. In those cases, we decreased the fold level to 1.1 to get some differentially expressed genes (Table 6).

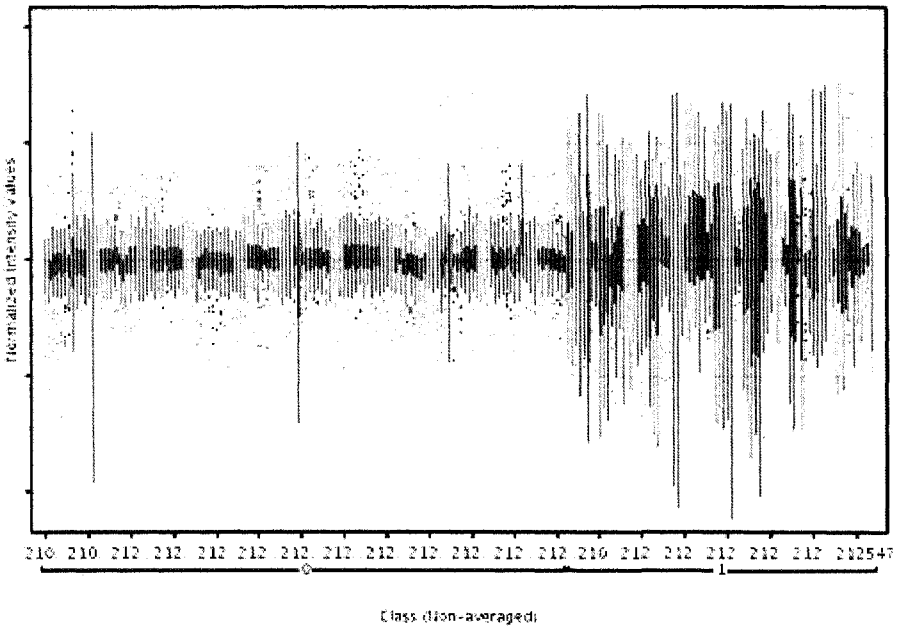
			BR		MM				NB			
[illegible]	[illegible]	[illegible]	pCR	erpos	OS_MO	EFS_MO	CPS1	CPR1	OS_MO	EFS_MO	NEP_S	NEP_R
600*	233*	602	106	197	2310*	1945*	876*	481*	199	112	139*	189*

Table 6: The differentially expressed genes passed the fold change (>2) and p-value (<0.05) using volcano plots for the total 13 end points. The '*' mark indicates the fold level is lowered to 1.1.

We observed the box plot distributions and scatter plots for each endpoint using the corresponding filtered differentially expressed genes for that endpoint. Here, I am showing the some images for few endpoints for visualizing the differentiation of the class labels. The distribution of differentially expressed genes for NIEHS class label, MDACC breast cancer pCR and *erpos* endpoints are shown in Figures 21, 22 and 23.

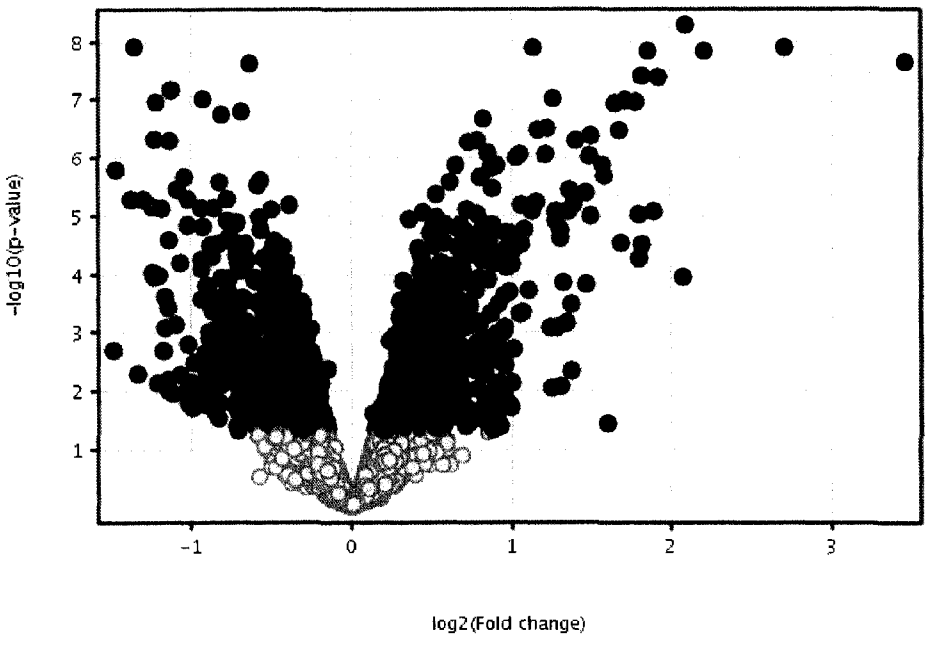


(a)

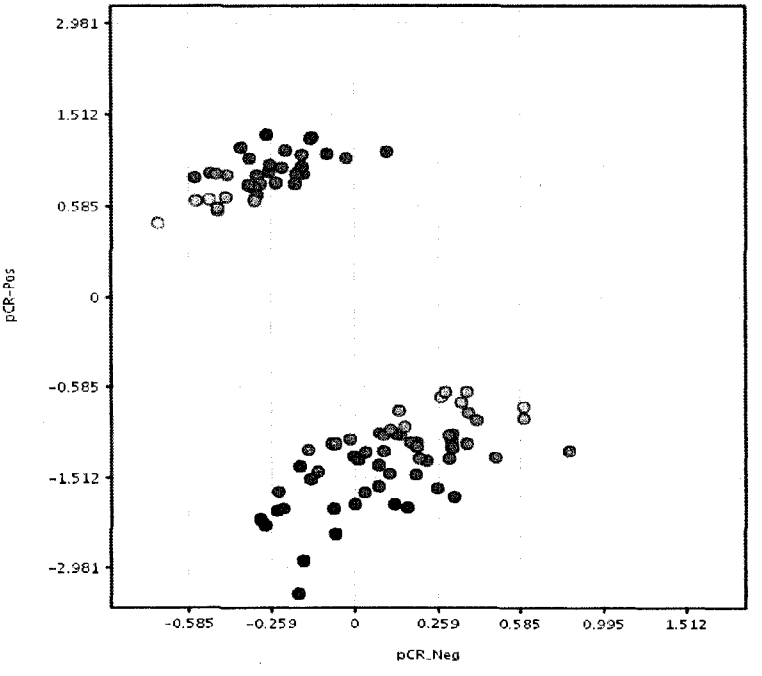


(b)

Figure 21: 512 differentially expressed genes (FC > 2 and P-value < 0.05) in NIEHS dataset with overall necrosis score as class label. (a) Volcano plot and (b) Box plot distribution of samples with differentially expressed genes, clearly shows the expression differentiation based on its class '1' (positive) and '0' (negative).

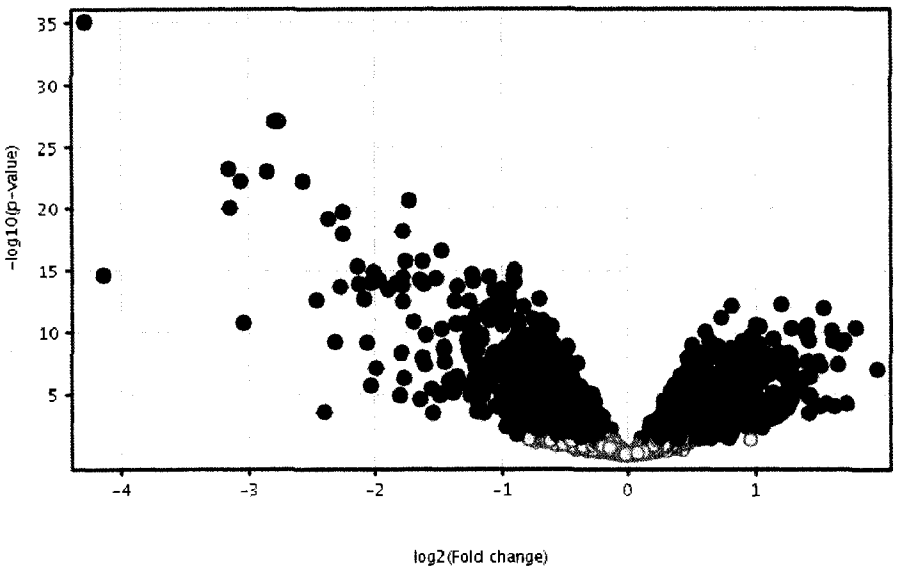


(a)

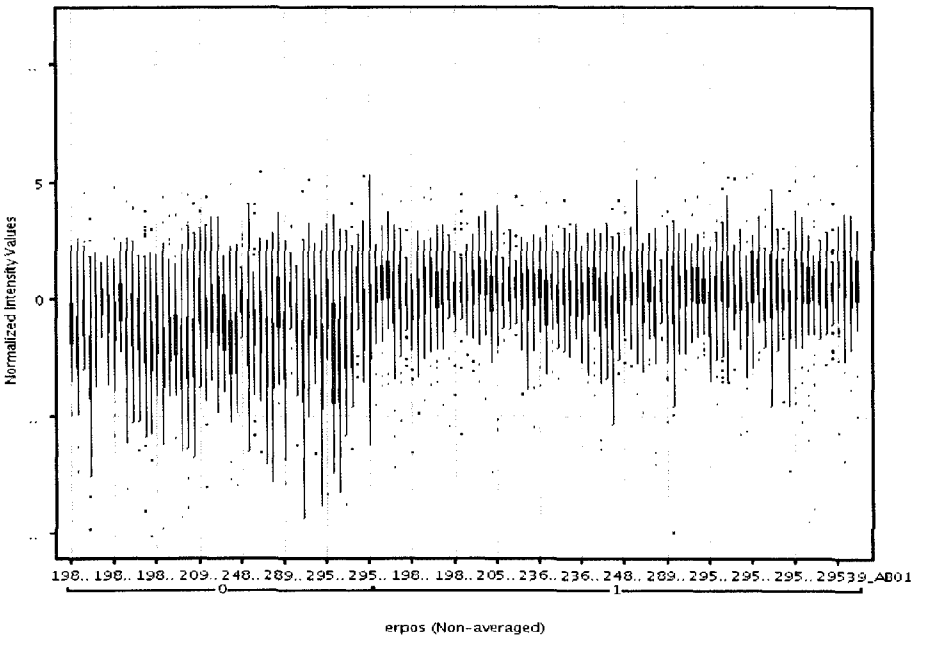


(b)

Figure 22: 106 differentially expressed genes (FC>2 and P-value < 0.05) in MDACC breast cancer dataset with pCR (pathological complete response) as class label. (a) Volcano plot and (b) Scatter plot distribution of samples with differentially expressed genes, clearly shows the expression differentiation based on its class pCR-Pos and pCR-Neg.



(a)



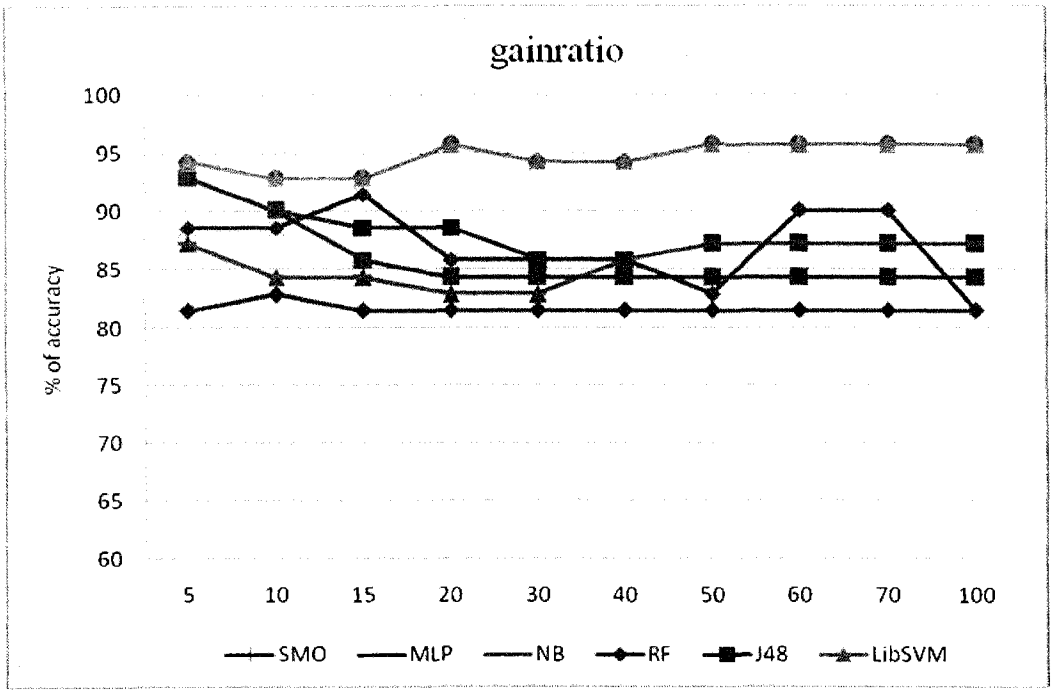
(b)

Figure 23: 197 differentially expressed genes (FC > 2 and P-value < 0.05) in MDACC breast cancer dataset with *erpos* (estrogen receptor positive) as class label. (a) Volcano plot and (b) Box plot distribution of samples with differentially expressed genes, clearly shows the expression differentiation based on its *erpos* class '1' (positive) and '0' (negative).

After dimensionality reduction of the expression data, our next step to follow is feature evaluation or feature selection methods, which detect and rank the best classifier genes for that particular endpoint.

Feature Selection / Evaluation

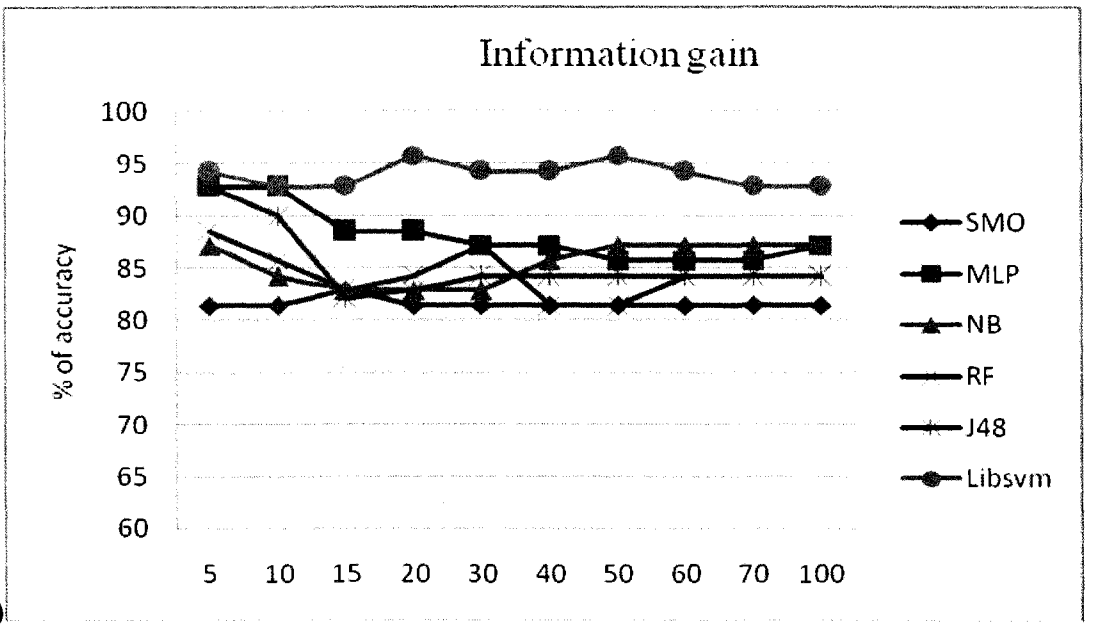
Feature selection or evaluation is very crucial step in developing better classifiers. So, we studied five feature selection or evaluation algorithms, namely gainratio (Table 7a), chi-square statistic (Table 7b), information gain (Table 7c), relief (Table 7d) and SVM (Table 7e) on Hamner lung tumor dataset and MDACC breast cancer dataset by applying SMO, LibSVM, Multi Layer Perceptron (MLP), NB, Random Forest (RF) and J48 classification algorithms with 10-f CV with 10 iterations to find the best algorithm.



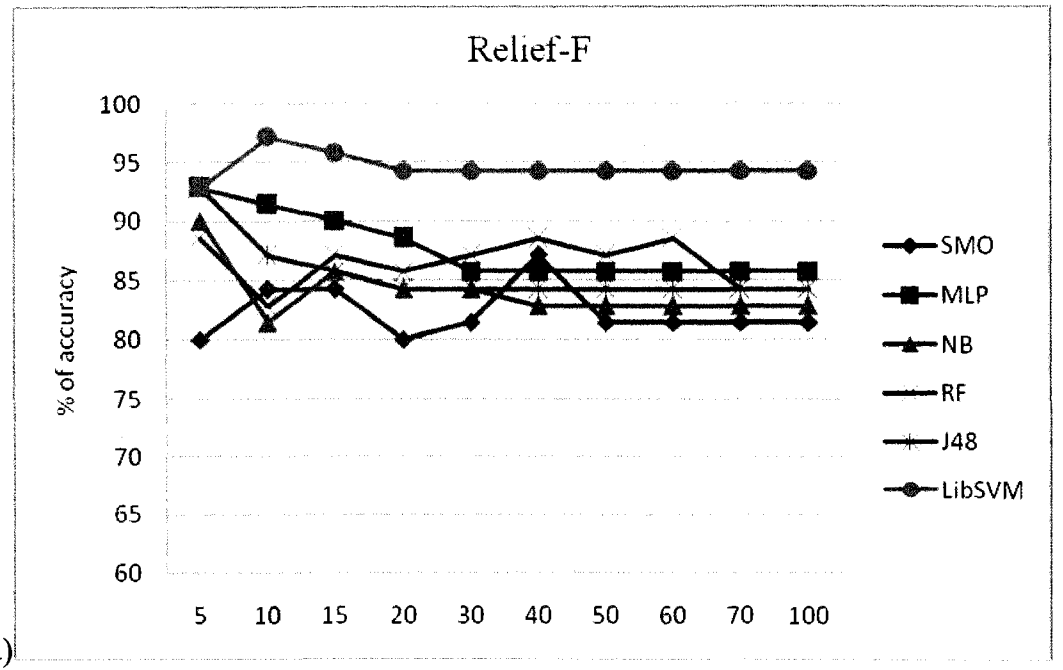
(a)


chi-square
100
95
90
85
80
75
70
65
60
% of accuracy
5
10
15
20
30
40
50
60
70
100
SMO
MLP
NB
RF
J48
LibSVM

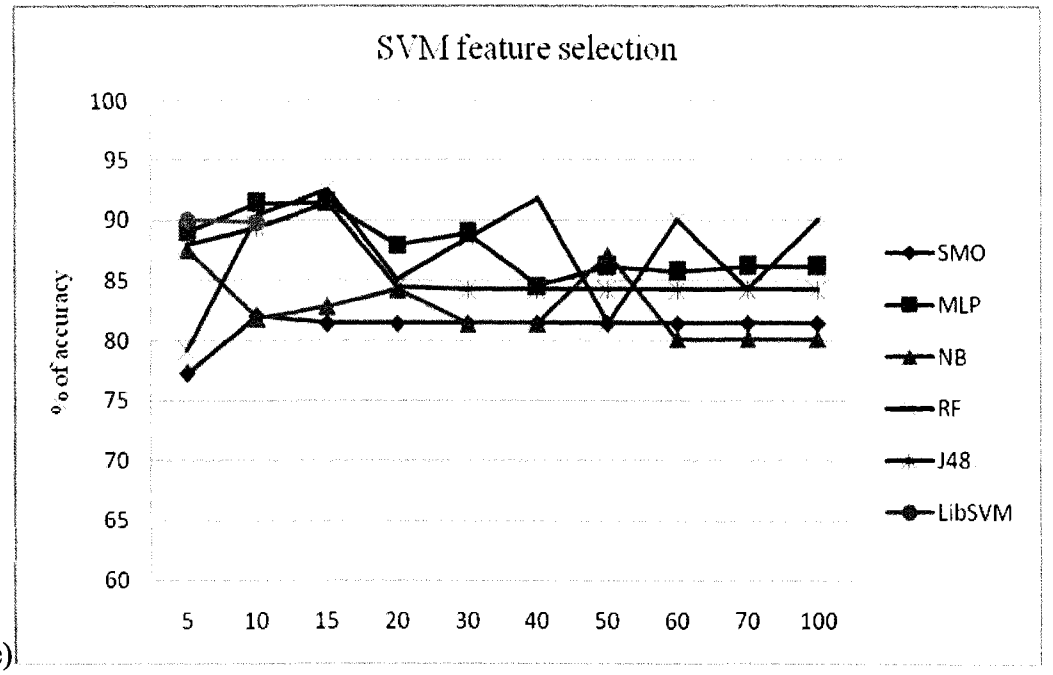
(b)


Information gain
100
95
90
85
80
75
70
65
60
% of accuracy
5
10
15
20
30
40
50
60
70
100
SMO
MLP
NB
RF
J48
Libsvm

(c)



(d)



(e)

Table 7: The five feature selection algorithms classification performance with six different classification algorithms. (a) Gain ratio (b) Chi-Square statistic (c) Information gain (d) relief-F and (e) SVM feature selection algorithms. Gain ratio algorithm performs better and consistent with all classification algorithms.

The feature selection algorithms also run with 10-f CV with 10 iterations. The performance of the internal CV error estimation was based on the percentage of accuracy. The graphs (Table 7 a, b, c, d, e) shown above from grain ratio performs better and consistent with all the classification algorithms we studied. Also, an information gain algorithm performs similar to gain ratio to some extent because of the similarity in their algorithms. From the above analysis on the Hamner dataset and also from breast cancer dataset, we decided to choose gain ratio algorithm as our choice of feature selection algorithm for further analysis in predicting classifiers.

After, deciding gain ratio as our feature selection algorithm, we implemented this feature selection algorithm within the cross validation of the classifier, also called as stratified cross validation.

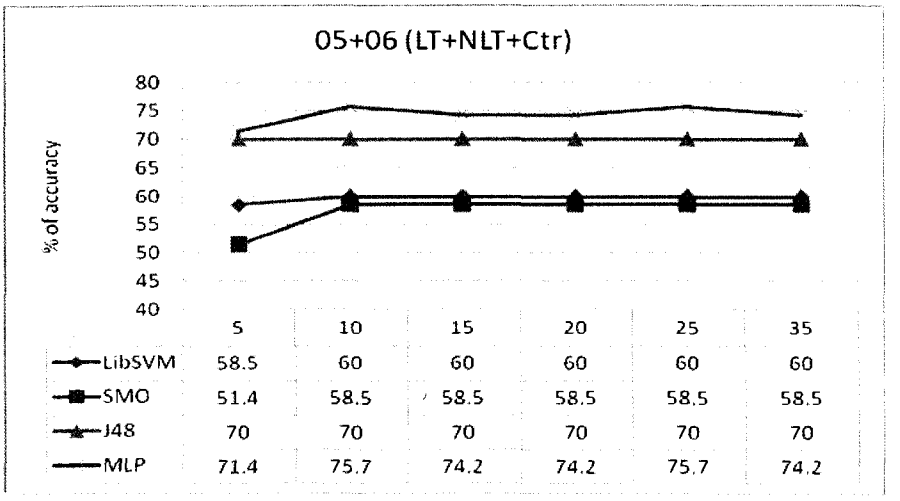
Classification / Error estimation

Initially, we performed several approaches and workflow designs to develop classifier models on the Hamner lung tumor dataset to overcome batch effect and other over-fitting bias. I will present those initial approaches and its brief results on the Hamner dataset before going to our final generic work flow for all the six datasets.

05+06 approach

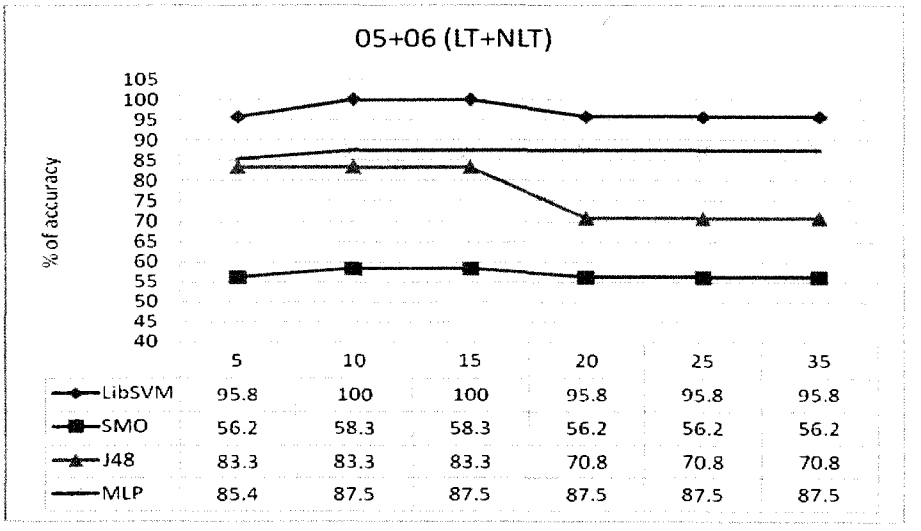
In the first approach, we studied three classes Lung Tumor, Non-Lung Tumor and Control (LT+NLT_Ctr) by combing year 2005 and 2006 (05+06) lung tumor data of Hamner with total 70 samples. In this three class prediction, the better performance came from

multilayer perceptron (MLP) with around 75 percentage of accuracy (Tables 8, 9). But the MLP algorithm is computationally expensive; it takes days, to even small matrix like Hamner dataset. It is not recommendable to do with big datasets like 200 – 300 samples.



	5	10	15	20	25	35
LibSVM	58.5	60	60	60	60	60
SMO	51.4	58.5	58.5	58.5	58.5	58.5
J48	70	70	70	70	70	70
MLP	71.4	75.7	74.2	74.2	75.7	74.2

Table 8: Classification performance of three class (LT, NLT and Ctr) prediction using Hamner lung tumor dataset with 05+06 approach.



	5	10	15	20	25	35
LibSVM	95.8	100	100	95.8	95.8	95.8
SMO	56.2	58.3	58.3	56.2	56.2	56.2
J48	83.3	83.3	83.3	70.8	70.8	70.8
MLP	85.4	87.5	87.5	87.5	87.5	87.5

Table 9: Classification performance of only two class (LT and NLT+Ctr) prediction using Hamner dataset with 05+06 approach.

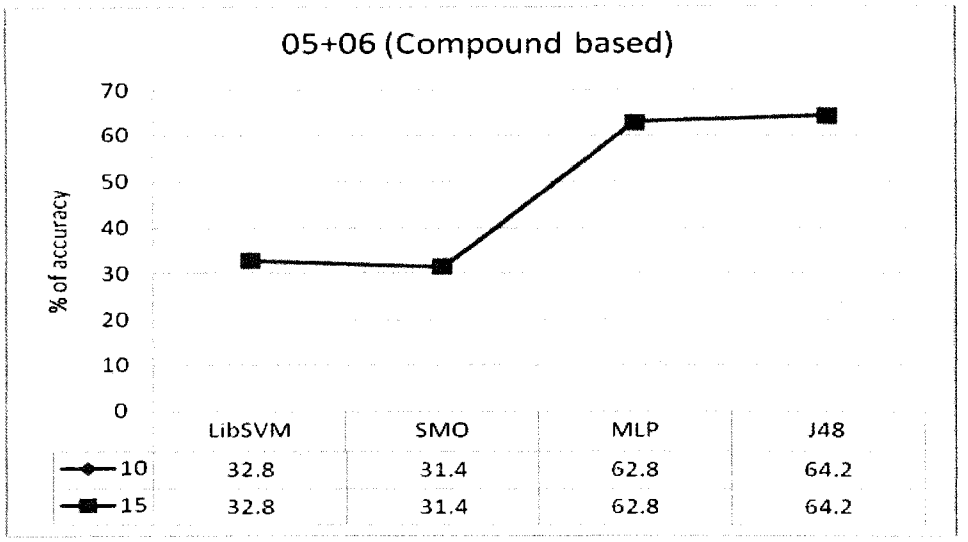


Table 10: Classification performance of chemical compound based (multi class) prediction using Hamner dataset with 05+06 approach.

Later,we studied the same using only two classes (LT and NLT), here we combined NLT and control samples and treated both of them as NLT class. In this case, the classification performance drastically increased with LibSVM with around 90 % of accuracy. This indicates that LibSVM or other SVMs perform better with two class dataset than three class or multi class datasets.

We also studied, the prediction of chemical compounds exposed to mice to study lung carcinogenicity using the Hamner dataset. There are total16 chemical compounds studied, among these, 7 lung carcinogenic, 6 non-carcinogenic chemicals and 3 control chemicals. Also the 70 samples are highly imbalanced between 2005 and 2006; 2005 has only total 18 samples (6-Ctr, 6-NLT and 6-LT) and 2006 has 52 samples in total (16-Ctr, 16-NLT and 20-LT). The above results (Table 10) shows very poor classification performance based on the chemical studied due to small number of samples and highly imbalanced classes. Because of

this poor performance, MAQC excluded the multiclass prediction from the studies and confined to only two class prediction for each endpoint.

Nested cross-validation with (05+06) and (06 →05) approaches

In the next appraoch, we implimented nested crossvalidation to avoid over-fitting bias with normal crossvalidation only on the test dataset samples using GEMS (Statnikov *et al.* 2005) tool. Nested crossvalidation (Figure 24) is embeding an another layer of crossvalidation within the training dataset of external crossvalidation for parameter tuning. There will two cross validations in this, the external cross validation for error estimation of the classifier and the internal cross validation for parameter tuning within the traning set of external CV.

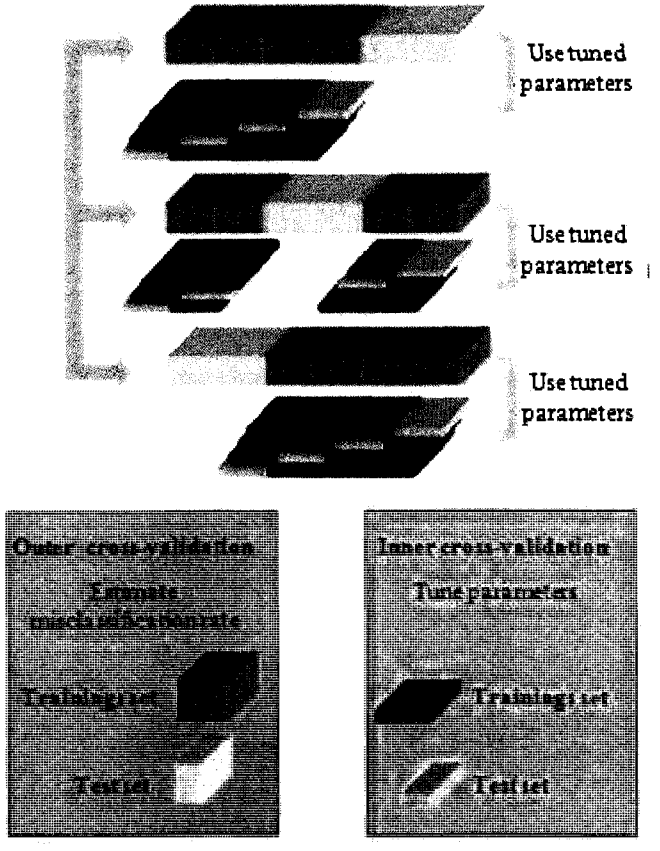


Figure 24: The schematic depiction of nested crossvalidation, with inner crossvalidation within the outer crossvalidation (Image courtesy from MCRestimate package in Bioconductor)

In the GEMS, we have the chance to use only several variants of SVM algorithms but several feature selection methods. It automatically builds the models and selects the best model by parameter (C and γ) tuning. We also tested our feature selection, gainratio method with this approach, performs better than the inbuilt features. The results are shown in the table (Table 11) below.

Classifiers	**Cross-validation design**	**Type of MC-SVM Classification (RBF with grid search)**	**Feature selection**	**Complexity**	**Best Model Characteristics**
Classifier 1	10-f CV (outer) 9-f CV (inner)	OVR, OVO	KW, S2N OVR, S2N OVO, BW	96011 models	**69.8% accuracy** OVR C=10, γ=0.1s 30 genes by Anova (KW)
Classifier 2	10-f CV (outer) 9-f CV (inner)	DAGSVM, WW	KW, S2N OVR, S2N OVO, BW	192011 models	**70.5% accuracy** WW C=10, γ=0.01 30 genes by Anova (KW)
Classifier *3*	10-f CV (outer) 9-f CV (inner)	CS	KW, S2N OVR, S2N OVO, BW	144011 models	**71.3% accuracy** CS C=1, γ=0.1 30 genes by Anova (KW)
Classifier 4	LOOCV (outer) 10-f CV (inner)	OVR, CS	KW, S2N OVR, S2N OVO, BW	739271 models	**67% accuracy** OVR C=10, γ=0.01 30 genes by Anova (KW)
Classifier 5	10-f CV (outer) 9-f CV (inner)	OVO, OVR, DAGSVM, CS	Gain Ratio from WEKA	192011 models	**72.8% accuracy** OVO C=10. γ=0.1

Table 11: The classification performance and the best classifiers using nested cross validation in GEMS. We approached 05+06 combined data for binary class prediction.

We also studied using 06 → 05 approach, in this, we used 2006 dataset to train and test with 2005 dataset. This results are shown below in the table (Table12).

Classifiers	**Cross-validation design**	**Type of MC-SVM Classification (RBF with grid search)**	**Feature selection**	**Complexity**	**Best Model Chracteristics**
Classifier 1	10-f CV (outer) 9-f CV (inner)	DAGSVM, WW	KW, S2N OVR, S2N OVO, BW	192011 models	**74.1% accuracy** WW C=10, γ=0.01 30 genes by Anova (KW)
Classifier *2*	10-f CV (outer) 9-f CV (inner)	CS	KW, S2N OVR, S2N OVO, BW	144011 models	**86.5% accuracy** CS C=1, γ=0.1 30 genes by Anova (KW)
Classifier 3	10-f CV (outer) 9-f CV (inner)	OVO, OVR, DAGSVM, CS	Gain Ratio from WEKA	192011 models	**89.7% accuracy** OVO C=10. γ=0.1

Table 12: The classification performance and the best classifiers using nested cross validation in GEMS. We used 06 →05 approach for binary class prediction (LT, NLT+Ctr).

From the above initial studies we observed that the 06 →05 approach performed better over combined (05+06) dataset and also binary class prediction gives better accuracy than the multi class prediction either three class (NT, NLT and Ctr) or chemical compound prediction. But nested cross validation using only SVM algorithms hinders our study using other types of algorithms. To avoid this, we used stratified cross validation by the recommendation of MAQC.

But after MAQC 7th face-to-face meeting in May 2007, analysis groups and RBWG (statisticians) recommended to use a generic work flow for all the datasets without much variation in the data analysis plan. They recommended in this meeting to submit a specific

single data analysis plan (DAP) (Appendix D) from each analysis group to the RBWG approval. This is because; the main objective of this project is to standardize a better work flow for predicting clinical outcomes and its reproducibility consistently with future datasets. So, the statisticians recommended only one data analysis plan from one group irrespective of dataset studying. This makes the work flow independent of the dataset being studied and has a chance to study the parameters affecting the classification performance.

Generic DAP from USM Group

We proposed our single generic data analysis plan (DAP) (Appendix D) for all the six datasets for biostatistics (RBWG) group approval. The main features of our data analysis plan are, (i) it includes fold change combined p-value as dimensionality reduction and gain ratio as feature selection algorithm. (ii) batch effect correction on the two datasets (iii) Feature selection algorithm is implemented inside the cross validation (called stratified cross validation). (iv) We proposed to use SMO (linear), LibSVM (linear and RBF kernels), NB, and Voted Perceptron classification algorithms on all the endpoints to generate candidate ("best" models) models. (v) MCC as the primary performance metric used to select candidate models due to highly imbalanced class datasets.

The schematic diagram of our final work flow is shown in the figure (Figure 25).

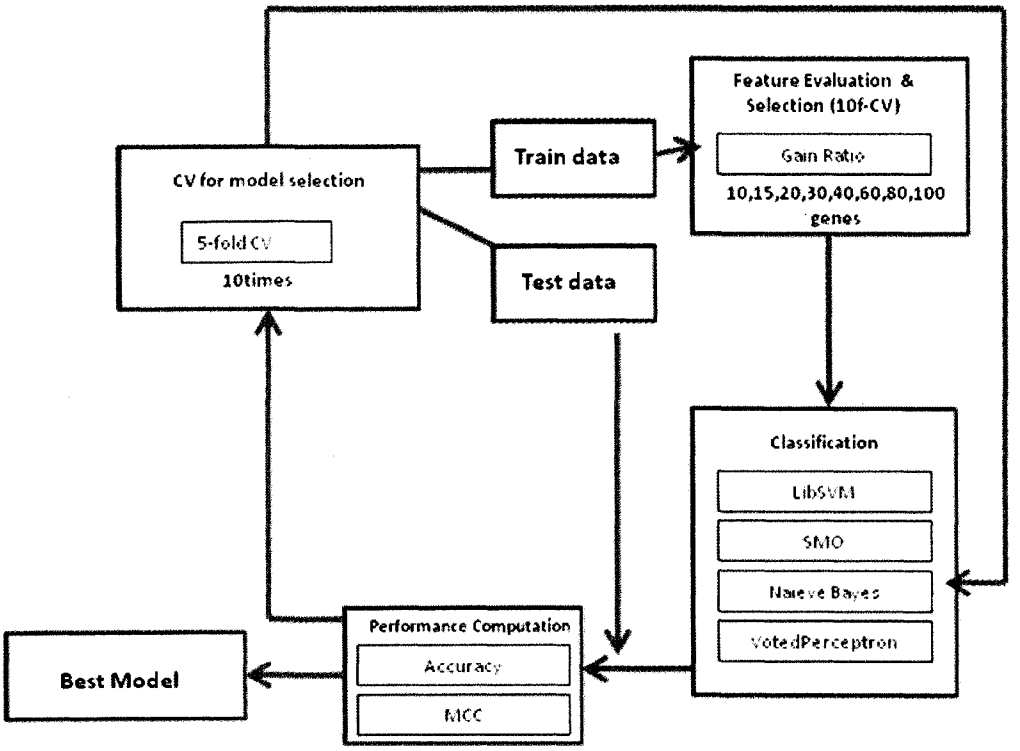


Figure 25: The schematic diagram of the data analysis plan we studied

Results for final candidate classifiers for each end point

We selected the candidate or best model for each end point based on the MCC performance from the models generated by 5-fold CV with 10 iterations and with 5 classification algorithms, so totally 250 models for each subset of gene lists generated by gain ratio algorithm. Also reported the popular performance metrics accuracy, sensitivity, specificity, AUC (area under ROC curve), RMSE along with MCC values for each model generated. Standard deviations are calculated for each performance metric from the models generated by the 10 iterations. We reported only the top five models for each endpoint along with the standard deviation values (Table 13-25). The first one among those five is our candidate model (best model) for that end point.

Table 13: The table shows the top five models for the NT_NLT class (A) of the Hamner lung tumor dataset. The first one in this table is the candidate model for this endpoint (or class).

Hamner lung tumor dataset

Endpoint studied: Lung tumor in mouse

Class code: NT_NLT (A)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.7566	0.881429	0.9419444	0.77933333	0.86064	0.311425	0.06316385	0.030897387	0.027976144	0.06137921	0.036342103	0.053760497
0.75107	0.877143	0.9308333	0.78733333	0.85908	0.318719	0.065270599	0.033128623	0.035823881	0.059499974	0.03686582	0.071846083
0.70416	0.855714	0.9130556	0.76266667	0.83786	0.344305	0.072949416	0.035283112	0.03880171	0.078391232	0.040398404	0.065918613
0.69122	0.85	0.9055556	0.75866667	0.83211	0.367686	0.072868776	0.036421568	0.03286711	0.059942359	0.039422529	0.056076376
0.61727	0.805714	0.7891667	0.83466667	0.83006	0.414617	0.030214279	0.015356681	0.023875606	0.029614811	0.011699725	0.012610905

Number features used for the top candidate model: 20 (features are shown in the appendix E)

Classification algorithm of the top candidate model: LibSVM with RBF kernel

Table 14: The table shows the top five models for the Class (B) of the Iconix liver cancer dataset. The first one in this table is the candidate model for this endpoint (or class).

Iconix dataset

End point studied: hepato-carcinogenicity in rats

Class code: Class (B)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.56179	0.80222	0.6949524	0.856133	0.77554	0.439861	0.040697175	0.018455501	0.02548022	0.018991782	0.018761424	0.022219505
0.55756	0.805412	0.4750476	0.97413793	0.72459	0.438335	0.013305324	0.004983583	0.013164463	0.003359158	0.00684379	0.005742132
0.53327	0.786934	0.704381	0.82827586	0.76633	0.457115	0.026475124	0.012805609	0.024368879	0.016648173	0.013662803	0.014409788
0.52791	0.785539	0.694381	0.8312069	0.76279	0.458733	0.028945876	0.013937801	0.02266431	0.015437345	0.014819361	0.016107819
0.52778	0.78834	0.6673333	0.85017241	0.75875	0.456441	0.01609267	0.006521255	0.023072399	0.009533042	0.009843787	0.008157422

Number features used for the top candidate model: 30 (features are shown in the appendix E)

Classification algorithm of the top candidate model: LibSVM with linear kernel

Table 15: The table shows the top five models for the Class (C) of the NIEHS dataset. The first one in this table is the candidate model for this endpoint (or class).

NIEHS dataset

End point studied: Overall necrosis score in rats

Class code: Class (C)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.38798	0.725316	0.8703704	0.47766667	0.67402	0.521715	0.032420393	0.013162227	0.017546507	0.034522313	0.015986743	0.014292512
0.37942	0.722536	0.8792593	0.455	0.66713	0.524011	0.030825558	0.010623696	0.013546535	0.036672979	0.015149181	0.010643893
0.36407	0.716478	0.8844444	0.42983333	0.65714	0.529329	0.034599255	0.012371288	0.012196631	0.033577348	0.016017683	0.012070429
0.34789	0.709911	0.8881481	0.40558333	0.64687	0.535926	0.034525195	0.010979322	0.014998857	0.032770833	0.014231492	0.009812884
0.14225	0.619181	0.7733333	0.35558333	0.56479	0.612413	0.056019134	0.026026526	0.032041735	0.034312134	0.036284223	0.02059882

Number features used for the top candidate model: 20 (features are shown in the appendix E)

Classification algorithm of the top candidate model: LibSVM with RBF kernel

Table 16: The table shows the top five models for the class pCR (D) of the MDACC human breast cancer dataset . The first one in this table is the candidate model for this endpoint (or class).

MDACC breast cancer (MDACC_BR) dataset

End point studied: Pathological complete response to treatment

Class code: pCR (D)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.57334	0.817692	0.8341579	0.77190476	0.86745	0.414906	0.023792846	0.008148845	0.01042185	0.020008816	0.009264803	0.008606792
0.54935	0.814615	0.8506316	0.71333333	0.86927	0.413335	0.028158953	0.013784287	0.015060767	0.020913866	0.007196209	0.009882633
0.53715	0.803077	0.8237895	0.7447619	0.87868	0.424548	0.015506678	0.006486723	0.005618419	0.012336101	0.004424741	0.006913577
0.50947	0.786923	0.8068947	0.73190476	0.81582	0.450254	0.035700938	0.012587628	0.019864809	0.050746506	0.016579444	0.017723428
0.50208	0.787692	0.8103158	0.72333333	0.87211	0.445617	0.014401729	0.006486723	0.004728388	0.013365419	0.004402099	0.005827232

Number features used for the top candidate model: 50 (features are shown in the appendix E)

Classification algorithm of the top candidate model: Naive Bayes (NB)

Table 17: The table shows the top five models for the class erpos (E) of the MDACC human breast cancer dataset . The first one in this table is the candidate model for this endpoint (or class).

MDACC breast cancer (MDACC_BR) dataset

End point studied: estrogen receptor postive (erpos)

Class code: erpos (E)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.92114	0.961538	0.9825	0.928	0.95525	0.153051	0.012950676	0.006280743	0.006454972	0.013984118	0.007331439	0.027671186
0.89095	0.946923	0.97375	0.904	0.93888	0.200971	0.016434195	0.007649453	0.007095578	0.018378732	0.009269312	0.029016711
0.8909	0.946923	0.97375	0.904	0.93888	0.203269	0.020200923	0.009209377	0.007095578	0.018378732	0.010662923	0.026571795
0.86894	0.936154	0.96875	0.894	0.97875	0.225511	0.011330693	0.005191912	0.008838835	0.00843274	0.003254271	0.019104457
0.85909	0.931538	0.965	0.878	0.9215	0.242307	0.013740641	0.006735346	0.005270463	0.011352924	0.007472171	0.014333722

Number features used for the top candidate model: 30 (features are shown in the appendix E)

Classification algorithm of the top candidate model: SMO

Table 18: The table shows the top five models for the class OS_MO (F) of the MM cancer dataset . The first one in this table is the candidate model for this endpoint (or class).

Multiple Myeloma (MM) dataset

End point studied: Overall survival (24months) milestone outcome (OS_MO)

Class code: OS_MO (F)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.50516	0.782353	0.6607143	0.84210526	0.751	0.4665	0.012377934	0.005849582	0.009339886	0.015319853	0.00710737	0.003676362
0.33322	0.656471	0.6180676	0.73482213	0.73038	0.546738	0.017378007	0.008060708	0.008365412	0.013977485	0.007892364	0.004862119
0.32519	0.651471	0.6114589	0.73304348	0.72246	0.55008	0.023654884	0.011026688	0.009337242	0.017735446	0.006819673	0.004661912
0.29998	0.699706	0.80657	0.48197628	0.64427	0.546604	0.034450287	0.013756967	0.016034314	0.032115797	0.016856616	0.012812047
0.26123	0.680588	0.7849855	0.46754941	0.62627	0.563857	0.031594906	0.013669353	0.01647016	0.02352762	0.014664296	0.012255184

Number features used for the top candidate model: 40 (features are shown in the appendix E)

Classification algorithm of the top candidate model: NB

Table 19: The table shows the top five models for the class EFS_MO (G) of the MM cancer dataset . The first one in this table is the candidate model for this endpoint (or class).

Multiple Myeloma (MM) dataset

End point studied: Even free survival milestone outcome (EFS_MO)

Class code: EFS_MO (G)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.32953	0.654471	0.688324	0.63324037	0.656	0.5665	0.034715458	0.017551484	0.023044962	0.025970483	0.017292073	0.01335553
0.06836	0.535294	0.5031629	0.56426984	0.53372	0.6809	0.019849683	0.009901475	0.018037015	0.017273888	0.009917915	0.007283939
0.05958	0.530882	0.4970076	0.56155556	0.52928	0.683919	0.039859276	0.019815081	0.030136961	0.033586658	0.019574826	0.014435194
0.05579	0.528824	0.4951705	0.55934921	0.52726	0.685474	0.035076538	0.016969568	0.031568466	0.014030734	0.017639108	0.012215296
0.04522	0.524706	0.4754924	0.56919048	0.52234	0.688432	0.049121618	0.024419441	0.030124844	0.028087061	0.024410153	0.017638526

Number features used for the top candidate model: 20 (features are shown in the appendix E)

Classification algorithm of the top candidate model: LibSVM linear kernel

Table 20: The table shows the top five models for the class CPS1 (H) of the MM cancer dataset . The first one in this table is the candidate model for this endpoint (or class).

Multiple Myeloma (MM) dataset

End point studied: Clinical parameter S1 (CPS1)

Class code: CPS1 (H)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.16953	0.606471	0.3504368	0.79882591	0.57463	0.626316	0.045809991	0.019695881	0.022338236	0.030275042	0.018860321	0.015823452
0.15857	0.599706	0.3751494	0.76832659	0.57174	0.631479	0.052416912	0.024586156	0.01891943	0.039979543	0.022682399	0.019116543
0.14914	0.596176	0.3552414	0.777139	0.56619	0.634252	0.041684586	0.018758568	0.01601736	0.034376225	0.01688562	0.014760497
0.1255	0.567647	0.5288046	0.59695007	0.57728	0.625436	0.004261983	0.001960784	0.003155842	0.002209455	0.003255714	0.002681645
0.12403	0.566765	0.5294943	0.59488529	0.5765	0.644505	0.006074922	0.002790245	0.004875454	0.002832544	0.002540701	0.00264335

Number features used for the top candidate model: 21 (features are shown in the appendix E)

Classification algorithm of the top candidate model: LibSVM with RBF kernel

Table 21: The table shows the top five models for the class CPR1 (I) of the MM cancer dataset . The first one in this table is the candidate model for this endpoint (or class).

Multiple Myeloma (MM) dataset

End point studied: Clinical parameter R1 (CPR1)

Class code: CPR1 (I)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.0936	0.571176	0.699	0.38857143	0.54379	0.65088	0.080758614	0.039161728	0.048177911	0.036762398	0.037889399	0.030081259
0.07011	0.562353	0.698	0.36857143	0.53329	0.658641	0.085215826	0.039528167	0.057310073	0.064873853	0.039704407	0.029745534
0.04821	0.548824	0.672	0.37285714	0.52243	0.669197	0.093272818	0.040093125	0.043153473	0.074215474	0.043274686	0.030264397
0.03856	0.554706	0.728	0.30714286	0.51757	0.664534	0.080724752	0.038026074	0.045166359	0.054398379	0.038675931	0.028368947
0.03678	0.550588	0.714	0.31714286	0.51557	0.667512	0.074780668	0.035097492	0.045264654	0.050305191	0.035237773	0.025699527

Number features used for the top candidate model: 40 (features are shown in the appendix E)

Classification algorithm of the top candidate model: LibSVM with linear kernel

Table 22: The table shows the top five models for the class OS_MO (J) of the NB cancer dataset . The first one in this table is the candidate model for this endpoint (or class). MCC values can range from -1 to +1, here 0 means it is moderate.

Neuroblastoma (NB) dataset

End point studied: Overall survival (900 days) milestone outcome (OS_MO)

Class code: OS_MO (J)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0	0.902943	0.958351	0.352	0.65518	0.308834	0	0.007997177	0.008178415	0.059404452	0.029403889	0.012663724
0	0.884016	0.9315328	0.414	0.67277	0.336336	0	0.015810054	0.017498896	0.086820377	0.0422678	0.023081336
0	0.906374	0.9986258	0	0.49931	0.305505	0	0.002012691	0.002212871	0	0.001106435	0.003191512
0	0.90469	0.9967759	0	0.49839	0.30818	0	0.002025234	0.00222532	0	0.00111266	0.00322314
0	0.90469	0.9967759	0	0.49839	0.30818	0	0.002025234	0.00222532	0	0.00111266	0.00322314

Number features used for the top candidate model: 20 (features are shown in the appendix E)

Classification algorithm of the top candidate model: LibSVM with linear kernel

Table 23: The table shows the top five models for the class EFS_MO (K) of the NB cancer dataset . The first one in this table is the candidate model for this endpoint (or class).

Neuroblastoma (NB) dataset

End point studied: Event-free survival milestone outcome (EFS_MO)

Class code: EFS_MO (K)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.64205	0.854424	0.85	0.87133333	0.89112	0.370068	0.008851623	0.002690732	0.002773928	0.012720483	0.002618812	0.005284102
0.63124	0.850665	0.8478947	0.86133333	0.88916	0.375975	0.010298948	0.002846789	0.002987612	0.013092475	0.00348637	0.004999724
0.62993	0.849406	0.8457895	0.86333333	0.88987	0.376041	0.013517033	0.004918069	0.003552361	0.015964639	0.002863713	0.005038674
0.62362	0.849424	0.8494737	0.84888889	0.8938	0.36993	0.010879301	0.00345511	0.004438284	0.014093554	0.005676657	0.004467685
0.55636	0.844007	0.8805263	0.70266667	0.82268	0.38964	0.033754801	0.011361069	0.017727113	0.048740516	0.011738489	0.014109056

Number features used for the top candidate model: 30 (features are shown in the appendix E)

Classification algorithm of the top candidate model: NB

Table 24: The table shows the top five models for the class NEP_S (L) of the NB cancer dataset . The first one in this table is the candidate model for this endpoint (or class).

Neuroblastoma (NB) dataset

End point studied: Newly established parameter S (NEP_S)

Class code: NEP_S (L)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.94271	0.971559	0.980381	0.96551724	0.97295	0.153092	0.000674732	5.7078E-05	0.000375624	0	0.000187812	0.008493038
0.94271	0.971559	0.980381	0.96551724	0.97295	0.153092	0.000674732	5.7078E-05	0.000375624	0	0.000187812	0.008493038
0.94271	0.971559	0.980381	0.96551724	0.97295	0.153092	0.000674732	5.7078E-05	0.000375624	0	0.000187812	0.008493038
0.94271	0.971559	0.980381	0.96551724	0.97295	0.153092	0.000674732	5.7078E-05	0.000375624	0	0.000187812	0.008493038
0.93736	0.968702	0.980381	0.96068966	0.96456	0.160209	0.003467312	0.001976192	0.000375624	0.003331351	0.002255416	0.010783818

Number features used for the top candidate model: 20 (features are shown in the appendix E)

Classification algorithm of the top candidate model: SMO

Table 25: The table shows the top five models for the class NEP_R (M) of the NB cancer dataset . The first one in this table is the candidate model for this endpoint (or class).

Neuroblastoma (NB) dataset

End point studied: Newly established parameter R (NEP_R)

Class code: NEP_R (M)

MCC	Accuracy	Sensitivity	Specificity	AUC	RMSE	MCC_StdDev	Accuracy_StdDev	Sensitivity_StdDev	Specificity_StdDev	AUC_StdDev	RMSE_StdDev
0.33378	0.679249	0.5667143	0.75793103	0.73024	0.489944	0.031816529	0.014494122	0.01934614	0.013580741	0.012253255	0.009618326
0.31834	0.676449	0.4775238	0.81517241	0.73369	0.522484	0.034066868	0.015494072	0.024765771	0.019178532	0.012281549	0.008624968
0.31133	0.672343	0.4943333	0.79655172	0.73868	0.505792	0.032766927	0.013840623	0.017564552	0.016336403	0.012881836	0.007890078
0.28927	0.665886	0.4310476	0.82965517	0.73738	0.524527	0.02732898	0.011705707	0.021312206	0.01455737	0.012541285	0.006971676
0.26455	0.646743	0.5289524	0.72896552	0.62896	0.591981	0.059578972	0.026406567	0.057285008	0.024121505	0.030222221	0.022323261

Number features used for the top candidate model: 20 (features are shown in the appendix E)

Classification algorithm of the top candidate model: NB

Among this 13 candidate models for 13 end points proposed by USM group, our candidate models chosen best for five end points based on the best MCC performance and better analysis plan accepting most of the RBWG recommendations. Here, I am providing the exact response sent by Campbell Gregory (FDA) with ratings and comments given by the RBWG.

Recommendations

"The 13 groups that reported all 13 endpoints *and* submitted a DAP are:

CAS, CBC, Cornell, FBK, GeneGo, GHI, GSK, NCTR, SAI, Tsinghua, UIUC, USM, ZJU

It is surprising SAS and NWU did not submit a DAP – presumably they did their analysis correctly, but it is unfair to assume so without reading the plan. Of these 13 groups, the following 7 appear to have the analysis done right, with the caveat that the more models the group chose from, the higher the scores are likely to be, representing greater overfitting. The two groups for which this is the largest problem are likely to be NCTR for the sheer number of models, and SAI with its very high standard deviations for model quality measures. CBC and Cornell also have an uncomfortably large number of models for my liking.

CBC, Cornell, GHI, NCTR, SAI, USM, and ZJU

Three additional groups (GSK, Tsinghua, and UIUC) could be included in this list if their write-ups provided better clarity that they were not somehow snooping/overfitting the data. SAS and NWU could also be considered if they submitted a write-up.

Based on the small chance of over-fitting, if I had to bet money on which models would hold up out of sample and be of high quality, I would select from only the following groups (and still look carefully under the hood):

GHI, USM, and ZJU

Taking these three groups and ranking models by MCC, and choosing the top model by MCC gives the following selections I would make for the 13 endpoints, pending verification of the methodologies of the top 3 groups.

A: ZJU
B: GHI
C: ZJU
D: USM
E: USM
F: USM
G: ZJU
H: GHI
I: ZJU
J: ZJU
K: USM
L: USM
M: GHI

For all three groups, batch effect correction was not really addressed. Hence we need to think carefully about potentially rerunning these methodologies on batch effect corrected data for those studies where the validation set comes from a different array type or there is an expectation of large batch differences between training and validation sets."

From comments and observations especially on the batch effect made by RBWG came from our mistake in filling the batch effect column in the analysis plan. I mentioned

about the batch effect correction method we applied in the summary of DAP, was not observed by the reviewers. Then we intimated about this mistake and corrected.

Discussion

This dissertation is about how to effectively apply data mining technologies to biological and clinical expression data. Some problems arising from gene expression profilings like batch effect are studied in depth using data mining techniques of feature generation, selection and integration with classification algorithms. Also this analysis effort in conjunction with MAQC consortium helps to facilitate a standard work flow for predicting better and reproducible classifiers using gene expression data.

Initially, we participated in array outlier identification analysis with other members to identify a consensus array outliers. The purpose of this QC assessment exercise is to reach consensus on a subset of arrays that should be considered with reasonable confidence as outliers due to array quality concerns and also clear outliers would impact the performance of classifiers significantly. Our outlier identification using dChip and box-plot distribution performed well based on the meta-analysis.

Our observations in the preprocessing stages indicated that the strong batch noise introduced at the level of array making could affect the performance of classifiers significantly. To overcome this difficulty we applied two batch adjusting algorithms in this study. In our compare and contrast studies of these batch adjustment algorithms, (Combat and Batchmatch) we observed that both performed similarly in correcting the batch but Batchmatch has several advantages over Combat in terms of class label information leakage

and batch reference. These features could help more when we are analyzing the performance of classifiers with external validation or blind datasets.

In order to identify genes associated with disease phenotype classification or patient survival prediction from gene expression data, we compared and analyzed the performance of five feature selection algorithms. Our observations from these studies, indicated that gainratio algorithm performs better and consistent over the other algorithms studied. This makes to take gainratio as our feature evaluation algorithm for further classification studies with other datasets.

When it comes to performance metric to choose the best classifiers, MAQC recommends of using Matthews' correlation coefficient (MCC) as primary performance metric especially when we are dealing with highly imbalanced class datasets. Because of, MCC takes all four elements of the two class confusion matrix into consideration avoids you the bias. our observation and studies strengthen the above recommendation that, MCC gives unbiased performance results over accuracy in some endpoints (K and M), where class imbalance is more.

In the aspect of classification algorithms, no single algorithm is absolutely superior to all others, though SVM achieved fairly good results in most endpoints. Naive bayes algorithm also performed well in some endpoints. In overall, from the total 60 models we reported (5 top models for 13 end points) SVM and SMO (a variant of SVM) dominates mostly, also the linear kernel performed well over RBF in our binary classifications.

Future work

Currently, our proposed generic data analysis work flow for this classification purpose would help along with other groups analysis to come to a conclusion about the standardize methodology. But the candidate models for all the endpoints generated from these training datasets from all groups should validate with the external blind datasets to know how well they perform with unknown data. Presently, the consortium is working hard on providing the blind datasets to analysis groups, most probably by the end of this month.

As part of this study and discussions with in the consortium, several groups proposed the manuscript ideas to publish the work done by the analysis groups by meta analysis. Among one of them, Dr. Deng proposed meta-analysis of gene features used in the candidate models across the groups in all endpoints. This study facilitates finding the consensus gene lists using the genes the groups used for their candidate model for a particular end point. After finding the consensus gene lists, ranking them based on the number of occurrences in the candidate models. Generating the new classifiers and comparing the performance validation with already known candidate models using the top consensus gene list could be an interesting work. We are currently working on these analysis after getting the summarized results from other groups.

We are also working the effect of batch on the classification and how could we quantify the batch noise along with other groups.

APPENDIX - A

Summary of MAQC-II Datasets (provided by Leming Shi, FDA)

Table 1. Summary of Datasets Being Analyzed by MAQC-II

	Toxicogenomics Datasets			**Clinical Datasets**		
Data Set #	**1**	**2**	**3**	**4**	**5**	**6**
Data Set Code	**Hamner**	**Iconix**	**NIEHS**	**BR**	**MM**	**NB**
Study Type	Lung Tumor (Mouse)	Liver Carcinogen (Rat)	Necrosis (Rat)	Treatment Outcome (Human, Breast Cancer)	Treatment Outcome and Survival (Human, Multiple Myeloma)	Overall Survival (Human, Neuroblastoma)
Training Data Sets						
Provider of Training Data	Hamner	Iconix	NIEHS*	MDACC	UAMS*	Cologne
Microarray Platform	Affymetrix Mouse 430 2.0	GE Healthcare CodeLink RU1	Affymetrix Rat 230 2.0	Affymetrix U133A	Affymetrix U133Plus2.0	Agilent NB Customized Array
No. of Training Samples	70	216	214	130	340	246
Number of Outlier Arrays	0	0	0	+19	+5	+5
Endpoint I	Lung Tumor	Liver Carcinogen	Overall Necrosis Score	Treatment Response	Overall Survival Censor	Overall Survival (<1000 days as "1")

Excel Column	**Class_LT_NLT** (C)	**Class** (B)	**Class** (C)	**pCR** (O)	**OS CENSOR** (1=death) (W)	**OS** (P)
Positives to Negatives Ratio	0.59 (26/44)	0.51 (73/143)	0.58 (79/135)	0.34 (33/97)	0.49 (112/228)	0.32 (59/187)
Endpoint II				Estrogen Receptor Status	Event-free Survival Censor	Newly Established Parameter S
Excel Column				**erpos** (H)	**EFS CENSOR** (1=event) (U)	**NEP_S** (AB) - *new*
Positives to Negatives Ratio				1.6 (80/50)	1.1 (179/161)	1.44 (145/101)
Endpoint III					Clinical Parameter S1	Newly Established Parameter R
Excel Column					**CPS1** (S)	**NEP_R** (AC) - *new*
Positives to Negatives Ratio					1.33 (194/146)	1.44 (145/101)
Endpoint IV					Clinical Parameter R1	Event-free Survival (<1000 days as "1")
Excel Column					**CPR1** (T)	**EFS** (L)
Positives to Negatives Ratio					1.43 (200/140)	0.65 (97/149)

Validation Data Sets						
Provider of Validation Data	Hamner	Iconix	NIEHS	MDACC	UAMS	Cologne
Microarray Platform	Affymetrix Mouse 430 2.0	GE Healthcare CodeLink RU1	Affymetrix Rat 230 2.0	Affymetrix U133A	Affymetrix U133Plus2.0	Agilent Customized NB
No. of Validation Samples	40	201	204	~150	214 (received) ~150 (Expecting) ~500 (with different treatments, A/B or Plus2 arrays)	200~300
Provider of Validation Data		EPA			Millennium Pharmaceuticals	
Microarray Platform		GE Healthcare CodeLink RU1			Affymetrix U133A/B (with 4-6 technical replicates per sample)	
No. of Validation Samples		55			~300	
Provider of Validation Data					Montpellier-Heidelberg	
Microarray Platform					Affymetrix U133A/B or Plus2	
No. of Validation Samples					130	

APPENDIX - B: Summary of outlier-voting results on the Hamner lung tumor data set

Status	RMS	NCTR	CAS	Ligand	USM	SAI	SAS	Hamner	Genedata	NIEHS_JC	NIEHS_JL	UAB	SIB	CSHS	AHIE
Good	**70**	**70**	**50**	**59**	**59**	**68**	**62**	**70**	**67**	**70**	**65**	**62**	**67**	**59**	**68**
Marginal	**0**	**0**	**15**	**4**	**10**	**2**	**6**	**0**	**3**	**0**	**5**	**3**	**2**	**6**	**2**
Outlier	**0**	**0**	**5**	**7**	**1**	**0**	**2**	**0**	**0**	**0**	**0**	**5**	**1**	**5**	**0**

Median number of good, marginal, and outlier: **67**, **3**, and **0**.

No.	Array	RMS	NCTR	CAS	Ligand	USM	SAI	SAS	Hamner	Genedata	NIEHS_JC	NIEHS_JL	UAB	SIB	CSHS	AHIE	**Consensus (%)**
55	GSM142182.CEL	0	0	1	0.5	1	0	0.5	0	0	0	0.5	0	1	0.5	0	**33.3**
30	GSM142157.CEL	0	0	1	1	0	0	0.5	0	0.5	0	0.5	0	0.5	0	0	**26.7**
20	GSM142147.CEL	0	0	0.5	0	0.5	0	1	0	0	0	0	1	0	0	0.5	**23.3**
24	GSM142151.CEL	0	0	1	1	0	0	0.5	0	0.5	0	0.5	0	0	0	0	**23.3**
56	GSM142183.CEL	0	0	1	0	0.5	0	0.5	0	0	0	0	0	0.5	1	0	**23.3**
32	GSM142159.CEL	0	0	0.5	1	0	0	0	0	0	0	0.5	0	0	1	0	**20.0**
2	GSM142129.CEL	0	0	0.5	0.5	0.5	0	0	0	0	0	0	1	0	0	0	**16.7**
13	GSM142140.CEL	0	0	0.5	1	0	0	0	0	0	0	0	0	0	0.5	0.5	**16.7**
23	GSM142150.CEL	0	0	0.5	0	0	0	1	0	0	0	0	1	0	0	0	**16.7**
62	GSM142189.CEL	0	0	1	0	0.5	0	0.5	0	0	0	0.5	0	0	0	0	**16.7**
69	GSM142196.CEL	0	0	0.5	1	0	0	0	0	0	0	0	0	0	1	0	**16.7**
1	GSM142128.CEL	0	0	0.5	0	0.5	0	0	0	0	0	0	1	0	0	0	**13.3**
6	GSM142133.CEL	0	0	0.5	1	0	0	0	0	0	0	0	0	0	0.5	0	**13.3**
4	GSM142131.CEL	0	0	0.5	0	0	0	0	0	0	0	0	0	0	1	0	**10.0**
5	GSM142132.CEL	0	0	0	0	0.5	0	0	0	0	0	0	1	0	0	0	**10.0**
19	GSM142146.CEL	0	0	0.5	0	0	0	0.5	0	0	0	0	0.5	0	0	0	**10.0**
27	GSM142154.CEL	0	0	0	1	0	0	0	0	0	0	0	0	0	0.5	0	**10.0**
3	GSM142130.CEL	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0.5	0	**6.7**

APPENDIX -C: Summary of outlier-voting results on the MDACC breast cancer dataset

Status	MDACC	NCTR	RMS	EA	UAB	Genedata	SIB	SAS	SAI	CAS	USM	Ligand	CSHS	AHIE
Good	161	157	164	151	142	122	161	163	160	66	134	126	136	167
Marginal	3	9	3	7	12	32	5	7	10	84	39	18	12	6
Outlier	14	12	11	20	24	24	12	8	8	28	5	34	30	5

Median number of good, marginal, and outlier: **154**, **10**, and **13**.

No.	Array	MDACC	NCTR	RMS	EA	UAB	Genedata	SIB	SAS	SAI	CAS	USM	Ligand	CSHS	AHIE	Consensus (%)
169	U133A_FL151_US129_12_08_05	1	1	1	1	1	1	1	1	1	1	1	1	1	1	**100.0**
143	29539_AB01833733_35649	1	1	1	1	1	1	1	1	0.5	1	1	1	1	1	**96.4**
171	U133A_FL175_US147_01_13_06_2	1	1	1	1	1	1	1	1	1	1	0.5	1	1	1	**96.4**
172	U133A_FL32-US2_05_19_05	1	1	1	1	1	1	1	1	1	1	0.5	1	1	1	**96.4**
170	U133A_FL161_US125_01_10_06	1	1	1	1	1	0.5	1	1	1	1	0.5	1	1	1	**92.9**
166	U133A_FL136_US123_11_14_05	1	1	1	1	1	1	1	1	0.5	1	0.5	1	1	0.5	**89.3**
175	U133A_FL80_US97_09_01_05	1	1	1	1	1	1	1	1	0.5	1	0.5	1	1	0.5	**89.3**
165	U133A_FL112_US120_10_13_05	1	0.5	1	1	1	1	1	1	0.5	1	0.5	1	1	0.5	**85.7**
168	U133A_FL15_03_17_05	1	1	1	1	1	1	1	0.5	0.5	1	0.5	1	1	0.5	**85.7**
174	U133A_FL78_US92_09_01_05	1	1	1	1	1	0.5	0.5	0.5	1	1	0.5	1	1	0.5	**82.1**
109	28998_AB02091099_34966	1	1	0	1	0.5	1	1	0.5	1	1	1	1	1	0	**78.6**
173	U133A_FL46-314_07_08_05	1	0.5	1	1	1	1	0.5	0.5	0.5	1	0.5	1	1	0.5	**78.6**
70	23678_AB01562100_26133	1	1	0.5	1	0.5	1	0.5	0	1	1	1	1	1	0	**75.0**
69	23678_AB01562100_24635	0	1	0.5	1	0	0.5	0	0	1	1	1	1	1	0	**57.1**
162	FL398-PERU53	0.5	0.5	0	1	1	0	0.5	0.5	0.5	1	0.5	1	1	0	**57.1**
164	FL454-713	0.5	0.5	0	1	1	0.5	0.5	0.5	0.5	1	0	0.5	1	0	**53.6**
167	U133A_FL137_US134_11_14_05	1	0.5	0	0.5	0	1	1	0	0	0.5	0	1	1	0	**46.4**
163	FL412-PERU55	0.5	0.5	0	1	0	0.5	0	0.5	0.5	1	0.5	0.5	0.5	0	**42.9**
135	29539_AB01833522_35706	0	0	0	1	0	1	1	0	0	0.5	0.5	1	0.5	0	**39.3**
86	24817_AB02261505_26168	0	0	0	1	0	1	0	0	0	1	0	0.5	1	0	**32.1**
75	23678_AB01562130_24648	0	0	0	0	0	1	0	0	0	1	0.5	0.5	1	0	**28.6**
16	19893_AB01983478_17035	0	0	0.5	0	0	1	0	0	0	0.5	0	1	0	0	**21.4**
74	23678_AB01562129_26136	0	0	0	0	1	0.5	0	0	0	0.5	0	0	1	0	**21.4**
77	23678_AB01562152_24646	0	0	0	0	0	0.5	0	0	0	1	0.5	0	1	0	**21.4**
89	24817_AB02262603_26162	0	0	0	0	0	1	0	0	0	1	0	1	0	0	**21.4**
92	24817_AB02263363_26173	0	0	0	0	0	0.5	0	0	0	1	0	1	0.5	0	**21.4**
97	24817_AB02263400_26163	0	0	0	0.5	0	0.5	0	0	0	1	0	1	0	0	**21.4**
141	29539_AB01833728_35659	0	0	0	0.5	0	0.5	0	0	0	0.5	0.5	1	0	0	**21.4**
62	23678_AB01542166_26134	0	0	0	0	0	1	0	0	0	0.5	0	0	1	0	**17.9**

APPENDIX - D: USM Data Analysis Plan (DAP)

Part A: Data Analysis Team						
University of Southern Mississippi **Venkata Thodima (venkata.thodima@usm.edu, 601-266-4353)** Primary Contact - **Dr. Youping Deng (youping.deng@usm.edu, 601-266-6678)**						
Part B: DAP Summary						
USM will analyze six datasets with four different machine learning algorithms (LibSVM, SMO, Naieve Bayes and Voted perceptron), and select one best model for each dataset as final model. We omitted the outliers from analysis specified in each dataset by the QC working groups. Batch effect will be treated in the Hamner and Iconix datasets based on parametric and nonparametric Empirical Bayes frameworks available in one of R function (Combat). We will use fold level and P-value (0.05) filter for dimensionality reduction and further ranking of features will be based on Gainratio feature selection algorithm.						
Part C: Brief Data Description – see Appendix A for more details						
Dataset Type	Toxicogenomic			Clinical		
Dataset Source	Hamner	Iconix-EPA	NIEHS	MDACC	UAMS	U Cologne
Disease or Toxicity (s)	Lung Cancer	Liver Toxicity	Rat Liver Toxicity	Breast Cancer	Multiple Myeloma	Neuroblastoma
Primary Prediction Endpoint (s)	Predict NTP long term assay lung carcinogenicity from 3 month exposure	Predict NTP long term hepatotoxicity from 5 to 7 day exposure	Liver necrosis	Treatment Outcome & Prognosis	Subtype & Treatment Outcome: a) even free survival (EFS); b) overall survival (OS);	Subtypes and three year: a) even free survival (EFS) b) overall survival (OS);
Microarray Platform	Affy 430.2 (Mouse)	Codelink-RU1 (Rat)	Affy-RG230_2 (Rat)	Affy HG_U133A (human)	Affy HG_U133_plus_2 (Human)	Agilent-NB-10707 (Human Custom)
Channel(s)	1	1	1	1	1	2 (Dye Swap)
Training Samples after QC	70	216	214	130	340	246
Sample QC	By QC subgroup (1)	By QC subgroup	By QC subgroup	By QC subgroup	By QC subgroup	By QC subgroup
Batch effect	Strong (2), Batch effect corrected	Strong (3), Batch effect corrected	Slight and equivocal (4)	Slight and equivocal	Slight and equivocal	Slight and equivocal
Other Endpoint (s)	See Appendix A We will think about the other endpoints after some more clarity about UAMS and Cologne datasets, seems the CWG will update about this soon according to the teleconference.					
Part D: Data Pre-Processing						
Raw data preprocessing	None	None	None	None	None	Background substraction (fg-mean BG)
Transforms	Baseline transformation and Log_2					
Summarization (for probe-level data)	MAS5		MAS5	MAS5	MAS5	NA
Normalization		Median scale 1000 (provided by MAQC)				Agilent Mean scale normalization

Part E.1: Classification Method (s) (Part E is intended to be a simplified matrix of the modeling algorithm (s) that will be used on each dataset. However, note that the final model will have to be based on one algorithm of the data analysis team's choosing.				
Chronology and parameter space reduction	Chronology of dataset analysis: In the anticipation that there may be some learning and reduction of the modeling process across each dataset considered, an indication of the chronology of analysis may be appropriate. For example, if after the first two datasets considered it is determined that one class of models is either too computationally expensive or appears to be leading to poor results, the class of models may be dropped for subsequent datasets. Alternatively, model parameter spaces (e.g., number of genes) may be reduced.			
Classification Method	LibSVM 5	SMO 6	Naieve Bayes 7	Voted perceptron 8
Classification Method(s) Eliminated	We eliminated KNN, J48 and Bagging algorithms because of poor performance in our exploratory investigation.			
Part E.2: Molecular Feature Filtering (i.e., features removed) and Selection (Note that some AGs might imbed or nest feature selection within the modeling algorithm such as by a Monte Carlo, genetic algorithm or within cross validation, etc., or by some combination of these. Other approaches could be based on biological insights or relevance, such as disease-related genes or pathways – there are endless possibilities)				
A Priori Feature Reduction Filtering	Filtering based on low signals and flags (P/M absolute calls)			
A Priori Feature Pool Selection	Further reduction in number of genes based on the fold change (2) but in some end points less than 2 and P-value (<0.05)			
Features removed based on biological considerations	None			
Features selection through cross validation	Gain ratio 11			
	The above feature selection method with 5f cross validation			
Part E.3: Non-Molecular Feature Selection (i.e., features added other than from the microarray data)				
Features derived from clinical data	NO			
Features derived from disease-associated genes, proteins and/or pathways	NO			
Features derived from other *in vitro* or *in vivo* data sources	NO			
Part E.4: Internal Validation (e.g., training and internal test set split and or Cross-validation (CV)				
Training set samples				
Internal test set set-aside				

samples						
OR by Cross validation	5f CV	5f CV	5f CV	5f CV	5f CV	5f CV
Level of Cross-validation	Stratified CV with 10 iterations					
Part E.5: Model Tuning Criteria by Dataset						
Single performance tuning criteria	MCC, Accuracy, sensitivity, specificity, RMSE and AUC are reported with std.dev. But model selection based on MCC					
Or, Dataset dependent performance	TBD	TBD	TBD	TBD	TBD	TBD
Part E.6: Modeling Procedure and Model Tuning by Method						
	FOUR different methods will be applied to each dataset to enable systematic comparison:					
Classification procedure flow and logic	LibSVM (linear and RBF)	SMO (linear)	Naieve Bayes	Voted Perceptron		
Method Parameter(s)	Stratified 5f CV with 10 iterations, c=10 and gamma=0.01					
Number of Features						
Level of cross-validation						
Process	See figure 1					
Part F: Confirmatory Blinded Test Procedure (Appendix A contains information germane to this part)						
Number samples from original data source held out for blinded, confirmatory test	40	201	204	About 100	214	200 to 300
Prospective dataset number 1						
Prospective dataset number 2						
Endpoints to be predicted	LT and NLT	Liver cancer and non-liver cancer	Overall necrosis score	Tr. response and ER	TBD	TBD
Procedure	Using the best model from the corresponding dataset					
Batch effect treatment	Yes	No	No	No	No	No
Prediction performance criteria	TBD	TBD	TBD	TBD	TBD	TBD

Footnotes:

1 - Used array data distributed after QC by MAQC QC subgroups, whereby suspect arrays were removed by a consensus method.

2 - Exploratory analysis revealed a highly relevant batch effect that dictated setting aside a portion of the training samples as a testing set. Specifically, training samples were divided between arrays from a 2005 batch and a 2006, with 2006 arrays used for training, and 2005 arrays used for external testing.

3 – The data exhibited a strong time temporal dependency, with distinct separation into three batches

4 - Exploratory analyses indicated that the batch effect was too small to affect modeling and predictions (results not shown)

5 - LibSVM: Library of SVM developed by Chung Chang and Jen Lin, both Linear and RBF kernel type with c=10 and gamma = 0.01

6 - SVM-linear: This method involves the construction of binary SVM classifiers for all pairs of classes;

7 - Class for a Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data.

8 - A variant of perceptron algorithm. Implementation of the voted perceptron algorithm by Freund and Schapire. Globally replaces all missing values, and transforms nominal attributes into binary ones.

11 - Evaluates the worth of an attribute by measuring the gain ratio with respect to the class. Ranks attributes by their individual evaluations.

APPENDIX - E: Summary of the candidate models for 13 endpoints and the gene lists used for each model

End point A:

Steps	Description	Coefficients (If applicable)
Serial #	1	
Model ID	USM_Hamner_A_1	
Normalization	MAS5	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	20 in final model	
Classifier	LibSVM-RBF	
Software Packages used	Weka, R	
Gene list and coefficients in model	1426280_at 1419476_at 1450251_a_at 1418668_at 1437580_s_at 1423410_at 1435647_at 1420683_at 1456823_at 1440314_at 1449555_a_at 1455048_at 1420723_at 1422531_at 1425767_a_at 1452804_at 1455760_at 1460012_at 1435323_a_at 1420377_at	

End point B:

Steps	Description	Coefficients (If applicable)
Serial #	6	
Model ID	USM_Iconix_B_1	
Normalization	MAS5	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	30 in final model	
Classifier	LibSVM-LIN	
Software Packages used	Weka, R	
Gene list and coefficients in model	NM_013200_Probe1 NM_019157_Probe1 AI715955_Probe1 BF387347_Probe1 X92495_Probe1 AW914013_Probe1 AF031879_Probe1 AW914913_Probe1 U37058_Probe1 BE108246_Probe1 M26199_Probe1 AW525290_Probe1 AW524548_Probe1 AW535381_Probe1 X61925_Probe1 AW533257_Probe1 BF286131_Probe1 AW529672_Probe1 AW525189_Probe1 BE109912_Probe1 BF404878_Probe1 BF401593_Probe1 D86345_Probe1 BF405177_Probe1 D12498_Probe1 BE118122_Probe1 AW531250_Probe1 AW525089_Probe1 AF024622_Probe1 AI113076_Probe1	

End point C:

Steps	Description	Coefficients (If applicable)
Serial #	11	
Model ID	USM_NIEHS_C_1	
Normalization	MAS5	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	20 in final model	
Classifier	LibSVM-RBF	
Software Packages used	Weka, R	
Gene list and coefficients in model	1370902_at 1370832_at 1371785_at 1371400_at 1371412_a_at 1370355_at 1370080_at 1370150_a_at 1370725_a_at 1370583_s_at 1370670_at 1374610_at 1374591_at 1375170_at 1374625_at 1374765_at 1373778_at 1372510_at 1372729_at 1374529_at	

End point D:

Steps	Description	Coefficients (If applicable)
Serial #	16	
Model ID	USM_BR_D_1	
Normalization	MAS5	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	50 in final model	
Classifier	NB	
Software Packages used	Weka, R	
Gene list and coefficients in model	208712_at 218236_s_at 204623_at 208711_s_at 201508_at 205225_at 216092_s_at 215867_x_at 204667_at 212190_at 218807_at 212956_at 212444_at 208103_s_at 214164_x_at 213134_x_at 209289_at 204825_at 210735_s_at 202088_at 205066_s_at 205347_s_at 203963_at 202870_s_at 203789_s_at 209366_x_at 209173_at 209604_s_at 202089_s_at 212960_at 201030_x_at 218211_s_at 203108_at 215726_s_at 213032_at 209290_s_at 213564_x_at 217762_s_at 209773_s_at 203476_at 204822_at 209459_s_at 211864_s_at 207843_x_at 218806_s_at 205548_s_at 217838_s_at 210652_s_at 212195_at 208682_s_at	

End point E:

Steps	Description	Coefficients (If applicable)
Serial #	21	
Model ID	USM_BR_E_1	
Normalization	MAS5	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	30 in final model	
Classifier	SMO	
Software Packages used	Weka, R	
Gene list and coefficients in model	205225_at 209602_s_at 203963_at 214164_x_at 215867_x_at 217838_s_at 212960_at 214440_at 204623_at 209173_at 209696_at 218195_at 212956_at 209604_s_at 214404_x_at 205066_s_at 221765_at 202089_s_at 212771_at 210735_s_at 221016_s_at 203749_s_at 212148_at 212190_at 218807_at 212209_at 212492_s_at 201508_at 220192_x_at 209289_at	

End point F:

Steps	Description	Coefficients (If applicable)
Serial #	26	
Model ID	USM_MM_F_1	
Normalization	RMA	
Pre-Filtering of genes	FoldChange + P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	40 in final model	
Classifier	NB	
Software Packages used	Weka, R	
Gene list and coefficients in model	236558_at 1555878_at 1554899_s_at 209945_s_at 225917_at 223625_at 202416_at 213194_at 201602_s_at 211908_x_at 205529_s_at 216956_s_at 204159_at 211641_x_at 242104_at 227751_at 218859_s_at 218187_s_at 211650_x_at 211576_s_at 209098_s_at 1569454_a_at 201614_s_at 228324_at 213320_at 201558_at 215982_s_at 218701_at 218984_at 204204_at 228955_at 224523_s_at 202107_s_at 212022_s_at 211973_at 212021_s_at 211944_at 211963_s_at 211979_at 211990_at	

End point G:

Steps	Description	Coefficients (If applicable)
Serial #	31	
Model ID	USM_MM_G_1	
Normalization	RMA	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	24 in final model	
Classifier	LibSVM-LIN	
Software Packages used	Weka, R	
Gene list and coefficients in model	217934_x_at 223506_at 209206_at 210205_at 210178_x_at 210244_at 210220_at 210231_x_at 210057_at 210052_s_at 200602_at 204379_s_at 209053_s_at 209374_s_at 211645_x_at 214768_x_at 214777_at 215176_x_at 216207_x_at 216401_x_at 216576_x_at 217378_x_at 222777_s_at 234764_x_at	

End point H:

Steps	Description	Coefficients (If applicable)
Serial #	36	
Model ID	USM_MM_H_1	
Normalization	RMA	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	21 in final model	
Classifier	LibSVM-RBF	
Software Packages used	Weka, R	
Gene list and coefficients in model	201909_at 204409_s_at 204410_at 205000_at 205001_s_at 206624_at 206700_s_at 209031_at 214131_at 214218_s_at 221728_x_at 223645_s_at 223646_s_at 224588_at 224589_at 224590_at 227671_at 228492_at 230760_at 232618_at 236694_at	

End point I:

Steps	Description	Coefficients (If applicable)
Serial #	41	
Model ID	USM_MM_I_1	
Normalization	RMA	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	40 in final model	
Classifier	LibSVM-LIN	
Software Packages used	Weka, R	
Gene list and coefficients in model	211302_s_at 212063_at 211026_s_at 212076_at 211084_x_at 212221_x_at 211919_s_at 212223_at 211962_s_at 212209_at 211473_s_at 212220_at 211505_s_at 209318_x_at 210568_s_at 209279_s_at 210756_s_at 209309_at 210479_s_at 209498_at 210538_s_at 209512_at 210807_s_at 209427_at 210986_s_at 209456_s_at 210785_s_at 208657_s_at 210788_s_at 208890_s_at 212338_at 208373_s_at 212233_at 212334_at 212415_at 212568_s_at 212392_s_at 212409_s_at 212085_at 212090_at	

End point J:

Steps	Description	Coefficients (If applicable)
Serial #	46	
Model ID	USM_NB_J_1	
Normalization	MAS5	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	20 in final model	
Classifier	LibSVM-LIN	
Software Packages used	Weka, R	
Gene list and coefficients in model	A_23_P74349 A_23_P401 A_23_P44155 A_23_P145529 A_32_P159234 A_32_P151800 A_32_P143245 A_32_P44831 A_32_P77989 A_23_P335329 A_24_P96780 A_24_P57047 A_23_P10385 A_23_P51085 A_23_P17575 A_23_P163306 Hs23960.1 Hs143769.1 A_23_P102331 A_23_P386	

End point K:

Steps	Description	Coefficients (If applicable)
Serial #	51	
Model ID	USM_NB_K_1	
Normalization	MAS5	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	30 in final model	
Classifier	NB	
Software Packages used	Weka, R	
Gene list and coefficients in model	A_23_P149668 A_32_P4981 A_23_P501831 A_32_P47538 A_32_P190303 A_32_P4985 A_32_P134756 A_23_P48669 A_23_P396765 A_24_P88696 A_24_P297539 A_23_P133123 Hs75426.3 A_23_P323751 A_23_P2543 A_23_P155765 Hs87507.1 A_23_P125680 A_23_P96325 A_23_P254733 A_23_P138507 A_23_P100711 A_24_P98021 A_32_P171043 A_23_P65757 A_24_P902509 A_23_P23303 A_23_P157027 A_23_P115872 A_32_P30874	

End point L:

Steps	Description	Coefficients (If applicable)
Serial #	63	
Model ID	USM_NB_L_1	
Normalization	MAS5	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	20 in final model	
Classifier	SMO	
Software Packages used	Weka, R	
Gene list and coefficients in model	A_23_P259314 A_24_P500584 A_23_P137238 Hs456200.1 A_23_P309224 A_23_P429950 A_32_P212471 A_23_P315345 A_23_P125519 A_23_P162766 A_24_P186030 A_32_P183001 A_23_P156970 A_24_P134653 A_24_P237389 A_23_P146997 A_23_P217409 A_23_P148629 A_23_P93009 A_23_P136870	

End point M:

Steps	Description	Coefficients (If applicable)
Serial #	68	
Model ID	USM_NB_M_1	
Normalization	MAS5	
Pre-Filtering of genes	FoldChange+P-value	
Feature Selection Step 1	Gain Ratio with Cross validation approach	
Number of Features	20 in final model	
Classifier	NB	
Software Packages used	Weka, R	
Gene list and coefficients in model	Hs32976.1 A_32_P83570 A_23_P251151 Hs301404.34 A_23_P35277 A_24_P37540 A_23_P316012 A_23_P151895 Hs284281.1 A_24_P184931 A_24_P260443 A_32_P97169 A_23_P214897 A_24_P63290 A_24_P389251 A_32_P196837 A_23_P45536 A_24_P372833 A_23_P132718 A_23_P257649	

APPENDIX - F: MAQC-II Participating Data Analysis Groups (DAGs)

Leming.Shi@fda.hhs.gov Tel: +1-870-543-7387

December 6, 2007

DAT#	Org. Code	DAP File(s)	Organization
1	CBC	CBC DAP - Liang Zhang.doc	CapitalBio Corporation, China
2	CAS[1]	CAS DAP - Tieliu Shi I.doc	Chinese Academy of Sciences, China
4	CDRH	CDRH DAP - Gene Pennello.doc	Center for Devices and Radiological Health, FDA
5	CDRH2	Wagner et al. Plan for MAQC2.doc	Center for Devices and Radiological Health, FDA
6	CIPF	CIPF DAP - Joaquin Dopazo.doc	Centro de Investigacion Principe Felipe, Spain
7	Cornell	ICB DAP baseline - Fabien Campagne.doc	Weill Medical College of Cornell University, Institute for Computational Biomedicine (ICB)
8	DKFZ	DKFZ DAP - Benedikt Brors.doc	German Cancer Research Center, Germany
11	EPA	EPA DAP - Richard Judson.doc	U.S. Environmental Protection Agency
13	GeneGO	GeneGo DAP - Weiwei Shi.doc	GeneGo Inc.
14	FBK	FBK-MPBA DAP - Cesare Furlanello.doc	Fondazione Bruno Kessler, Italy
16	Ligand	Ligand Pharma DAP - Wen Luo.doc	Ligand Pharmaceuticals Inc.
17	NCTR	NCTR DAP - Weida Tong.doc	National Center for Toxicological Research, FDA
18	NIEHS[2]	NIEHS Chou Bushel DAP - Pierre Bushel I.doc NIEHS Li Bushel DAP - Pierre Bushel II.doc	National Institute of Environmental Health Sciences, NIH
18'	NIEHS[2]	NIEHS DAP - Jennifer Fostel.doc	National Institute of Environmental Health Sciences, NIH
19	NWU	NWU DAP - Simon Lin.doc	Northwestern University
20	Princeton	Princeton DAP - Jianqing Fan.doc	Princeton University
21	Roche	Roche DAP - Mark Fielden.doc	Roche Palo Alto LLC
22	SAI	SAI_DAP - John Zhang.pdf	Systems Analytics Inc.
23	SAS	SAS DAP - Russ Wolfinger.doc	SAS Institute Inc.
24	SIB	SIB DAP - Vlad Popovici.doc	Swiss Institute of Bioinformatics, Switzerland
25	Spheromics	Trygg Bylesjo Scherer DAP - Andreas Scherer.doc	Spheromics, Finland
26	SuperArray	SA DAP - Guozhen Liu.doc	SuperArray Bioscience Corporation
27	Tsinghua	Tsinghua University DAP - Xuegong Zhang.doc	Tsinghua University, China
29	Cedars-UCLA	Cedars-Sinai UCLA DAP - Xutao Deng.doc	University of California at Los Angeles Cedars-Sinai Medical Center
30	UIUC	UIUC DAP - SHeng Zhong.doc	University of Illinois at Urbana-Champaign
31	UML	UML DAP - Dalila Megherbi.doc	University of Massachusetts Lowell
32	USM	USM DAP - Venkata Thodima.doc	University of Southern Mississippi
35	Almac	Almac DAP - Juergen Frese.doc	Almac Diagnostics, UK
36	ZJU	ZJU DAP - Xiaohui Fan.doc	Zhejiang University, China
37	JHSPH	JHSPH DAP - Rafael Irizarry.doc	Johns Hopkins Bloomberg School of Public Health

[1]Three DAPs were submitted from the same Data Analysis Team (DAT) from the CAS; [2]Two DAPs were submitted from the same NIEHS DAT. 'There are two independent DATs from the NIEHS.

REFERENCES

Affymetrix. 2002 Affymetrix Microarray Suite User Guide, Version 5 edn. *Affymetrix* Santa Clara, CA.

Aizerman, M. A., Braverman, E. M., & Rozonoer, L. I. 1964. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, **25**, 821–837.

Alizadeh, A.A., M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu et al. 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* **403:** 503-511.

Alon, U., N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci U S A* **96:** 6745-6750.

Bair, E. and R. Tibshirani. 2004. Semi-supervised methods to predict patient survival from gene expression data. *PLoS Biol* **2:** E108.

Baldi, P., S. Brunak, Y. Chauvin, C.A. Andersen, and H. Nielsen. 2000. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* **16:** 412-424.

Beer, D.G., S.L. Kardia, C.C. Huang, T.J. Giordano, A.M. Levin, D.E. Misek, L. Lin, G. Chen, T.G. Gharib, D.G. Thomas et al. 2002. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat Med* **8:** 816-824.

Bhattacharjee, A., W.G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette et al. 2001. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc Natl Acad Sci U S A* **98:** 13790-13795.

Bolstad, B.M., R.A. Irizarry, M. Astrand, and T.P. Speed. 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19:** 185-193.

Brown, M.P., W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares, Jr., and D. Haussler. 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc Natl Acad Sci U S A* **97:** 262-267.

Butte, A. 2002. The use and analysis of microarray data. *Nat Rev Drug Discov* **1:** 951-960.

Burges C.J.C. 1998. A tutorial on support vector machines for pattern recognition *DataMining and Knowledge Discovery*, **2**(2):121–167.

Canales, R.D., Y. Luo, J.C. Willey, B. Austermiller, C.C. Barbacioru, C. Boysen, K. Hunkapiller, R.V. Jensen, C.R. Knight, K.Y. Lee et al. 2006. Evaluation of DNA microarray results with quantitative gene expression platforms. *Nat Biotechnol* **24:** 1115-1122.

DeRisi, J.L., V.R. Iyer, and P.O. Brown. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* **278:** 680-686.

Dudoit, S. and J. Fridlyand. 2002. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biol* **3:** RESEARCH0036.

Ein-Dor, L., O. Zuk, and E. Domany. 2006. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc Natl Acad Sci U S A* **103:** 5923-5928.

Eisen, M.B., P.T. Spellman, P.O. Brown, and D. Botstein. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A* **95:** 14863-14868.

Eisenstein, M. 2006. Microarrays: quality control. *Nature* **442:** 1067-1070.

Fielden, M.R., R. Brennan, and J. Gollub. 2007. A gene expression biomarker provides early prediction and mechanistic assessment of hepatic tumor induction by nongenotoxic chemicals. *Toxicol Sci* **99:** 90-100.

Fayyad U. and K. Irani. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1029.

Freund .Y., Schapire . R., 1999. Large margin classification using perceptron algorithm. *Machine Learning* **37**(3), 277-296.

Furey, T.S., N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, and D. Haussler. 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16:** 906-914.

Golub, T.R., D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri et al. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286:** 531-537.

Guo, L., E.K. Lobenhofer, C. Wang, R. Shippy, S.C. Harris, L. Zhang, N. Mei, T. Chen, D. Herman, F.M. Goodsaid et al. 2006. Rat toxicogenomic study reveals analytical consistency across microarray platforms. *Nat Biotechnol* **24:** 1162-1169.

Han, E.S., Y. Wu, R. McCarter, J.F. Nelson, A. Richardson, and S.G. Hilsenbeck. 2004. Reproducibility, sources of variability, pooling, and sample size: important considerations for the design of high-density oligonucleotide array experiments. *J Gerontol A Biol Sci Med Sci* **59:** 306-315.

Hall, M.A. 1998. Correlation-based feature selection for machine learning. *PhD thesis* Department of Computer Science, University of Waikato, Hamilto, New Zealand.

Hall M.A.and G. Holmes. May/June 2003. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transaction on Knowledge and Data Engineering*, **15**(3).

Harrington, C.A., C. Rosenow, and J. Retief. 2000. Monitoring gene expression using DNA microarrays. *Curr Opin Microbiol* **3:** 285-291.

Herrero, J., R. Diaz-Uriarte, and J. Dopazo. 2003. Gene expression data preprocessing. *Bioinformatics* **19:** 655-656.

Helmbold, D. P., & Warmuth, M. K. 1995. On weak learning. *Journal of Computer and System Sciences*, **50**, 551–573.

Hess, K.R., K. Anderson, W.F. Symmans, V. Valero, N. Ibrahim, J.A. Mejia, D. Booser, R.L. Theriault, A.U. Buzdar, P.J. Dempsey et al. 2006. Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *J Clin Oncol* **24:** 4236-4244.

Huiqing, L. 2004. Effective use of data mining technologies on biological and clinical data. *PhD thesis*, National University of Singapore, Singapore

Irizarry, R.A., Z. Wu, and H.A. Jaffee. 2006. Comparison of Affymetrix GeneChip expression measures. *Bioinformatics* **22:** 789-794.

Jaynes. ET. 2003. *Probability Theory: The Logic of Science* Cambridge University Press, ISBN 0-521-59271-2.

Johnson, W.E., C. Li, and A. Rabinovic. 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8:** 118-127.

Khan, J., J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson et al. 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat Med* **7:** 673-679.

Kira K. and L. Rendell. 1992. A practical approach to feature selection. Proceedings of the International Conference on Machine Learning. Morgal Kaufmann. pages 249-256.

Kohavi R. and G.H. John.1997.Wrappers for feature subset selection. *Artificial Intelligence*, **97**:273–324.

Li, B.Y., S. Kasemsumran, Y. Hu, Y.Z. Liang, and Y. Ozaki. 2007. Comparison of performance of partial least squares regression, secured principal component regression, and modified secured principal component regression for determination of human serum albumin, gamma-globulin and glucose in buffer solutions and in vivo blood glucose quantification by near-infrared spectroscopy. *Anal Bioanal Chem* **387:** 603-611.

Li, C. and W.H. Wong. 2001. Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc Natl Acad Sci U S A* **98:** 31-36.

Liu H. and R. Setiono. November 1995. Chi2: Feature selection and discretization of numeric attributes. *Proceedings of the IEEE 7th International Conference on Tools with Artificial Intelligence*, pages 388–391.

Liu, H. and L. Wong. 2003. Data mining tools for biological sequences. *J Bioinform Comput Biol* **1:** 139-167.

Lobenhofer, E.K., G.A. Boorman, K.L. Phillips, A.N. Heinloth, D.E. Malarkey, P.E. Blackshear, C. Houle, and P. Hurban. 2006. Application of visualization tools to the analysis of histopathological data enhances biological insight and interpretation. *Toxicol Pathol* **34:** 921-928.

Lockhart, D.J., H. Dong, M.C. Byrne, M.T. Follettie, M.V. Gallo, M.S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton et al. 1996. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat Biotechnol* **14:** 1675-1680.

Marshall, E. 2004. Getting the noise out of gene arrays. *Science* **306:** 630-631.

Matthews, B.W. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta* **405:** 442-451.

Michiels, S., S. Koscielny, and C. Hill. 2005. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet* **365:** 488-492.

Miller, L.D., P.M. Long, L. Wong, S. Mukherjee, L.M. McShane, and E.T. Liu. 2002. Optimal gene expression analysis by microarrays. *Cancer Cell* **2:** 353-361.

Mitchell, T.M., J.G. Carbonell, and R.S.a. Michalski. 1986. *Machine learning : a guide to current research*. Kluwer Academic Publishers, Boston.

Nguyen, D.V. and D.M. Rocke. 2002. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* **18:** 39-50.

Oberthuer, A., F. Berthold, P. Warnat, B. Hero, Y. Kahlert, R. Spitz, K. Ernestus, R. Konig, S. Haas, R. Eils et al. 2006. Customized oligonucleotide microarray gene expression-based classification of neuroblastoma patients outperforms current clinical risk stratification. *J Clin Oncol* **24:** 5070-5078.

Patterson, J.C., 2nd, J. Holland, and R. Middleton. 2006. Neuropsychological performance, impulsivity, and comorbid psychiatric illness in patients with pathological gambling undergoing treatment at the CORE Inpatient Treatment Center. *South Med J* **99:** 36-43.

Perez, E.A., L. Pusztai, and M. Van de Vijver. 2004. Improving patient care through molecular diagnostics. *Semin Oncol* **31:** 14-20.

Petricoin, E.F., A.M. Ardekani, B.A. Hitt, P.J. Levine, V.A. Fusaro, S.M. Steinberg, G.B. Mills, C. Simone, D.A. Fishman, E.C. Kohn et al. 2002. Use of proteomic patterns in serum to identify ovarian cancer. *Lancet* **359:** 572-577.

Platt. J. 1998. Fast training of support vector machines using sequential minimal optimization. In B. Schlkopf, C. Burges, and Smola A., editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press.

Pusztai, L., F.W. Symmans, and G.N. Hortobagyi. 2005. Development of pharmacogenomic markers to select preoperative chemotherapy for breast cancer. *Breast Cancer* **12:** 73-85.

Quinlan J.R. 1986. Induction of decision trees. *Machine Learning*, **1**:81–106.

Ramaswamy, S., K.N. Ross, E.S. Lander, and T.R. Golub. 2003. A molecular signature of metastasis in primary solid tumors. *Nat Genet* **33:** 49-54.

Richard T. Cox. 2001. Algebra of Probable Inference, *The Johns Hopkins University Press.*

Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, **65**, 386–407. (Reprinted in Neurocomputing (MIT Press, 1988).

Rosenblatt, F. 1962. *Principles of Neurodynamics*. Spartan, New York.

Rosenwald, A., G. Wright, W.C. Chan, J.M. Connors, E. Campo, R.I. Fisher, R.D. Gascoyne, H.K. Muller-Hermelink, E.B. Smeland, J.M. Giltnane et al. 2002. The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *N Engl J Med* **346:** 1937-1947.

Santos E.M. and H.M. Gomes. August, 2002. A comparative study of polynomial kernel SVM applied to appearance-based object recognition. *International Workshop on*

Pattern Recognition with Support Vector Machines.

Schena, M., D. Shalon, R.W. Davis, and P.O. Brown. 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **270:** 467-470.

Schölkopf, B. and A.J. Smola. 2002. *Learning with kernels : support vector machines, regularization, optimization, and beyond.* MIT Press, Cambridge, Mass.

Shaughnessy, J.D., Jr., J. Haessler, F. van Rhee, E. Anaissie, M. Pineda-Roman, M. Cottler-Fox, K. Hollmig, M. Zangari, A. Mohiuddin, Y. Alsayed et al. 2007a. Testing standard and genetic parameters in 220 patients with multiple myeloma with complete data sets: superiority of molecular genetics. *Br J Haematol* **137:** 530-536.

Shaughnessy, J.D., Jr., F. Zhan, B.E. Burington, Y. Huang, S. Colla, I. Hanamura, J.P. Stewart, B. Kordsmeier, C. Randolph, D.R. Williams et al. 2007b. A validated gene expression model of high-risk multiple myeloma is defined by deregulated expression of genes mapping to chromosome 1. *Blood* **109:** 2276-2284.

Shi, L. L.H. Reid W.D. Jones R. Shippy J.A. Warrington S.C. Baker P.J. Collins F. de Longueville E.S. Kawasaki K.Y. Lee et al. 2006. The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nat Biotechnol* **24:** 1151-1161.

Shippy, R., S. Fulmer-Smentek, R.V. Jensen, W.D. Jones, P.K. Wolber, C.D. Johnson, P.S. Pine, C. Boysen, X. Guo, E. Chudin et al. 2006. Using RNA sample titrations to assess microarray platform performance and normalization techniques. *Nat Biotechnol* **24:** 1123-1131.

Simon, R. 2005. Roadmap for developing and validating therapeutically relevant genomic classifiers. *J Clin Oncol* **23:** 7332-7341.

Spellman, P.T., G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, and B. Futcher. 1998. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell* **9:** 3273-3297.

Statnikov, A., I. Tsamardinos, Y. Dosbayev, and C.F. Aliferis. 2005. GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *Int J Med Inform* **74:** 491-503.

Tamayo, P., D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, and T.R. Golub. 1999. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc Natl Acad Sci U S A* **96:** 2907-2912.

Tan, P.K., T.J. Downey, E.L. Spitznagel, Jr., P. Xu, D. Fu, D.S. Dimitrov, R.A. Lempicki, B.M. Raaka, and M.C. Cam. 2003. Evaluation of gene expression measurements from commercial microarray platforms. *Nucleic Acids Res* **31:** 5676-5684.

Thomas, R.S., T.M. O'Connell, L. Pluta, R.D. Wolfinger, L. Yang, and T.J. Page. 2007. A comparison of transcriptomic and metabonomic technologies for identifying biomarkers predictive of two-year rodent cancer bioassays. *Toxicol Sci* **96:** 40-46.

Tong, W., A.B. Lucas, R. Shippy, X. Fan, H. Fang, H. Hong, M.S. Orr, T.M. Chu, X. Guo, P.J. Collins et al. 2006. Evaluation of external RNA controls for the assessment of microarray performance. *Nat Biotechnol* **24:** 1132-1139.

Troyanskaya, O., M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R.B. Altman. 2001. Missing value estimation methods for DNA microarrays. *Bioinformatics* **17:** 520-525.

Vapnik V.N. 1995. The Natural of Statistical Learning Theory. Springer.

van 't Veer, L.J., H. Dai, M.J. van de Vijver, Y.D. He, A.A. Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen et al. 2002. Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415:** 530-536.

West, M., C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J.A. Olson, Jr., J.R. Marks, and J.R. Nevins. 2001. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc Natl Acad Sci U S A* **98:** 11462-11467.

Weston, J., F. Perez-Cruz, O. Bousquet, O. Chapelle, A. Elisseeff, and B. Scholkopf. 2003. Feature selection and transduction for prediction of molecular bioactivity for drug design. *Bioinformatics* **19:** 764-771.

Weston J., S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. 2001. Feature selection for SVMs. *Advances in Neural Information Processing Systems*, **13**:668–674.

Witten, I.H. and E. Frank. 2000. *Data mining : practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, San Francisco, CA.

Xing, E.P. and R.M. Karp. 2001. CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics* **17 Suppl 1:** S306-315.

Yeang, C.H., S. Ramaswamy, P. Tamayo, S. Mukherjee, R.M. Rifkin, M. Angelo, M. Reich, E. Lander, J. Mesirov, and T. Golub. 2001. Molecular classification of multiple tumor types. *Bioinformatics* **17 Suppl 1:** S316-322.

Yeoh, E.J., M.E. Ross, S.A. Shurtleff, W.K. Williams, D. Patel, R. Mahfouz, F.G. Behm, S.C. Raimondi, M.V. Relling, A. Patel et al. 2002. Classification, subtype discovery,

and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell* **1:** 133-143.

Zeng, F., R.H. Yap, and L. Wong. 2002. Using feature generation and feature selection for accurate prediction of translation initiation sites. *Genome Inform* **13:** 192-200.

Zien, A., G. Ratsch, S. Mika, B. Scholkopf, T. Lengauer, and K.R. Muller. 2000. Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics* **16:** 799-807.

Zweig, M.H. and G. Campbell. 1993. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. *Clin Chem* **39:** 561-577.