Large Scale Disease Prediction

by

Patrick R. Schmid

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2008

Author ....
Department of Electrical Engineering and Computer Science
February, 2008

Certified by. ....
Bonnie Berger
Professor
Thesis Supervisor

Accepted by ....
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Large Scale Disease Prediction

by

Patrick R. Schmid

Submitted to the Department of Electrical Engineering and Computer Science
on February, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

The objective of this thesis is to present the foundation of an automated large-scale disease prediction system. Unlike previous work that has typically focused on a small self-contained dataset, we explore the possibility of combining a large amount of heterogenous data to perform gene selection and phenotype classification. First, a subset of publicly available microarray datasets was downloaded from the NCBI Gene Expression Omnibus (GEO) [18, 5]. This data was then automatically tagged with Unified Medical Language System (UMLS) concepts [7]. Using the UMLS tags, datasets related to several phenotypes were obtained and gene selection was performed on the expression values of this tagged micrarray data. Using the tagged datasets and the list of genes selected in the previous step, classifiers that can predict whether or not a new sample is also associated with a given UMLS concept based solely on the expression data were created. The results from this work show that it is possible to combine a large heterogenous set of microarray datasets for both gene selection and phenotype classification, and thus lays the foundation for the possibility ofautomatic classification of disease types based on gene expression data in a clinical setting.

Thesis Supervisor: Bonnie Berger
Title: Professor

Acknowledgments

I would like to thank Prof. Bonnie Berger and Dr. Isaac Kohane for the help, guidance, and support in performing this research. In addition, I would also like to thank Nathan Palmer for all his insight and time he spent on helping me with challenges that were faced during this work. Finally, I would like to thank Candice Chow for taking the time to proof read this document, and for providing me with helpful solutions to the complex problem known as the English language.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Imagine going to a doctor's office and having the physcian tell you your likelihood of having a certain disease. Imagine a surgeon being able to conclusively determine the origin of a tumor so that the correct treatment can be implemented. This will become the norm in the near future. In order for this to become a reality, however, a vast amount of data needs to be leveraged and combined to produce accurate predictors for the wide array of clinical outcomes. The aim of the work in this thesis is to provide the groundwork to show the feasibility of such an automated disease prediction system.

1.2 Motivation

With ever-growing repositories of public microarray data, the notion of using multiple different datasets spanning various diseases, treatment conditions, and tissue types to create a classification system becomes feasible. Similarly, with the constant decrease in price and complexity of performing microarray experiments, the clinical application of microarrays is within reach. Unfortunately, without a so-called "black box" that a clinician can use to test a given patient's gene expression data against a multitude of diseases, gene expression data cannot be used as a diagnostic tool.

Recent work utilizing large disparate datasets by Butte et al. [9] and Segal et al. [41] show that finding statistically significant genes and gene modules is indeed possible. What they failed to show, however, is that this same data can then be used to generate useful classifiers for these different conditions. That is, they only showed that the phenotypic data can be used to find significant genes or gene modules, but not that these same genes can then be used to classify new instances of data. To the best of our knowledge, all phenotype classification methods thus far have used a single dataset, a strictly standardized set of data, or a computationally intensive normalization method on the input data. For example, Furey et al. [23] performed their SVM classification independently on three different homogeneous microarray datasets. Similarly, Anderson et al. used a nearest neighbor algorithm to predict childhood leukemia [1]. While these disease specific methods are vital in furthering the understanding of individual diseases, for diagnostic purposes it will become necessary to use a large quantity of heterogenous data. Furthermore, complex normalization methods, such as Loess normalization [19], cannot be used as they require the normalization to be performed on *all* of the data. Not only is this impractical when analyzing thousands of experiments, but one would also have to re-normalize the data each time a new experiment is added to a database. The only example of using multiple datasets, albeit on a small scale, without a complex normalization method that we are aware of, was proposed by Warnat et al. [45]. They show that by either performing quantile discretization or by converting the expression values to median rank scores, accurate classification results are plausible. As they themselves mentioned, however, since only two datasets were used for each of the three phenotypes they examined, their results required further study.

In this work, we show that the use of a large heterogenous database as the basis of phenotype classification is not only feasible, but it also gives promising results and can be used as the foundation of a large-scale phenotype prediction tool. We first employed a natural language processing tool to annotate free text with domain specific concepts and then used these concepts in conjunction with regular expressions to automatically select datasets associated with phenotypes of interest. Using these datasets, we performed gene selection and then phenotype classification. Our results show that it is possible to integrate a large quantity of heterogeneous microarray datasets, and the results are comparable to other methods that used homogeneous input data.

1.2.1 Terminology

Throughout this work the following definitions will be used unless explicitly stated otherwise. A microarray *dataset* will be a set of microarray *experiments* that were conducted by a specific lab for a specific purpose. For example, if a group of scientists were studying lung cancer and performed ten microarray experiments, five disease state experiments and five control experiments, then the set of these ten experiments is a dataset. Each experiment will also have associated with it a *platform*. The platform is the the actual chip that the microarray experiment was conducted on, for example the Affymetrix HGU-133A chip. Figure 1-1 shows a pictorial representation.

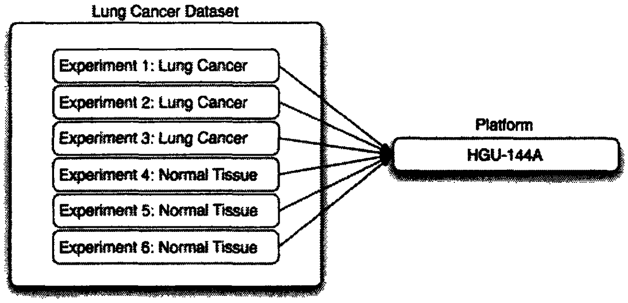

Lung Cancer Dataset
Experiment 1: Lung Cancer
Experiment 2: Lung Cancer
Experiment 3: Lung Cancer
Experiment 4: Normal Tissue
Experiment 5: Normal Tissue
Experiment 6: Normal Tissue
Platform
HGU-144A

Figure 1-1: The relationship of a *dataset*, an *experiment*, and a *platform*.

1.2.2 Microarrays

The term *microarray* has been previously used but never clearly defined. At the most basic level, microarrays are used to perform high throughput biological gene expression experiments. In essence, a microarray experiment simultaneously measures the quantity of thousands of genes in a sample. For most common microarrays, a scientist starts by extracting mRNA from a tissue or system of interest and creates a fluorescence-tagged cDNA copy of this mRNA. These *sample probes* are then hybridized to a microarray chip that have cDNA *probes* attached to the surface in a predetermined grid pattern. The underlying idea behind this process is that a sample probe will only bind to its complementary probe, thus allowing a scientist to measure the quantity of the sample probe present. After leaving the microarray chip submerged in the solution containing the sample probes for several hours, any excess unhybridized sample probes are washed off. The microarray is then scanned using laser light and a digital image scanner records the brightness level at each probe location. It has been shown that the brightness at a particular spot is correlated with the RNA level in the original tissue or system of interest [39].

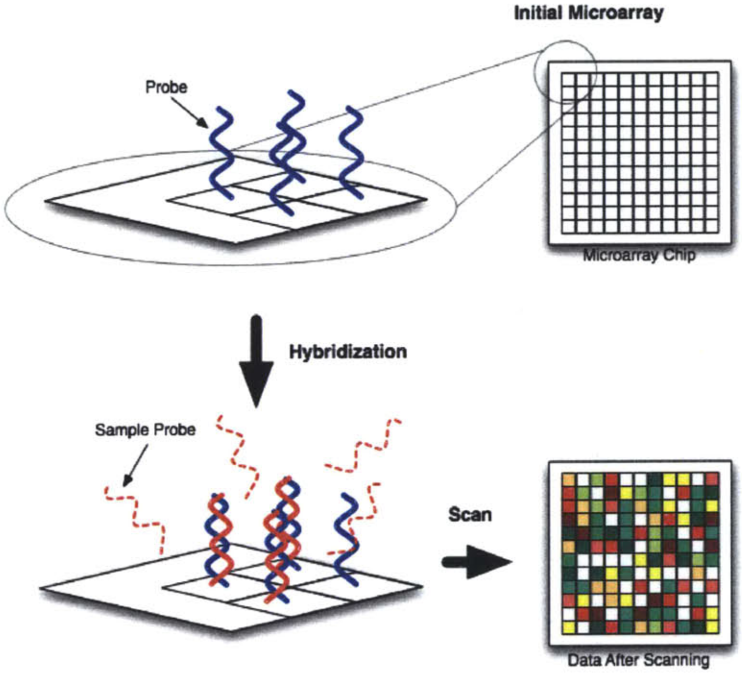


Figure 1-2: The basics of microarray technology. Fluorescence-tagged cDNA *sample probes* for a tissue or system of interest are hybridized to a microarray chip containing cDNA *probes*. After the hybridization process, the chip is scanned using a laser, and the intensity levels at each probe location are measured to determine the expression level for a particular gene.

There are multiple different forms of microarray technologies, the two major ones being *spotted cDNA arrays* and *oligonucliotide arrays*. While both of them measure gene intensity levels, the approach of how they are created and the way in which the intensities are measured differ. The former was introduced by Mark Shena et al.

[39] in 1995 and is also known as a cDNA microarray. Typically, a robotic spotter picks up cDNA that has been amplified using PCR and places it on a glass slide. When performing the experiment, two conditions are actually tested simultaneously, each with a different fluorescent color. The intensity levels are then measured as a ratio of the two conditions. On the other hand, oligonucleotide arrays are generated by a photolithographic masking technique first described by Stephen Fodor et al. [22] and were made popular by Affymetrix. Unlike the cDNA arrays, oligonucleotide arrays only measure one condition at a time. One therefore needs to perform multiple experiments in order to compare multiple conditions. A more in-depth explanation about microarray technology and the various types of microarrays can be found in [28].

Difficulties in Dealing with Microarrays

Although microarray technology enables one to get a genome-wide snapshot of the quantity of RNA levels in a sample, there are many factors that make this data difficult to deal with. Simply put, the data is *noisy*. For example, a replicate experiment that uses exactly the same experimental setup can, and often does, report different expression levels. While this may seem disconcerting, this irreproducibility of data is not limited to microarray technology, but also occurs in most types of experiments in which miniscule quantities are measured with a highly sensitive device. The standard approach to dealing with this problem is to make many replicates and hope that the intensity values of the repeats converge to the true measure. Unfortunately, not only are microarray experiments very expensive, but these sort of repeats tend to eliminate noise caused by measurement errors and not the biological variation inherent in the samples being studied.

Another major obstacle in dealing with microarray technology is the lack of cross platform reproducibility. As detailed in [28], high intensity levels in a cDNA experiment did not correspond well with high levels in oligonucleotide experiments. In light of this, this work only uses single channel data. Furthermore Hwang et al. [26] performed a study where they compared two human muscle biopsy datasets that used two generations of the Affymetrix arrays (HG-U95Av2 and HG-U113A) and showed that they obtained differences in both cluster analysis and the differentially expressed genes. While this is an unfortunate conclusion, this sort of noise is inevitable and cannot be countered. Interestingly, however, our results show that while it may be true that the results from different datasets are not identical, reasonable results both for gene selection and phenotype classification can be attained.

1.2.3 Problem Statement

The aim of this thesis is to show the plausibility of using a large set of microarray experiments to automatically build classifiers for various diseases. Unlike previous work that has typically utilized a single or a few datasets to generate these classifiers, the objective is to show that a large heterogeneous database filled with microarray experiments can be used to generate classifiers.

1.3 Data

1.3.1 GEO Data

The Gene Expression Omnibus (GEO) [18, 5] is a public database containing gene expression and molecular abundance provided by the National Center for Biotechnology Information (NCBI). GEO data is divided into GEO Data Sets (GDS), GEO Series (GSE), GEO Samples (GSM), and GEO Platforms (GPL) files as depicted by Figure 1-3. In terms of the terminology introduced earlier, GDS and GSE files are datasets, GSM files are experiments, and GPL files are the platforms. The difference between a GDS and GSE file is that a GDS file contains additional meta information that the curators of GEO added to the original GSE file that was uploaded. For example, GDS files contain *subset* information about each experiment such that one can see what condition a given experiment has in the dataset. The dataset with the identifier GDS1, for instance, was an experiment conducted to find genes related to reproductive tissue in *Drosophila melanogaster* (data accessible at NCBI GEO database, accession number GDS1; `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GDS1`). The various subset information provided includes information such as gender of the fly for the given sample and the tissue the sample was created from. Another important difference between GDS and GSE files is that a GDS may only contain experiments that were conducted on a singled GPL platform. It is possible for a GSE to contain experiments with multiple platforms because there are instances when an experimenter compared multiple microarray platform technologies or performed a cross-species study. It is important to note that there are many more GSE files in GEO than GDS files, as there are many datasets which have yet to be manually annotated. Due to the large size of the GEO database, only a subset of the entire database was downloaded. More specifically, all GDS files pertaining to *Homo sapiens*, *Mus musculus*, and *Drosophila melanogaster* were downloaded on 13 June 2007. All GSE and GPL files relating to these GDS files were also obtained. This amounts to 1317 GDSs that were curated from 1086 GSEs, which are made up of 25128 experiments, and 1171 platforms. While the data for mouse and fly was downloaded, this study concentrated only on the microarrays pertaining to human data.

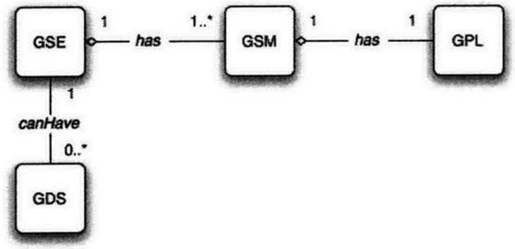


Figure 1-3: The relationship of GEO files as represented by a UML diagram.

Other Data

Other supporting materials were obtained directly from the National Center for Biotechnology Information (NCBI) website [33]. These files include mapping of gene symbol to Entreze identifier (gene_info), Refeq and GenBank identifiers to Entrez Gene identifiers (gene2accession), UniGene identifeir to Entrez Gene identifier (gene2unigene), and Enterz Gene identifiers to GO terms (gene2go). The Gene Ontology data was obtained directly from Gene Ontology website [4]. These files were all downloaded on 13 June 2007.

1.4 Code Developed

In order to perform this work, various libraries were implemented and used. A brief description of the major components are discussed below.

1.4.1 Majik: Microarray Java Interface Kit

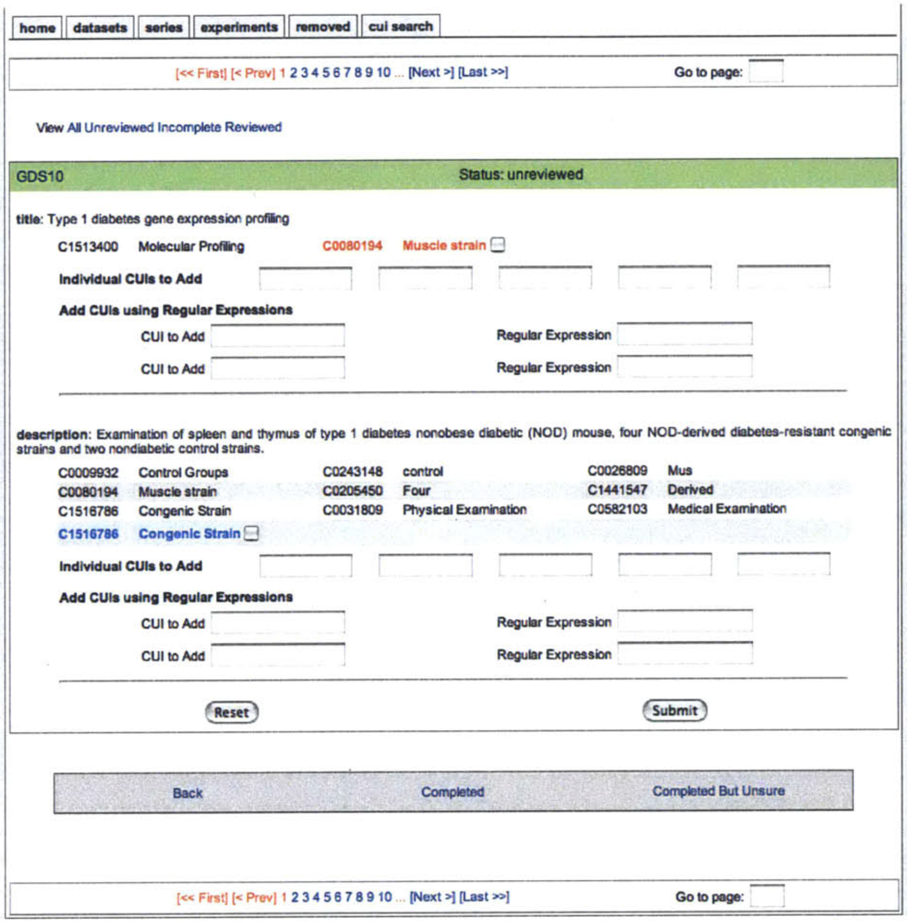
To deal with the vast amount of microarray data, the various file types, etc., a library to work with microarray data was developed. This library contains methods to read the various GEO files and to perform manipulations of the data. Most of the manipulation of GEO files is performed using this library.

1.4.2 JMath

There is no effective statistical package available for Java. Since the majority of this work was performed in Java it became necessary to create a library for statistical methods. Many of the methods were ported over from the R statistical package. JMath is thus an object-oriented framework for various statistical functions that were used throughout this work.

1.4.3 WekaX

The WEKA data mining toolkit was extended with various new functionality. For example, filters based on statistical methods (using the JMath library) and a kernel density estimate based classifier were implemented.


home
datasets
series
experiments
removed
cui search
[<< First] [< Prev] 1 2 3 4 5 6 7 8 9 10 ... [Next >] [Last >>]
Go to page:
View All Unreviewed Incomplete Reviewed

GDS10 Status: unreviewed

title: Type 1 diabetes gene expression profiling

C1513400 Molecular Profiling C0080194 Muscle strain ☐

Individual CUIs to Add □ □ □ □ □

Add CUIs using Regular Expressions

CUI to Add □ Regular Expression □

CUI to Add □ Regular Expression □

description: Examination of spleen and thymus of type 1 diabetes nonobese diabetic (NOD) mouse, four NOD-derived diabetes-resistant congenic strains and two nondiabetic control strains.

C0009932	Control Groups	C0243148	control	C0026809	Mus
C0080194	Muscle strain	C0205450	Four	C1441547	Derived
C1516786	Congenic Strain	C0031809	Physical Examination	C0582103	Medical Examination
C1516786	Congenic Strain ☐				

Individual CUIs to Add □ □ □ □ □

Add CUIs using Regular Expressions

CUI to Add □ Regular Expression □

CUI to Add □ Regular Expression □

Reset
Submit
Back
Completed
Completed But Unsure
[<< First] [< Prev] 1 2 3 4 5 6 7 8 9 10 ... [Next >] [Last >>]
Go to page:

Figure 1-4: Screenshot of the website used to remove incorrectly tagged text.

1.4.4 UMLS Annotation Correction Site

A web-based front end to the MySQL database containing all the UMLS tags for the microarray datasets, shown in figure 1-4, was created to enable users to efficiently remove incorrect labelings. Through the website, users were allowed to examine the tags associated with the text and supply regular expressions to remove falsely tagged pieces of text. Users were also allowed to insert new tags that were missed by the MMTx software.

Chapter 2

Methods

To perform the automatic classification of various phenotypes from disparate microarray datasets, the text of the microarray datasets was first automatically tagged with medical concepts. In addition, the expression intensity values were made comparable in a non-compute intensive preprocessing step. This was then followed by gene selection and finally by classification. During the gene selection phase, a total of six different variations were performed. The workflow is detailed below and is depicted in Figure 2-1.

2.1 Data

The Gene Expression Omnibus (GEO) [18, 5] is a public database provided by the National Center for Biotechnology Information (NCBI) that contains gene expression and molecular abundance data. All GEO Datasets (GDS) files pertaining to *Homo sapiens* were downloaded on 13 June 2007. All GES Series (GSE) and GEO Platform (GPL) files relating to these GDS files were also obtained. This amounts to 612 GDSs that were curated from 500 GSEs, which are made up of 14454 experiments and performed on 142 different platforms.

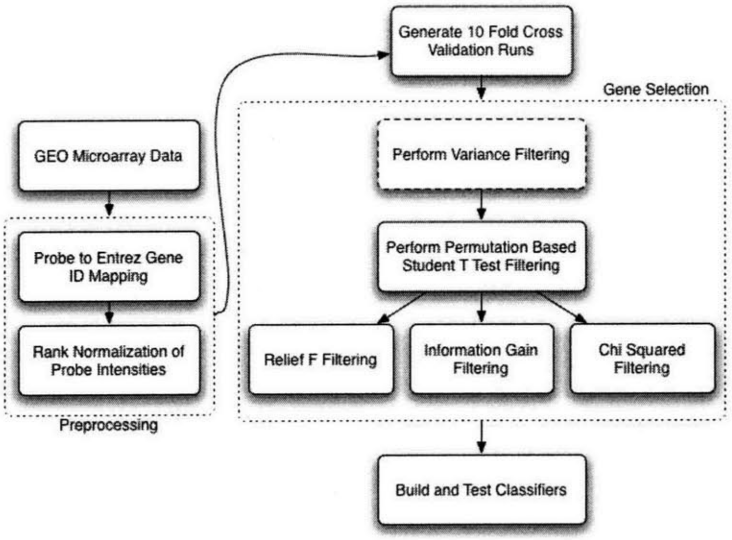

Generate 10 Fold Cross Validation Runs
Gene Selection
GEO Microarray Data
Perform Variance Filtering
Probe to Entrez Gene ID Mapping
Perform Permutation Based Student T Test Filtering
Rank Normalization of Probe Intensities
Relief F Filtering
Information Gain Filtering
Chi Squared Filtering
Preprocessing
Build and Test Classifiers

Figure 2-1: The processing workflow used to conduct this study. First, the GEO data is made comparable by mapping all the microarray probe identifiers to NCBI Entrez Gene identifiers. This comparable data is then rank normalized and used for gene selection and phenotype classification. A total of six different gene selection methods were performed, and the then multiple different classifiers were tested on the selected genes.

2.2 Automated Text Tagging

One of the largest drawbacks to the data in GEO is the lack of machine interpretability. While one can simply browse GEO using the web if only looking for a few datasets, this becomes cumbersome if attempting to obtain a large quantity of datasets. More importantly, browsing the web is not a viable solution for a phenotype predicting "black box." Thus, instead of using the online search tool, we downloaded the aforementioned human datasets. Along with the intensity data, all of the files in the database contain only text that was written by either the experimenter that performed the initial experiments or the curators that added meta data. This does not allow one to easily find, for example, all datasets pertaining to lung cancer without some search tool. The most naïve option would be to simply use regular expressions to search all of the text to find experiments of interest. Unfortunately this does not work well in a domain that has so many synonyms. For example, "breast cancer," "breast carcinoma," "carcinoma of the breast," "malignant breast carcinoma," and so forth, all refer to the same concept. To overcome this barrier, the MetaMap Transfer (MMTx) software [3] was used to annotate the text within the GDS and GSE files with Unified Medical Language System (UMLS) concepts [7]. A MySQL database was used to store the 11858 distinct concepts associated with the data. Thus, the problem of finding related datasets was turned into a problem of finding datasets with the same UMLS identifiers. As explained by Butte et al. [9], the annotation process is not perfect and thus we used regular expressions to remove many of the incorrect annotations. Annotations that were missed by the software were also added.

To perform phenotype classification, two distinct test groups of microarray datasets were created; one for disease specific phenotypes, and the other for tissue specific phenotypes. These groups of microarray datesets were generated by searching for the relevant UMLS concepts in the database. For example, one of the phenotypes in the diseased group was "breast cancer" and we searched for datasets that were related to the UMLS concept "Breast Carcinoma" (C0678222). The candidate list of datasets for each phenotype was then reduced to only those performed using an Affymetrix chip as it is hard to interpret two-channel data in conjunction with single-channel data. This produced an initial list of 122 candidate datasets for the seven disease phenotypes. We then further pruned this list to eliminate any datasets that made reference to treatment conditions, such as knock-outs and treatment therapies. As a matter of fact, simply using a rule to exclude any dataset with the term "response" in the title or description produced a nearly final list of datasets. We had to manually include only seven datasets that were removed by the previous rule as they contained some diseased controls along with the treatment data. We then also incorporated experiments from two large-scale analyses of the human transcriptome datasets (GDS596 and GDS181). We also had to remove eight datasets that did not have at least two diseased tissue samples.

Unfortunately, the efficacy of this process was drastically different for the disease and tissue specific data due to the nature of the microarray experiments in GEO. Most of the datasets pertaining to diseases contained only experiments pertaining to one disease, while the majority of the tissue datasets contained a vast number of different tissue types. Due to this difference, the automatic selection of datasets was only performed for the diseased group. The datasets for the tissue group were manually curated and the phenotypes of each experiment were supplied by hand.

With a set of datasets in hand, the next task was to group the individual experiments with those pertaining to the disease state of interest. All experiments in the disease data group that were not annotated as being diseased were removed. Similarly, all experiments in the tissue data group that were not annotated as being normal controls were discarded. This state was inferred from the `subset` information in the GDS files. Since the majority of the tissue specific datasets contained multiple normal tissue samples, it was not possible to automatically label each experiment with both its phenotype and disease state accurately. Unlike the tissue data, however, the datasets containing diseased samples are by and large specific to a single disease phenotype. We therefore used regular expressions to automatically assign a disease state label to each experiment. This process achieved a sensitivity of 62% and specificity of 98% on the giving data for inferring whether an experiment was diseased or not. To ensure the reproducibility of the subsequent gene selection and classification steps, the incorrect annotations were then modified manually to the correct labelings.

Following this process, the data for the disease group consisted of 40 distinct GDS files made up of 894 GSM files. Of those, only 782 were used, as those were the ones representing the disease state. These experiments are made up of samples of arthritis, breast cancer, leukemia, lung cancer, lymphoma, prostate cancer, and renal cancer. For tissue classification, 346 normal tissue state experiments of the 684 GSM experiments from 23 GDS files were used. Bone marrow, brain, heart, liver, lung, muscle, pancreatic, prostate, renal, spinal cord, and thymus tissues were represented in these experiments. Tables A.1 and A.2 contain the detailed listing of the different datasets.

In order to simulate miss-annotation of data, incorrect data was incorporated into some of the disease phenotype data. We introduced emphysema samples (GDS737) and experiments containing data for lung pneumocytes infected with Pseudomonas aeruginosa (GDS1022) into the lung cancer data. The prostate cancer data was augmented with two datasets containing treated prostate adenocarcinoma cell line data (GDS2057 and GDS2058). Expression data from peripheral blood mononuclear cells from patients with renal cancer following treatment with rapamycin analog CCI-779 (GDS1021) was added to the renal cancer data. Finally, some spondyloarthropathy samples were left in with the arthritis data from GDS711. The data for the disease group thus ultimately consisted of 45 distinct GDS files made up of 906 usable experiments.

2.3 Preprocessing

As each of the GDS files is already internally normalized as part of the uploading requirements to GEO, intra-dataset normalization was not necessary. Unfortunately the different datasets obtained from GEO are not directly comparable to each other. Each experiment is based on a specific platform and thus the intensity values provided are values for a specific probe on the microarray. Furthermore, the identifiers used are platform specific and cannot be directly translated between multiple platforms. To overcome this obstacle, each probe was mapped to its corresponding Entrez Gene identifier. Entrez Gene identifiers, which will be referred to as either the Entrez ID or gene ID from here on, were chosen due to the universality of the identifier. Unlike gene names or symbols, NCBI guarantees that a given gene ID will only be used for a single gene.

The following procedure, which is pictorially represented in figure 2-2, was used to perform the probe to gene ID mapping. First, regular expressions were used to attempt to find the gene symbol, GenBank identifier, RefSeq identifier, UniGene identifier and the Entrez identifier in the platform file. If an Entrez identifier was present, then this was immediately used and the subsequent steps were not performed. If either a gene symbol or a UniGene ID was present, those were used to find the Entrez identifier. If no Entrez IDs were found based on the gene symbol or UniGene ID, then the RefSeq and GenBank identifiers were mapped to the Entrez IDs. Unfortunately, this step does not guarantee a one-to-one mapping and thus the gene ID that occurred

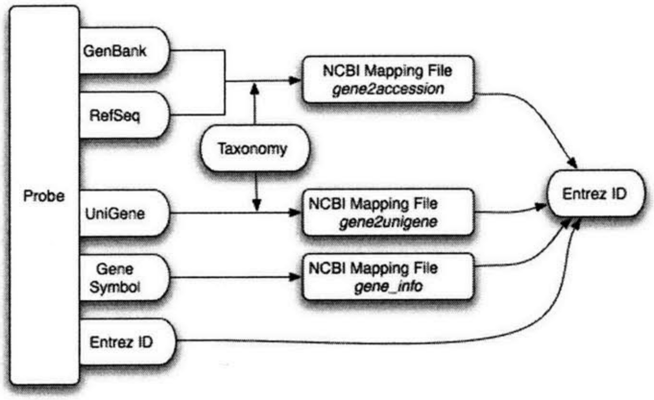

Probe
GenBank
RefSeq
UniGene
Gene Symbol
Entrez ID
Taxonomy
NCBI Mapping File gene2accession
NCBI Mapping File gene2unigene
NCBI Mapping File gene_info
Entrez ID

Figure 2-2: The process for mapping platform probe identifiers to Entrez Gene identifiers.

the most frequently was used. If the counts were tied, the top four Entrez IDs were used. Multiple Entrez IDs were used as opposed to the single best one due to the possibility of an erroneous match. It was assumed that by duplicating the information over multiple genes, at least one of them should be correct. A similar method was described by Chen et al. [11].

While mapping all probe identifiers to Entrez IDs allows for mapping probe values between different platforms, it does not address the issue of different normalization schemes in the various datasets. This was addressed by rank normalizing each probe intensity value as performed by Butte et al. [9].

We performed leave-dataset out cross-validation to perform the gene selection and classification. For each phenotype, the data was split into ten cross-validation runs such that the phenotype of interest consisted of one class and all remaining data from the other phenotypes made up the other. We also ensured that each cross-validation run held out at least one dataset with the phenotype of interest. If there were fewer than ten datasets assigned with a given phenotype, then datasets were reused for the positive class (those with the phenotype of interest) while ensuring that the negative set (those with a different phenotype) contained different datasets than in the previously generated cross-validation runs.

Unlike traditional cross-validation schemes that hold out individual experiments,

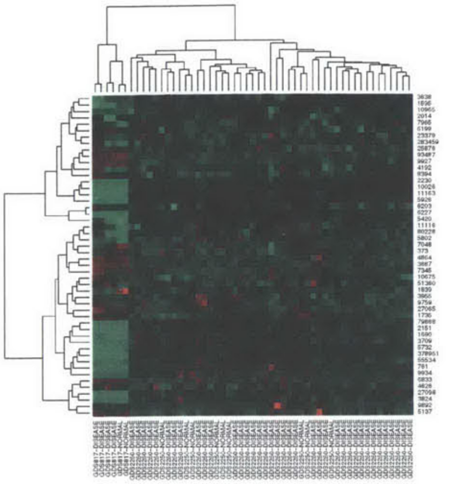


Figure 2-3: A heatmap generated by the heatmap() function in the R statistical computing package [36] of 50 random genes from two different breast cancer datasets. Due to differences in normalization, the experiments do not cluster by phenotype (diseased or normal) but rather by dataset.

when combining multiple datasets it was imperative to remove entire datasets. The reasoning behind this can be seen in figure 2-3 where 50 randomly selected genes from two breast cancer datasets, GDS2250 and GDS817, were clustered. One would hope that the two major clusters consisted of diseased and normal samples, but instead the experiments clustered by dataset rather than phenotype. As a matter of fact, Warnat et al. [45] report a similar result when clustering leukemia samples from two different datasets. Thus, if one had excluded individual experiments, then the likelihood would be much higher that a classifier would perform well due to it picking the dataset the experiment came from rather than the phenotype it belongs to. For example, assume that experiments from the aforementioned GDS2250 breast cancer dataset are used both in the training and testing set. Since this dataset has four times as many diseased samples as control samples, a majority of both the training and testing would consist of diseased tissue. As intra-dataset similarity is greater than phenotype similarity, a classifier would most likely classify a testing experiment from GDS2250 as coming from GDS2250. Considering that a majority of the GDS2250 experiments have the class label "diseased," this testing instance would probably also be labeled as such as well. If a classifier does so, it will automatically get 75% of the classifications correct since only about a quarter of the testing data from GDS2250 are normal control samples. As a matter of fact, we performed tests where experiments rather than entire datasets were left out, and the performance of the classifiers was better (results not shown).

2.4 Gene Selection

Utilizing a set of datasets with comparable probe identifiers and intensity values, six different gene selection methods were performed. This process can be described as passing the input data through a set of filters, each of which only keeps the best genes, until a final list of most significant genes is produced. The six selection methods performed are variations on which filters were used.

The first filter used was based on the difference of inter-class and intra-class variance. Due to the heterogeneity of the data there were values for a given gene that were both extremely high and low values within a single class. Furthermore, there were cases in which such genes were actually chosen as part of the set of significant genes. These genes, however, are probably not important marker genes, as for a given phenotype, either low or high values are possible. To remove these offending genes, the genes were ranked in decreasing order based on a score computed by the difference in intra-class to inter-class variance using the formula:

$$score = Var(G) - \sum_{c \in C} Var(G_c),$$

where G is the vector of all intensity values for a given gene, C is the set of classes, and G_c is the vector of intensity values for the given gene in class c. In other words, genes that displayed low intra-class variance but high inter-class variance are preferable. This filter was applied independently to all ten cross-validation runs after which each gene was scored by the sum of the ranks produced during each run. Using this method, the bottom 10% of the genes were removed.

A permutation based Student's T test filter was also used. If the F test returned a p-value of less that 0.05, Welch correction was used. An unpermuted t-test could not be used as it could not be guaranteed that the underlying distribution of intensities would be normal. More formally, for each gene, the data was split into the two classes. A t-test was then performed on 1000 random permutations of the class labels to produce a null distribution. Based on this null distribution a gene was considered to have a significant association with the phenotype if it had a multiple hypothesis corrected p-value of 0.01. After all ten cross-validation runs were complete, any gene that did not appear to be significant in at least eight of the ten runs was immediately discarded and not considered in any subsequent step.

Finally, three different "standard" machine learning feature selection methods were employed: relief F [27, 29, 38], information gain, and chi-squared. Briefly, the relief F algorithm selects genes by randomly sampling experiments, computing its nearest neighbors, and then adjusting the score based on how well the gene can discriminate experiments from neighbors of different classes. Information gain measures the difference between two probability distributions based on comparing the entropy of a given class to the entropy of the class given the gene. Chi-squared evaluates genes by the chi-squared statistic with respect to the class. Similar to the variance based filtering, the selection was performed independently on each of the ten cross-validation runs and genes that were selected multiple times were deemed more significant. Since the feature selection methods provide a score for each gene, individual genes in each cross-validation run were ranked according to its score as determined by the selection algorithm. To produce the list of the top ten significant genes, for instance, the ranks of all the genes were added and the ten genes with the highest cumulative rank were chosen.

One may notice that the cross-validation runs are being used at each step as opposed to over the entire selection (and later, classification) process. The reason behind this relates back to the way the cross-validation runs were initially generated and is based on the requirement of finding the most robust set of genes for a given phenotype. Since each cross-validation run uses a different subset of the data, requiring a gene to be significant in multiple cross-validation runs produces the most robust set of significant genes.

As aforementioned, six different combinations of these filters were tested. The first three were the permutation based t-test filter followed by one of the standard machine learning selection algorithms. The second three were the same as the first three except that the variance filter was used prior to the permutation based t-test filter.

2.5 Classification

With the set of significant genes at hand, several classifiers were compared using ten cross-validation runs for each phenotype. We examined kNN [2], C4.5 decision tree [34], random forest [8], SVM [10, 20], and K* [14] classifiers. Boosted versions of the kNN and decision trees were also tested. Not only were various classifier-specific parameters tested, the number of input genes was also altered between 10 and 300. These classifiers were chosen on the basis of past performance on microarray data [23, 32, 37, 15].

2.5.1 KDE: Kernel Density Estimate Based Classifier

In addition to these well known classifiers, a nearest neighbor based classifier using kernel density estimates [42, 40] was also developed and tested. A kernel density estimate can be thought of as a smoothed estimate of a distribution where the kernel describes the smoothing function. For example, 2-4 shows the density estimate of 1000 random normally distributed values. To train this classifier, a separate density estimate using a Gaussian kernel is generated for each class and for each gene. For example, to build a classifier based on ten genes and two classes, a total of 20 kernel density estimates are generated. Given a trained distribution for each class for a gene, one can classify a new value by computing the likelihood the value came from each of the trained distributions. To classify a new experiment, the classifier computes

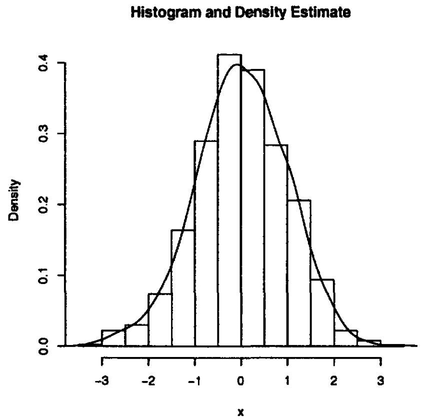


Figure 2-4: A histogram and density estimate of 1000 random normally distributed points. The density estimate is essentially a smoothed density estimate generated from discrete data points.

the probability for each gene's value using each of the trained distributions. The experiment is then classified by assigning it the class label that most of the individual genes' values belonged to.

A version of this classifier that assigns weights to each of the genes based on the separation of the density estimates was also created. While there are numerous ways in which one could potentially measure the difference between two distributions, we chose a method based on sampling. For each density estimate for a given gene the the following procedure was performed. First, n equally spaced points in the domain of the density estimate were chosen. The probability of each of the n points was then computed in all of the *other* density estimates and summed up. If the distributions are completely disjoint then all of these probabilities are zero. The weight for a gene is then inversely proportional to the sum such that the smaller the sum the higher the weight. The lowest weight for a gene was bounded by 1 / $(numGenes * 10)$ where *numGenes* is the number of genes used to build the classifier.

Chapter 3

Results & Discussion

3.1 Gene Selection

While the objective of this work was to show that it is possible to generate accurate classifiers from a heterogenous database, it is important to verify that the genes deemed significant during the selection process are indeed related to the phenotype in question. Since heterogenous data is being used, it is much more likely that a gene is selected as significant just because of the large inter-dataset differences mentioned earlier. In other words, a cancer classifier built from genes that are not related to cancer is probably not as significant as an accurate classifier built from genes that are related to cancer.

To verify the selection process for the phenotypes relating to cancer, the following procedure was implemented. For each selection method, we iterated over all the different sets of top genes. For each gene, the corresponding NCBI Gene website (`http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene&cmd=search&term=GENEID` where `GENEID` is the Entrez Gene ID of the gene in question) was downloaded and the text was analyzed using regular expressions. If any of the regular expressions matched any of the text on the page, it was considered a hit. Since we varied the number of significant genes from 10 to 300 the hit count was normalized to one. For example, if 15 out of the 30 selected genes were hits, then a score of 0.5 was recorded. This same process was then also applied ten times to randomly selected genes and the averages of the

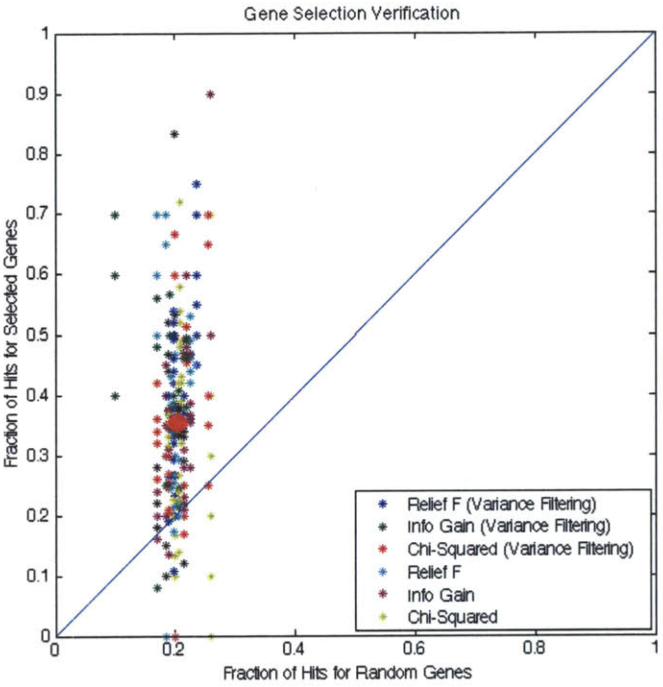


Figure 3-1: Comparison of the fraction of genes that are related to cancer when performing the selection to randomly selected genes. This plot was generated by analyzing the text of the NCBI Gene website for the selected set of genes. Any point that lies above the 45 ° line represents a set of selected genes that had more terms on the website associated with cancer compared to a set of equal size of randomly chosen genes. With only a few exceptions, all points lie above the 45 ° line.

runs were recorded.

Figure 3-1 shows that the selected set of genes are implicated in cancer more often than a random set of genes. Each point is the result of the random selection for the given number of attributes compared with the corresponding result for a given selection method and cancer phenotype. Any point that lies above the 45 ° line depicts a point where the score for the set of selected genes was greater than the score of the randomly selected genes. With a few minor exceptions, this figure shows that the set of selected genes corresponds to genes related to cancer. More precisely, while on average about 20% of the random genes were classified as hits, about twice that many were deemed to be related to cancer when using the selected genes. This fact is represented by the large point roughly in the center of the cloud of points in the plot.

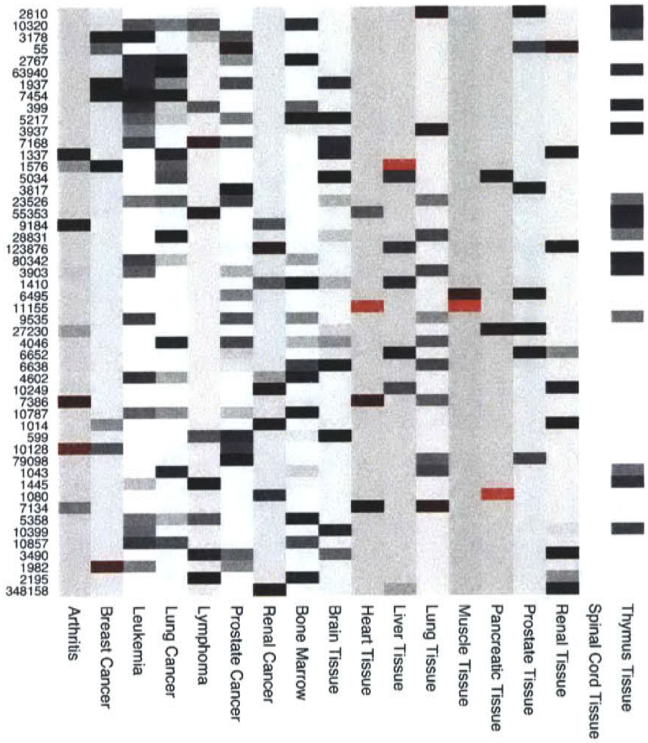


Figure 3-2: 50 most significant genes that appear in multiple phenotypes. These genes were selected on the basis that each one of them was not only in the top 300 most significant genes during the gene selection process, but also was significant in more than one phenotype.

To further verify the validity of the gene selection processes, the 300 most significant genes of all the selection methods for the phenotypes were compared. Each selected gene was scored by its ranking and the 50 highest scoring genes that appeared in multiple phenotypes are reported in figure 3-2. This figure shows the relative scores of the genes of the different phenotypes such that white is a low score, black is medium, and red is the highest. Interestingly, gene 11155 (LDB3) has been shown to be significant in both muscle and heart tissue [46, 35]. Furthermore, gene 399 (RHOH), which has been shown to be associated with lymphoma [24, 43], shows up as being a prominent gene in thymus tissue, leukemia, lymphoma and bone marrow tissue. Similarly, gene 55 (ACPP), which is a gene that is secreted by the epithelial cells of the prostate gland [33], has a high score for both prostate tissue and prostate cancer. Finally, gene 8685 (MARCO) has shown to be expressed in the lung and liver [21, 6] and appears to be significant in the arthritis samples as well. Surprisingly, there are no genes that were selected for spinal chord tissue that are also deemed to be significant in the other phenotypes. Intuitively this does make sense as spinal chord tissue is drastically different from the other phenotypes tested. While figure 3-2 shows the genes and how they relate to the various phenotypes, table A.4 contains the top ten genes from each selection method for each phenotype that were selected at least in two of the selection methods.

Since noise was introduced into four of the seven disease phenotype data, we compared the overlap of the genes that were selected in both the clean and noisy data. Recall, to simulate a real-world situation we added incorrect data to the original set of datasets and repeated the experiments. Table 3.1 contains the fraction of the top 100 genes that overlapped for the six variations of the gene selection method. The values are split up by the three machine learning gene selection methods and by whether or not the variance filter (var filter) was used. Interestingly, there is a large span of the level of conservation between the fraction of genes that were selected in both the "clean" and "noisy" data. At closer examination, however, it appears that the amount of change is correlated with the fraction of experiments that were contributing to noise. For example, the noisy data for lung cancer contained 97 experiments, but

	Var Filter			No Var Filter		
	Relief F	Info Gain	Chi Squared	Relief F	Info Gain	Chi Squared
Arthritis	.40	.36	.31	.41	.36	.32
Lung Cancer	.07	.08	.06	.10	.07	.08
Prostate Cancer	.18	.32	.28	.17	.26	.27
Renal Cancer	.25	.04	.04	.32	.08	.07

Table 3.1: Comparison of Gene Selection With and Without Noise

the clean data only contained 54 samples. As approximately half of the data was data for different phenotypes, it is not surprising that there is such a large change in the genes that were selected. On the other hand, the arthritis data only contained ten erroneous samples and thus it was more likely to pick out the same significant genes in both runs.

3.2 Classification

Using the genes generated from the gene selection phase, the classification was performed independently on the disease and tissue datasets. Since all the testing experiments in each cross validation run are from completely different datasets than those used during training, the results reported show the predictive power of classifiers on previously unseen data. To evaluate each classifier's strength, the F measure was used as the performance measure of each classifier as it was shown to be a a good statistic when there are a lot of negative cases [25]. Briefly, the F measure is the weighted harmonic mean of sensitivity and specificity and is calculated as follows:

$$F = \frac{(1+\alpha) \times \text{sensitivity} \times \text{specificity}}{\alpha \times \text{specificity} + \text{sensitivity}}.$$

α was set to 1 as this gives equal weighting to the sensitivity and specificity. Figure 3-3(a) shows six charts for the results of the disease classification, one for each of the gene selection methods. Every individual bar represents the average F measure for all the different disease phenotypes for a given number of genes. Each group of bars corresponds to the results for a single classifier.

Looking at the charts notice that the relative performance of all the classifiers stay relatively consistent regardless of the gene selection method used. This shows that while selecting a different set of genes to generate classifiers affects the performance, it does not affect how well the classifier performs in general on this type of data. For instance, regardless of the genes used, the K* classifier was among the poorest performers while the KDE and SVM classifiers were among the best. Another interesting observation is that as opposed to the other classifiers, both weighted and unweighted versions of the KDE classifier along with the K* classifier perform better with fewer input genes. As the KDE classifier based the prediction on the (weighted) sum of the probabilities, it is not surprising that this is the case. Since the genes are ranked by their selection score, the top few genes are the most important when performing predictions. As more genes are added, the more noise is introduced into the system because the 300th gene is much less important in predicting the outcome as compared to the first. This notion is further supported by the fact that, in general, the decrease in performance is not as great for the weighted KDE compared to the unweighted KDE. A more detailed table listing the best results for each classifier can be found in table A.5.

Applying the same selection and classification methods to tissue data yielded promising results as well. Figure 3-3(b) depicts the results, again separated by the six different gene selection methods. Here we show the F measure values for the different selection methods for each tissue type when ten genes were used to build each classifier. In this chart, each vertical bar represents the F measure for a particular classifier for a given phenotype. The most striking difference between the tissue data and the disease data is that the average F statistic value is higher in the tissue data. Interestingly, unlike the disease data in which the KDE classifier performed comparably to the SVM classifier, the score of the KDE classifier is only average when using tissue data.

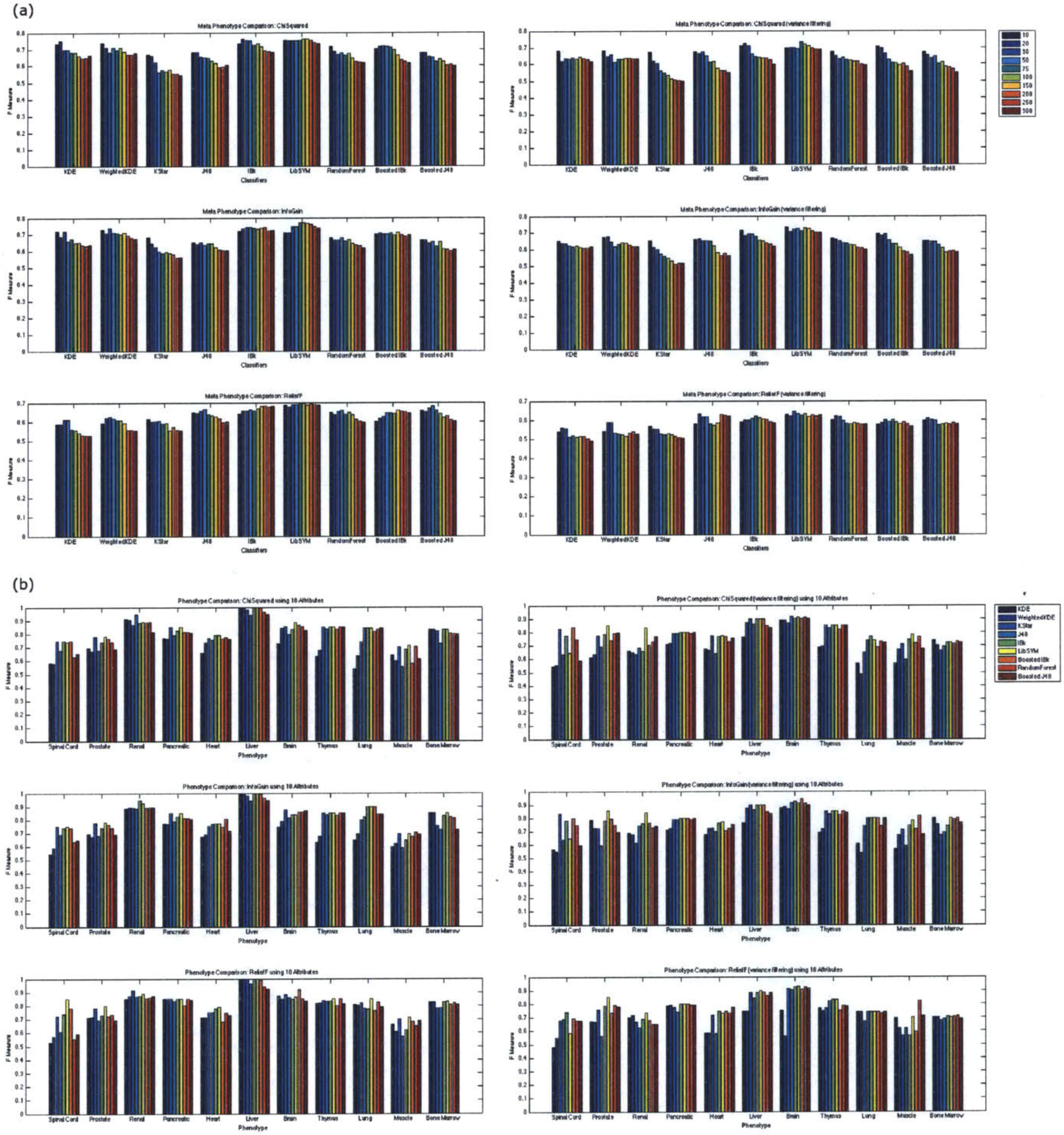


Figure 3-3: (a) Comparison of the results of disease classification using the six different gene selection methods. Each vertical bar is the average F measure for all disease phenotype classification runs for a given number of genes used to build the classifier. For example, the first bar in each group represents the result when ten genes were used to build the classifiers. The second bar represents the result when 20 were used, and so forth. Figure (b) Comparison of the results of tissue classification using the six different gene selection methods. This chart compares the results of the various classifiers for the various tissue types when 10 genes were used to build the classifiers.

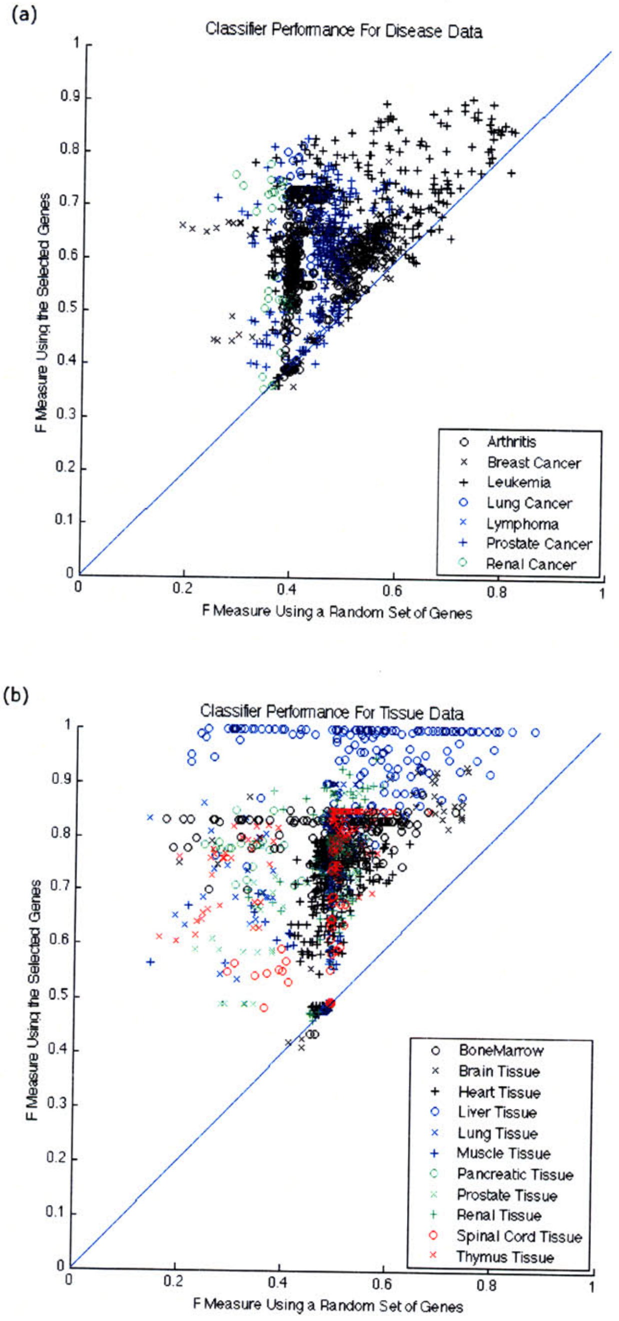


Figure 3-4: Comparison of performance between classifiers generated using a random set of genes and the corresponding classifier made from the selected set of genes. Each point represents the F measure of the classifier built from the genes from one of the six selection methods and a fixed number of attributes (such as 10, 20, etc) and the corresponding classifier built by picking a random set of 10, 20, etc. genes. Any point that lies above the 45 ° line depicts a classifier that performed better when using the selected set of genes as opposed to a random set of genes. Figure (a) represents the results from the disease specific phenotypes while (b) depicts the results for the tissue specific phenotypes.

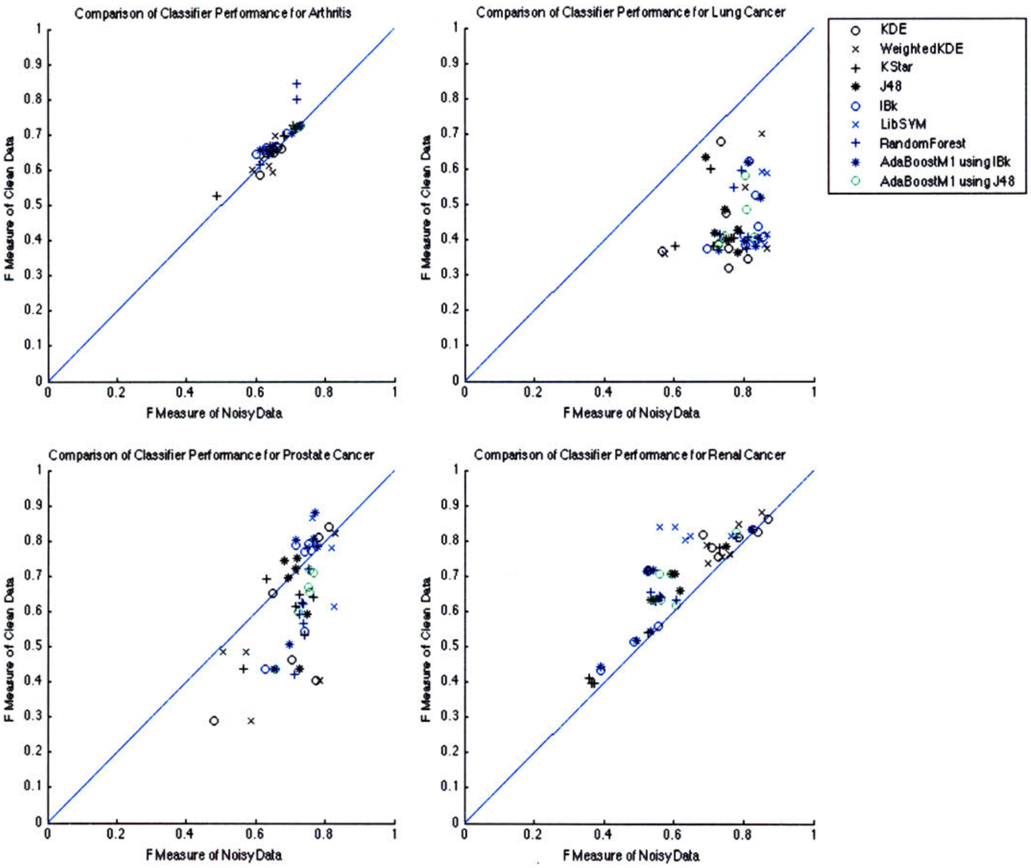


Figure 3-5: Performance comparison of classifiers built using the genes from the various gene selection methods on the clean and noisy data. Any point that lies above the 45 ° line represents a point where the classifier built and tested using the clean data outperformed the corresponding classifier built and tested using the noisy data.

Just as random genes were selected to perform the gene selection validation shown in figure 3-1, random genes were selected and the classification process was performed ten times each. If the aforementioned results were obtained merely by chance, one would expect that selecting random genes and using them to build classifiers would yield similar results. Figure 3.2 shows the results of this test and shows that the vast majority performed better when using the selected set of genes compared to a random set. As one would expect, using a random set of genes created classifiers that were correct approximately half the time with F measures of, on average 0.46 for the diseased data and 0.49 for the tissue data.

Since we expect noisy data to be present in a large-scale phenotype prediction database, the performance of the classifiers built using the clean data was compared to the performance of the classifiers built using the noisy data. The results in figure 3-5 show that the performance of many of the classifiers stays roughly the same. Each plot has six points for the various classifiers, one point for each of the six gene selection methods. The two exceptions are the classifiers for lung cancer and renal cancer data. The performance of the former is worse using the clean data while the latter performs better when using the clean data.

Although increased performance using less noisy data is a good sign, the substantial decrease in performance of the lung cancer classifiers is initially worrisome. This issue was addressed by testing whether the lung cancer classifier, built using approximately half cancer and half other data, was in actuality classifying whether a new sample was a lung tissue sample or not. To perform this test we built the classifiers from the lung cancer data and then used those classifiers to classify whether or not a new sample was lung tissue. As depicted in figure 3-6, it appears that the classifiers built using the noisy lung cancer data were indeed classifying lung tissue! This figure shows the box plot of the F measures obtained by all classifiers either trained using the clean data or noisy data, and then testing those classifiers using the lung data. Surprisingly, the performance of the classifiers trained on the disease data show very comparable performance in the classification of lung data as to those classifiers built using the tissue data.

Our primary results obtained by combining multiple datasets are confirmed by the much smaller study of Warnat et al. [45]. They examined six datasets pertaining to three different cancers and achieved classification accuracies of 97% for prostate cancer, 89% for breast cancer, and 90% for leukemia. For these phenotypes, we achieved accuracies of 90%, 89%, and 92% using the KDE classifier on our expanded dataset. Unlike their study, however, we show that these classification results are possible when using many different datasets that were performed on a wide array of platforms. Furthermore, our 90% accuracy for the prostate cancer data included noise! Without the noise we achieve 93% accuracy for prostate cancer. Similarly, our classification results are in line with the accuracies compiled in Cho et al. [12] for

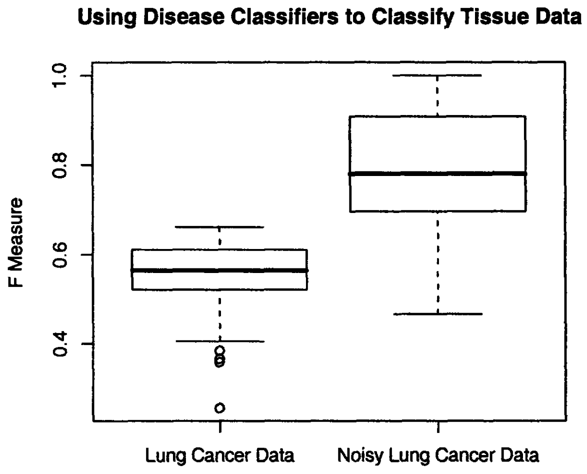


Figure 3-6: This figure shows the performance of classification of lung tissue samples using either classifiers built from the clean lung cancer data or the noisy lung cancer data. The good performance of the classifiers built using the noisy lung cancer data indicates that the classifier was most likely classifying items as lung samples as opposed to lung cancer samples.

classifiers built from single datasets. In addition, through the use of random sampling for both the NCBI gene validation and phenotype classification, we verified that it is possible to use a heterogenous database to perform gene selection and phenotype classification.

3.3 Discussion

We have presented the foundation for an automated large-scale phenotype prediction system based on microarray data. Large heterogenous sets of datasets relating to disease and tissue phenotypes were used to select significant genes for seven disease phenotypes and 11 tissue types. Using those genes, various classifiers were trained and tested through leave-dataset-out cross validation. The results of both the gene selection and phenotype classification show that it is possible to use a large microarray database as a "black-box" to classification tool. Although our results are promising, there are still critical pieces of the puzzle that require further attention.

Using the rank normalization and permutation t-test based gene selection modeled on the work by Butte and Kohane [9], the number of genes that were deemed significant varied greatly. For example, only 11 genes for the muscle tissue had a multiple-hypothesis corrected p-value of below 0.01 in 80% of the cross validation runs. On the other hand, there were over 8000 genes that matched this criteria for the breast cancer data. As a matter of fact, as depicted in figure 3-7(a), there appears to be a linear relation between the number of experiments used to perform the analysis and the number of genes that pass the permutation t-test filter. A possible explanation for this result is the difference in sample size. As all experiments were used for all phenotypes, the phenotypes that have the fewest associated experiments naturally have the most experiments that are not associated with it; thus, the largest difference in sample size. It has previously been shown by Legendre and Borcard [31] that a large difference in sample size can reduce the power of the t-test, whether it be using Welch correction or permutation based. The problem with sample imbalance affecting the differentially expressed genes was also noted in Yang et al. [47]. For example, they noted that for a given dataset that had the experiments equally divided between 2 classes they achieved a precision rate of about 80% and a recall rate a little under 75%. When the ratio of the number of experiments in the two classes was changed to five to one, the precision rate dropped on average 5% and the recall rate 20%.

To further examine this issue, 200 genes were randomly selected for each phenotype. The variance of the rank normalized intensities for each of the genes was then used to generate the boxplot depicted in figure 3-7(b). One of the most striking features of this plot is how the average variance of the genes used for arthritis are nearly four times greater than that of all other phenotypes. Even more surprising is that the so-called "noisy" arthritis data has a slightly higher average variance than the "clean" data. When we analyzed the raw data, we saw that indeed, the variance of the intensity values in one of the two arthritis datasets used (GDS711) is extremely high. For comparison, examine figure 3-8 that depicts a heatmap of 100 random genes and their corresponding rank normalized intensity values in two breast cancer and two arthritis datasets. The difference in the variance between the clean and noisy

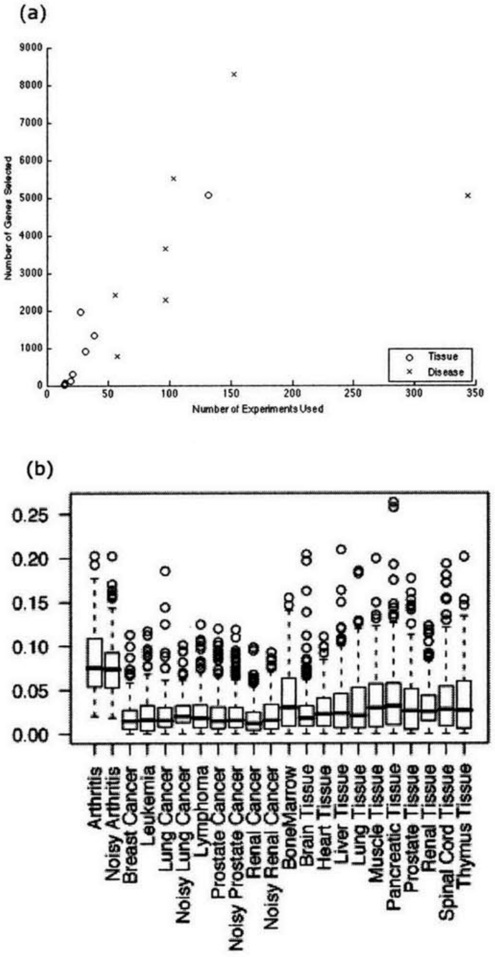


Figure 3-7: (a) Comparison of the number of experiments used to perform the gene selection to the number of genes that were deemed significant after the permutation t-test filter for the disease and tissue data. (b) The box plot for the average variance for each phenotype. 200 genes were randomly selected for each phenotype and the variance of the ranked intensity values with respect to the phenotype.

arthritis data may be explained by the difference in sample size. Since there are only 56 experiments for the noisy data, and ten less for the clean data, it is hard to get a true estimate of the underlying distribution and thus the variance. Thus, it is also not surprising to note that the classifiers built using this arthritis data also had the lowest sensitivity and specifitcity.

If one ignores the arthritis data, however, the remaining variances are all quite low and correlate well (correlation of -0.81) with the number of genes that are deemed significant using the permutation t-test filter. In other words, the lower the average

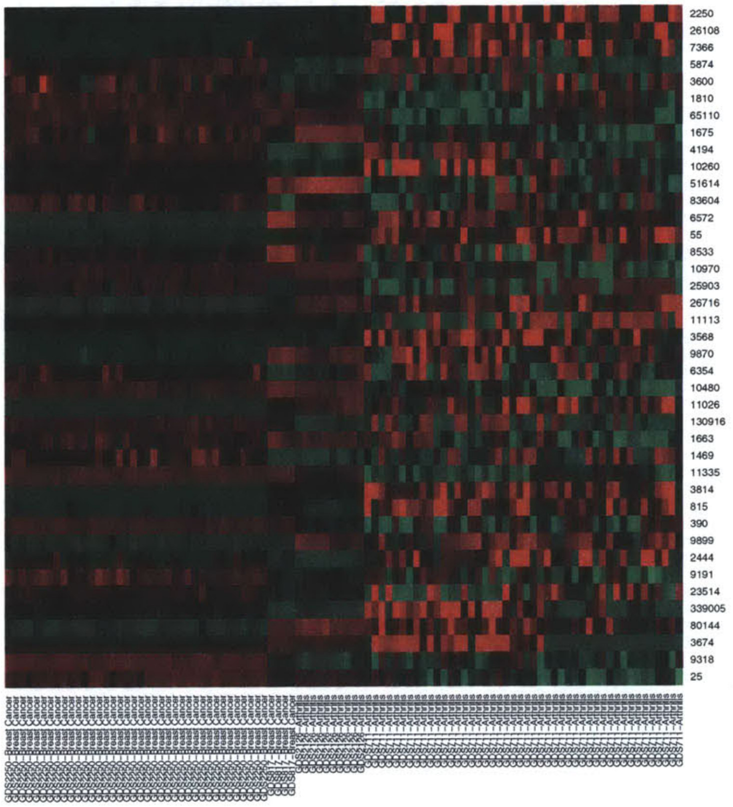


Figure 3-8: Comparison of 50 random genes in two breast cancer datasets and two arthritis datasets. One will notice the large variation of rank normalized intensity values for one of the arthritis datasets (GDS711) is causing the large variance noted in figure 3-7(b).

variance within the group, the greater the number of genes that had lower p-values. As one would expect, the average variances for lung cancer, prostate cancer, and renal cancer were also lower when the erroneous datasets were removed from the data. Interestingly, this variance correlates to some degree with the number of experiments used (correlation of -0.69) but not with the number of different datasets used (correlation of 0.03). Furthermore, there is a slight correlation between the number of datasets or the number of experiments and the performance of the classifiers as measured by the F measure (0.24 for the former and 0.11 for the latter). While these correlations are very low, it does point to the fact that adding data is not penalizing the classification process.

The large difference in average variances between the disease data and tissue data is also related to the sample size. Examining the tissue specific data and excluding brain tissue, each phenotype in the tissue data had about 20 experiments associated with it. This is in contrast with at least 46 samples in the disease specific data. One will also notice that the brain tissue data, which had 132 experiments associated with it, had the lowest average variance. In other words, as the amount of data is increased, a better estimate of the true underlying distribution can be generated.

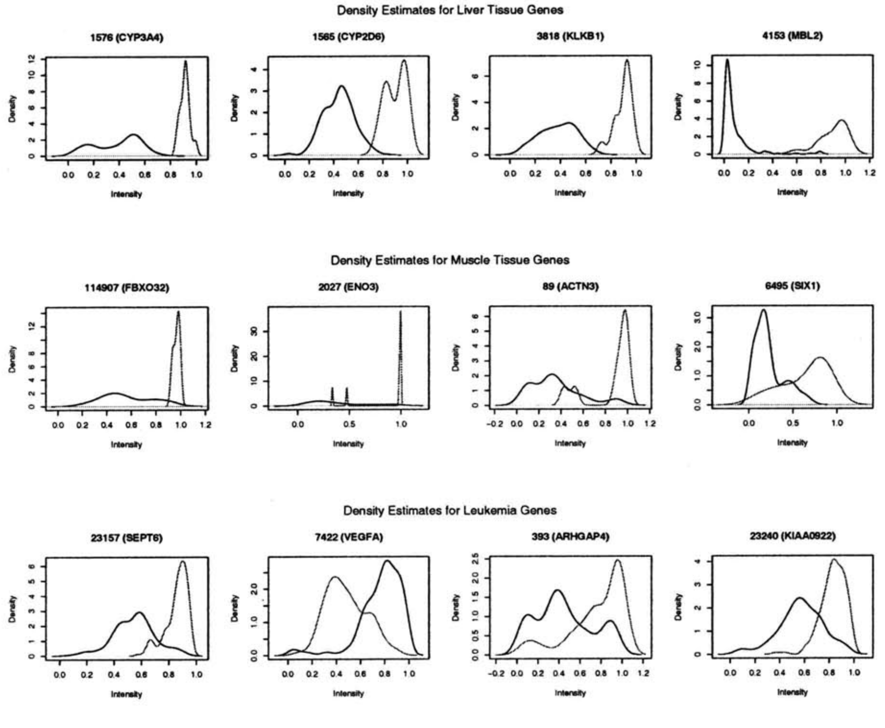


Figure 3-9: Density estimates for four of the most predictive genes for (a) liver tissue, (b) muscle tissue, (c) and leukemia.

To address the variability of the performance of the KDE classifier, the density estimates of the genes for some of the best performing KDE classifiers were compared with density estimates of the genes for the poorest performer. The plots of four of the top ten genes from liver and muscle tissue along with leukemia are shown in figure 3.3, such that one curve in each plot shows the rank normalized intensity values for the phenotype of interest and the other depicts the data for all other tissues. One will immediately notice that the separation of the density estimates for the liver tissue's genes is substantially greater than that for the other phenotypes. In other words, the average ranks for the intensity values for these four genes across all experiments with normal liver tissue samples are significantly different from the average ranks of these four genes in other tissues. Unlike the density estimates for the liver tissue, the estimates for the muscle tissue data show both greater overlap and longer tails, both of which are problematic for classification. Clearly, the greater the overlap the greater the intersection between the two curves and thus the greater the change of erroneous classification. While long tails do not imply a large overlap between the two density estimates, they do imply that the range of values for these genes are greater. Since the range is greater it is much harder to predict the correct class as there are large regions where either class could potentially be correct. Although the predictive power of the muscle tissue classifier was not optimal, it is worthy to note that many of the genes depicted in figure 3.3 have been shown to be associated with their respective phenotypes [33, 30, 13, 16].

Furthermore, the previous discussion about the amount of data present when training the classifiers sheds light on an important characteristic of the KDE classifier. In general, with the exception of liver tissue which only had 21 samples, the classifiers for the phenotypes with the most data performed better. Although the average score for all classifiers was higher when using more data, the average difference between the top F measure for brain and renal tissue, which had 132 and 39 samples respectively, was significantly lower than that of the difference between the scores of the best classifiers and the KDE classifiers built on the other tissue data. Therefore, as public microarray repositories grow and more data related to each phenotype can be found, the more accurate the classifier will become.

One of the largest bottlenecks in this study was obtaining the training data. Although a large portion was automated, we still had to intervene at several points in the process. While the use of UMLS concepts is a logical starting point, it does not allow for great enough sensitivity in understanding what the dataset is truly about.

The use of regular expressions to annotate the individual experiments with being diseased or normal was a simple way to begin. As a matter of fact, the recent and independent work by Dudley and Butte [17] shows that that classifying experiments as "diseased" or "normal" can be performed fairly accurately using regular expressions. In the case of this work, we made certain to be overly stringent with our rules, as evident by the high specificity and relatively low sensitivity. This ensured that the experiments that were labeled as "diseased" were truly diseased, but also deprived the system of many other possible experiments. As a matter of fact, a large portion of the experiments that were missed were those that were labeled with something to the effect of "control" in experiments where the control referred to the untreated disease state. Another large source of missed experiments were those that were labeled with specific cell lines that can only be deemed as "diseased" through expert knowledge or explicit rules. As is evident by the significant fluctuation in the gene selection process in the presence of noise, it is vital to minimize the number of mislabeled experiments.

Any strict rule-based system is limited and will not be able to capture all the intricacies of a language such as English. Furthermore, using natural language processing to automatically label datasets and experiments with phenotypic information is undoubtedly useful for data that has already been published, but an alternative approach may be more fruitful for new data. For example, if scientists who submit microarray datasets are required to label their data when submitting it, many of these problems could be solved at the source. We are currently working on a method that allows a user to add new datasets and experiments to an online database. Through this online portal users will be able to first annotate their own experiments with the correct phenotypic data and then view their data in the context of all other data already in the database. Unlike GEO, which only provides a repository of microarray data, we envision an exploratory tool that can be used to leverage the vast amount of existing knowledge. Another possibility is to formulate the natural language processing problem in the form of a CAPTCHA problem [44] and ask web users to annotate the data when signing up for online accounts on various websites.

It is with this framework in mind that the usefulness of the KDE classifier becomes apparent. Although the SVM classifier matches or outperforms the KDE classifier in many instances when there was little data, there are three important advantages of the KDE classifier for large datasets. First the density estimates are independent of the number of experiments. Since a fixed number of points are used to describe a density estimate, each estimate will only ever be as large as this fixed number. Secondly, and more importantly, density estimates are easy to update. A new training experiment can be added to an existing density estimate by simply adding it to the estimate. Even if the entire density estimate needs to be recomputed, it can be recomputed on a gene by gene basis rather than by experiment or dataset. This is a highly desirable characteristic if a classifier needs to be kept up-to-date while new data is being added to a database. Finally, the more data present, the better the predictive results become. In other words, the KDE classifier represents a non-memory and non-compute intensive classifier that performs better as more information is added that performs equally or better to many existing classifiers.

Appendix A

Tables

Table A.1: Datasets used for disease phenotypes

Phenotype	ID	Dataset Title
Arthritis		
C0003864	GDS2126	Rheumatoid arthritis: synovial tissues
	GDS711	Juvenile rheumatoid arthritis expression profiles in mononuclear cells
Breast Cancer		
C0678222	GDS1069	Homeobox gene HOXA5 induction: time course
	GDS823	Breast cancer cell expression profiles (HG-U133B)
	GDS483	DACH1-responsive genes
	GDS1329	Molecular apocrine breast tumors
	GDS2250	Basal-like breast cancer tumors
	GDS1508	Tamoxifen effect on endometrioid carcinomas
	GDS817	Breast cancer cell expression profiles (HG-U95A)
	GDS1925	Estrogen receptor alpha positive breast cancer cells response to hyperactivation of MAPK pathway
	GDS820	Breast cancer cell expression profiles (HG-U133A)
	GDS1664	Parathyroid hormone-related protein knockdown effect on breast cancer cells
	GDS360	Breast cancer and docetaxel treatment
	GDS992	Endoplasmic reticulum membrane-associated genes in breast cancer cell line MCF-7
Leukemia		
C0023418	GDS1454	B-cell chronic lymphocytic leukemia subtypes
	GDS1604	Ionizing radiation effect on monocytic leukemia cells
	GDS1064	Acute myeloid leukemia subclasses
	GDS1388	B-cell chronic lymphocytic leukemia progression
	GDS760	T-cell acute lymphoblastic leukemia and T-cell lymphoblastic lymphoma comparison
	GDS1886	Moderate hypothermia effect in vitro
	GDS2251	Myeloid leukemia cell lines
	GDS330	Acute lymphoblastic leukemia treatment responses
	GDS596*	Large-scale analysis of the human transcriptome (HG-U133A)

Continued on Next Page...

Table A.1 – Continued

Phenotype	ID	Dataset Title
Lung Cancer		
C0684249	GDS1650	Pulmonary adenocarcinoma
C0152013	GDS1688	Various lung cancer cell lines
	GDS1312	Squamous lung cancer
Lymphoma		
C0024299	GDS1617	Motexafin gadolinium and zinc effect on Ramos B-cell lymphoma line
	GDS1750	Mantle cell lymphoma cell lines (HG-U133A)
	GDS1751	Mantle cell lymphoma cell lines (HG-U133B)
	GDS2295	Aplidin and cytarabine effect on diffuse large B cell lymphoma cell line
	GDS1419	Classical Hodgkin's lymphoma: T cell expression profile
	GDS596*	Large-scale analysis of the human transcriptome (HG-U133A)
	GDS181*	Large-scale analysis of the human transcriptome (HG-U95A)
Prostate Cancer		
C0600139	GDS1736	Arachidonic acid effect on prostate cancer cells
C0033578	GDS1390	Prostate cancer progression after androgen ablation
	GDS1439	Prostate cancer progression
	GDS1423	Lunasin effect on prostate epithelial cells
	GDS1746	Primary epithelial cell cultures from prostate tumors
	GDS181*	Large-scale analysis of the human transcriptome (HG-U95A)
Renal Cancer		
C1378703	GDS1344	Papillary renal cell carcinoma classification
	GDS507	Renal clear cell carcinoma (HG-U133B)
	GDS505	Renal clear cell carcinoma (HG-U133A)

* Indicates that these datasets were manually added to the list

Table A.2: Datasets used for tissue phenotypes

	GDS1059	GDS1096	GDS1663	GDS1726	GDS181	GDS1962	GDS2190	GDS2191	GDS422	GDS423	GDS424	GDS425	GDS426	GDS505	GDS507	GDS53	GDS552	GDS596	GDS651	GDS670	GDS707	GDS838	GDS969
Bone Marrow	●	●							●	●	●	●	●			●	●	●				●	●
Brain Tissue		●		●	●	●	●	●	●	●	●	●	●					●			●		
Heart Tissue		●			●				●	●	●	●	●					●	●				
Liver Tissue		●	●		●				●	●	●	●	●					●					
Lung Tissue		●			●				●	●	●	●	●					●		●			
Muscle Tissue		●							●	●	●	●	●					●					
Pancreatic Tissue		●			●				●	●	●	●	●					●					
Prostate Tissue		●			●				●	●	●	●	●					●					
Renal Tissue		●	●		●				●	●	●	●	●	●	●			●					
Spinal Cord Tissue		●			●				●	●	●	●	●					●					
Thymus Tissue		●			●				●	●	●	●	●					●					

Table A.3: The titles of the tissue phenotype datasets in table A.2

ID	Dataset Title
GDS1059	Acute myeloid leukemia response to chemotherapy
GDS1096	Normal tissues of various types
GDS1663	Expression data from different research centers
GDS1726	HIV encephalitis: brain frontal cortex
GDS181	Large-scale analysis of the human transcriptome (HG-U95A)
GDS1962	Glioma-derived stem cell factor effect on angiogenesis in the brain
GDS2190	Bipolar disorder: dorsolateral prefrontal cortex
GDS2191	Bipolar disorder: orbitofrontal cortex
GDS422	Normal human tissue expression profiling (HG-U95A)
GDS423	Normal human tissue expression profiling (HG-U95B)
GDS424	Normal human tissue expression profiling (HG-U95C)
GDS425	Normal human tissue expression profiling (HG-U95D)
GDS426	Normal human tissue expression profiling (HG-U95E)
GDS505	Renal clear cell carcinoma (HG-U133A)
GDS507	Renal clear cell carcinoma (HG-U133B)
GDS53	CD34+ cell analysis
GDS552	Essential thrombocythemia megakaryocytes
GDS596	Large-scale analysis of the human transcriptome (HG-U133A)
GDS651	Heart failure arising from different etiologies
GDS670	Emphysema lung tissue expression profiling
GDS707	Aging brain: frontal cortex expression profiles at various ages
GDS838	Imatinib effects on chronic myelogenous leukemia CD34+ cells
GDS969	Bone marrow prolonged storage effects

Table A.4: Top selected genes for various phenotypes

		Var Filter			No Var Filter		
Phenotype	Gene ID	Relief F	Info Gain	Chi Squared	Relief F	Info Gain	Chi Squared
Arthritis							
	3020 (H3F3A)	●	●	●	●	●	●
	6201 (RPS7)	●	●	●	●	●	●
	8667 (EIF3H)	●	●	●	●	●	●
	6161 (RPL32)	●	●	●	●	●	●
	6136 (RPL12)	●	●	●	●	●	
	6128 (RPL6)		●	●		●	●
	4691 (NCL)		●	●		●	●
	6147 (RPL23A)		●	●		●	●
	3320 (HSP90AA1)		●	●		●	●
	3150 (HMGN1)		●	●		●	●
	10575 (CCT4)	●			●		
	1915 (EEF1A1)	●			●		
	6146 (RPL22)	●			●		
	7178 (TPT1)	●			●		
Breast Cancer							
	1982 (EIF4G2)	●	●	●	●	●	●
	23352 (UBR4)		●	●		●	●
	3049 (HBQ1)		●	●		●	●
	6168 (RPL37A)		●	●		●	
	64816 (CYP3A43)		●	●			●
	2547 (XRCC6)				●	●	●
	6013 (RLN1)					●	●
	64096 (GFRA4)		●	●			
	56673 (C11orf16)					●	●
	10500 (SEMA6C)					●	●
	150684 (COMMD1)		●	●			
	91746 (YTHDC1)					●	●
	2044 (EPHA5)		●	●			
	829 (CAPZA1)					●	●
Leukemia							
	23157 (SEPT6)		●	●		●	●
	4602 (MYB)				●	●	●
	9555 (H2AFY)		●	●			●
	566 (AZU1)	●			●		
	6189 (RPS3A)		●	●			
	51433 (ANAPC5)		●	●			
	896 (CCND3)		●	●			

Continued on Next Page...

Table A.4 – Continued

Phenotype	Gene ID	Var Filter			No Var Filter		
		Relief F	Info Gain	Chi Squared	Relief F	Info Gain	Chi Squared
	9535 (GMFG)					•	•
	863 (CBFA2T3)					•	•
	3676 (ITGA4)				•		•
	7454 (WAS)		•	•			
	5579 (PRKCB1)					•	•
	3635 (INPP5D)					•	•
	23240 (KIAA0922)		•	•			
	100 (ADA)					•	•
	7422 (VEGFA)		•	•			
Lung Cancer							
	1521 ()		•	•		•	•
	28831 (IGLJ3)		•	•		•	•
	7454 (WAS)		•	•		•	•
	10901 (DHRS4)	•			•		
	2123 (EVI2A)					•	•
	2877 (GPX2)			•			•
	3537 (IGLC1)					•	•
	3538 (IGLC2)					•	•
	3880 (KRT19)	•			•		
	28299 (IGKV1-5)		•	•			
	28815 (IGLV2-14)					•	•
	28793 (IGLV3-25)					•	•
	51400 (PPME1)	•			•		
	3535 (IGL@)					•	•
Lymphoma							
	57379 (AICDA)	•	•	•		•	•
	5079 (PAX5)		•		•	•	•
	1488 (CTBP2)	•	•	•		•	
	933 (CD22)	•		•	•		
	4099 (MAG)	•	•	•			
	149699 (GTSF1L)			•		•	•
	931 (MS4A1)				•	•	•
	55653 (BCAS4)					•	•
	5872 (RAB13)		•			•	
	29802 (VPREB3)					•	•
	50865 (HEBP1)			•			•
	6689 (SPIB)		•	•			
	90925 ()	•		•			
	23240 (KIAA0922)		•	•			

Continued on Next Page...

Table A.4 – Continued

Phenotype	Gene ID	Var Filter			No Var Filter		
		Relief F	Info Gain	Chi Squared	Relief F	Info Gain	Chi Squared
	5450 (POU2AF1)					•	•
	939 (CD27)					•	•
	928 (CD9)	•			•		
	8204 (NRIP1)	•			•		
Prostate Cancer							
	51109 (RDH11)		•	•			•
	10257 (ABCC4)			•			•
	27122 (DKK3)					•	•
	397 (ARHGDIB)					•	•
	205860 (TRIML2)	•			•		
	4853 (NOTCH2)		•	•			
	1452 (CSNK1A1)		•	•			
	8992 (ATP6V0E1)		•	•			
	7082 (TJP1)					•	•
	9231 (DLG5)					•	•
	6170 (RPL39)		•	•			
	292 (SLC25A5)		•			•	
	5587 (PRKD1)					•	•
	283677 ()					•	•
	90993 (CREB3L1)					•	•
Renal Cancer							
	55195 (C14orf105)	•	•	•	•	•	•
	348158 (ACSM2B)	•	•	•	•	•	•
	123876 (ACSM2A)	•	•	•		•	•
	3773 (KCNJ16)	•	•	•	•		•
	1014 (CDH16)	•		•	•		
	4036 (LRP2)				•	•	•
	10249 (GLYAT)		•	•			•
	6519 (SLC3A1)	•	•	•			
	83737 (ITCH)		•	•		•	
	6299 (SALL1)					•	•
	6540 (SLC6A13)					•	•
	79799 (UGT2A3)	•			•		
	2222 (FDFT1)		•			•	
	55867 (SLC22A11)	•			•		
	2018 (EMX2)	•			•		
BoneMarrow							
	4353 (MPO)	•	•	•	•	•	•
	1991 (ELA2)	•	•	•	•		•

Continued on Next Page...

Table A.4 – Continued

		Var Filter			No Var Filter		
Phenotype	Gene ID	Relief F	Info Gain	Chi Squared	Relief F	Info Gain	Chi Squared
	5796 (PTPRK)	●	●	●		●	●
	6037 (RNASE3)		●	●		●	●
	5654 (HTRA1)		●	●		●	●
	2526 (FUT4)		●	●		●	●
	8404 (SPARCL1)	●	●		●	●	
	6284 (S100A13)		●	●		●	
	1410 (CRYAB)			●	●		
	6036 (RNASE2)			●			●
	25893 (TRIM58)	●			●		
	3045 (HBD)					●	●
	2993 (GYPA)					●	●
	932 (MS4A3)	●			●		
	10562 (OLFM4)	●			●		
	212 (ALAS2)	●			●		
	4171 (MCM2)		●			●	
Brain Tissue							
	230 (ALDOC)	●	●	●			
	6638 (SNRPN)		●	●			
	6812 (STXBP1)					●	●
	8926 (SNURF)		●	●			
	6616 (SNAP25)					●	●
	10900 (RUNDC3A)					●	●
	801 (CALM1)		●	●			
	599 (BCL2L2)		●	●			
	1808 (DPYSL2)		●	●			
	6000 (RGS7)					●	●
	22883 (CLSTN1)		●	●			
	3800 (KIF5C)					●	●
	2664 (GDI1)		●	●			
	8237 (USP11)		●	●			
	1759 (DNM1)					●	●
	2775 (GNAO1)					●	●
	10439 (OLFM1)					●	●
	7102 (TSPAN7)					●	●
Heart Tissue							
	79933 (SYNPO2L)	●	●	●	●		
	2170 (FABP3)		●	●		●	●
	7139 (TNNT2)	●	●	●	●		
	1760 (DMPK)		●	●		●	●

Continued on Next Page...

Table A.4 – Continued

		Var Filter			No Var Filter		
Phenotype	Gene ID	Relief F	Info Gain	Chi Squared	Relief F	Info Gain	Chi Squared
	7134 (TNNC1)				●	●	●
	27231 (ITGB1BP3)	●	●	●			
	58498 (MYL7)				●	●	●
	1158 (CKM)	●	●		●		
	5441 (POLR2L)		●			●	
	70 (ACTC1)					●	●
	6331 (SCN5A)					●	●
	4634 (MYL3)		●	●			
	8048 (CSRP3)	●			●		
	27129 (HSPB7)					●	●
	518 (ATP5G3)		●	●			
	6508 (SLC4A3)					●	●
	4607 (MYBPC3)		●	●			
	51778 (MYOZ2)	●			●		
	1160 (CKMT2)					●	●
Liver Tissue							
	4153 (MBL2)	●	●	●	●		
	130 (ADH6)					●	●
	3080 (CFHR2)					●	●
	1361 (CPB2)					●	●
	1576 (CYP3A4)		●	●			
	3273 (HRG)					●	●
	3697 (ITIH1)					●	●
	1565 (CYP2D6)		●	●			
	3240 (HP)					●	●
	5950 ()					●	●
	3700 (ITIH4)		●	●			
	3698 (ITIH2)		●	●			
	5004 (ORM1)					●	●
	720 (C4A)		●	●			
	1559 (CYP2C9)		●	●			
	3929 (LBP)		●	●			
	3818 (KLKB1)		●	●			
	197 (AHSG)					●	●
	1571 (CYP2E1)					●	●
	721 (C4B)		●	●			
Lung Tissue							
	7080 (NKX2-1)	●	●	●	●	●	●
	177 (AGER)	●	●	●	●	●	●

Continued on Next Page...

Table A.4 – Continued

		Var Filter			No Var Filter		
Phenotype	Gene ID	Relief F	Info Gain	Chi Squared	Relief F	Info Gain	Chi Squared
	6439 (SFTPB)	●	●	●		●	●
	7356 (SCGB1A1)	●	●	●		●	●
	3107 (HLA-C)		●	●		●	●
	4091 (SMAD6)		●	●		●	●
	51208 (CLDN18)	●		●	●		●
	6441 (SFTPD)				●		●
	3949 (LDLR)		●	●			
	5225 (PGC)		●			●	
	6440 (SFTPC)					●	●
	27074 (LAMP3)				●		●
Muscle Tissue							
	11047 (ADRM1)	●	●	●		●	●
	11155 (LDB3)	●	●	●	●	●	
	2027 (ENO3)	●	●	●			
	6495 (SIX1)	●	●	●			
	2314 (FLII)	●	●	●			
	786 (CACNG1)				●	●	●
	81786 (TRIM7)	●	●	●			
	7957 (EPM2A)				●	●	●
	5708 (PSMD2)	●	●	●			
	4837 (NNMT)				●	●	●
	10653 (SPINT2)				●	●	●
	57157 (PHTF2)	●	●	●			
	10324 (KBTBD10)	●			●		●
	2997 (GYS1)				●	●	●
	781 (CACNA2D1)				●	●	●
	89 (ACTN3)	●	●	●			
	4330 (MN1)				●	●	●
	114907 (FBXO32)		●	●			
	8260 (ARD1A)					●	●
Pancreatic Tissue							
	1357 (CPA1)	●	●	●	●	●	●
	5407 (PNLIPRP1)	●	●	●	●	●	●
	51032 ()	●	●	●		●	●
	63036 ()	●	●	●		●	●
	5644 (PRSS1)		●	●		●	●
	1506 (CTRL)		●	●		●	●
	1080 (CFTR)	●			●	●	●
	2813 (GP2)		●	●		●	●

Continued on Next Page...

Table A.4 – Continued

		Var Filter			No Var Filter		
Phenotype	Gene ID	Relief F	Info Gain	Chi Squared	Relief F	Info Gain	Chi Squared
	5406 (PNLIP)				●	●	●
	5319 (PLA2G1B)				●	●	●
	3375 (IAPP)	●			●		
	5645 (PRSS2)		●	●			
	1056 (CEL)	●			●		
	154754 ()		●	●			
	2641 (GCG)	●			●		
	5646 (PRSS3)		●	●			
	5408 (PNLIPRP2)	●			●		
Prostate Tissue							
	6652 (SORD)	●	●	●	●	●	●
	4477 (MSMB)	●	●	●	●	●	●
	3817 (KLK2)	●	●	●	●	●	●
	6495 (SIX1)		●	●		●	●
	57535 (KIAA1324)	●	●	●	●		
	2316 (FLNA)		●	●		●	●
	6406 (SEMG1)	●	●	●			
	354 (KLK3)				●	●	●
	8000 (PSCA)	●	●	●			
	7103 (TSPAN8)				●	●	●
	10481 (HOXB13)	●	●	●			
	55 (ACPP)				●	●	●
	1292 (COL6A2)					●	●
	79098 (C1orf116)	●		●			
	25800 (SLC39A6)					●	●
Renal Tissue							
	5174 (PDZK1)				●	●	●
	6819 (SULT1C2)	●		●			●
	2168 (FABP1)		●	●			
	64849 (SLC13A3)		●	●			
	9356 (SLC22A6)					●	●
	6561 (SLC13A1)	●		●			
	5340 (PLG)		●	●			
	7369 (UMOD)					●	●
	51463 (GPR89B)		●	●			
	949 (SCARB1)		●	●			
	4036 (LRP2)					●	●
	54852 (PAQR5)		●	●			
	3772 (KCNJ15)					●	●

Continued on Next Page. . .

Table A.4 – Continued

		Var Filter			No Var Filter		
Phenotype	**Gene ID**	Relief F	Info Gain	Chi Squared	Relief F	Info Gain	Chi Squared
	51626 (DYNC2LI1)		●	●			
	6568 (SLC17A1)					●	●
	159963 (SLC5A12)		●	●			
Spinal Cord Tissue							
	2342 (FNTB)	●	●	●	●	●	●
	975 (CD81)		●	●		●	●
	65108 (MARCKSL1)		●	●		●	●
	780 (DDR1)		●	●		●	●
	6678 (SPARC)		●	●		●	●
	2261 (FGFR3)	●	●	●			
	1028 (CDKN1C)	●	●	●			
	358 (AQP1)	●	●	●			
	4359 (MPZ)				●	●	●
	7368 (UGT8)				●	●	●
	5653 (KLK6)					●	●
	79152 (FA2H)				●		●
	3730 (KAL1)					●	●
	7846 (TUBA1A)		●	●			
	4744 (NEFH)				●	●	
Thymus Tissue							
	3932 (LCK)	●	●	●	●	●	●
	6955 (TRA@)	●	●	●	●	●	●
	915 (CD3D)		●	●		●	●
	51176 (LEF1)	●	●	●	●		
	10279 (PRSS16)		●	●		●	●
	914 (CD2)		●	●		●	●
	10803 (CCR9)	●	●	●	●		
	3861 (KRT14)				●	●	●
	6504 (SLAMF1)				●	●	●
	913 (CD1E)	●	●		●		
	28738 (TRAJ17)	●	●	●			
	28611 (TRBV5-4)					●	●
	28566 (TRBV21-1)					●	●
	3866 (KRT15)	●			●		
	925 (CD8A)		●	●			
	28568 (TRBV19)					●	●

Table A.5: Classifier performance for various phenotypes

Phenotype	Classifier	Num Genes	F	Sensitivity	Specificity
Arthritis	RandomForest	100	.84	83.42	95.27
	KStar	10	.72	75.00	70.47
	J48	10	.72	75.00	70.47
	IBk	10	.72	75.00	70.47
	LibSVM	10	.72	75.00	70.47
	Boosted IBk	10	.72	75.00	70.47
	Boosted J48	10	.72	75.00	70.47
	WeightedKDE	10	.72	73.89	70.35
	KDE	10	.71	73.59	69.79
Breast Cancer	LibSVM	75	.79	77.77	80.88
	IBk	300	.74	77.74	77.25
	Boosted IBk	300	.70	72.39	72.83
	WeightedKDE	10	.69	71.65	67.80
	KDE	150	.67	67.38	67.00
	Boosted J48	100	.67	67.66	66.22
	KStar	50	.66	67.12	67.25
	J48	200	.65	66.71	71.29
	RandomForest	200	.65	66.09	71.20
Leukemia	IBk	100	.92	95.39	90.64
	LibSVM	150	.91	94.75	90.38
	KDE	30	.89	94.41	88.23
	WeightedKDE	30	.87	92.79	86.48
	Boosted IBk	30	.86	89.21	85.14
	RandomForest	30	.81	81.97	82.94
	Boosted J48	20	.81	84.03	82.57
	KStar	30	.80	81.66	81.56
	J48	20	.78	83.30	76.80
Lung Cancer	WeightedKDE	10	.70	74.04	68.43
	KDE	10	.68	73.60	68.10
	J48	10	.63	68.48	74.70
	IBk	10	.62	68.25	63.19
	Boosted IBk	10	.62	68.22	62.71
	KStar	10	.60	66.79	61.27
	RandomForest	10	.60	64.06	66.93
	LibSVM	250	.59	64.99	59.20
	Boosted J48	10	.58	63.00	67.40
Lymphoma	WeightedKDE	20	.83	89.11	82.14
	KDE	100	.82	86.68	84.47
	IBk	10	.82	83.57	81.09
	LibSVM	30	.82	83.57	80.66
	KStar	20	.79	82.02	78.24

Continued on Next Page. . .

Table A.5 – Continued

Phenotype	Classifier	Num Genes	F	Sensitivity	Specificity
	Boosted IBk	200	.79	81.44	80.67
	Boosted J48	100	.78	82.50	78.49
	J48	200	.77	81.37	77.26
	RandomForest	10	.75	77.80	76.25
Prostate Cancer	WeightedKDE	10	.90	92.90	89.86
	KDE	10	.88	92.65	88.88
	Boosted IBk	75	.88	88.20	94.85
	LibSVM	50	.86	86.39	92.43
	IBk	50	.84	83.36	90.73
	J48	10	.75	76.51	79.57
	RandomForest	10	.72	73.07	78.15
	Boosted J48	30	.72	72.70	76.66
	KStar	10	.70	73.44	73.17
Renal Cancer	IBk	10	1.00	100.00	100.00
	LibSVM	10	1.00	100.00	100.00
	Boosted IBk	10	1.00	100.00	100.00
	KDE	10	.99	99.48	99.20
	KStar	10	.98	97.89	99.20
	WeightedKDE	10	.97	98.11	96.67
	J48	20	.87	86.17	93.73
	Boosted J48	20	.87	86.17	93.73
	RandomForest	10	.86	84.12	95.46
Bone Marrow	LibSVM	75	.88	87.08	89.67
	KDE	10	.85	85.00	85.00
	WeightedKDE	10	.85	85.00	85.00
	IBk	75	.85	85.00	85.00
	Boosted IBk	75	.85	85.00	85.00
	RandomForest	75	.85	85.00	85.00
	Boosted J48	250	.84	84.94	83.33
	J48	50	.82	81.86	83.42
	KStar	10	.82	80.42	84.61
Brain Tissue	IBk	75	.95	95.00	95.00
	LibSVM	250	.95	95.00	95.00
	Boosted IBk	75	.95	94.83	94.88
	RandomForest	30	.94	94.23	94.76
	KStar	200	.94	93.48	94.30
	Boosted J48	250	.93	92.99	93.76
	J48	150	.92	93.52	91.75
	KDE	50	.92	93.26	91.06
	WeightedKDE	300	.92	93.26	90.91
Heart Tissue	LibSVM	250	.82	81.38	86.85
	RandomForest	10	.81	81.13	82.27

Continued on Next Page...

Table A.5 – Continued

Phenotype	Classifier	Num Genes	F	Sensitivity	Specificity
	Boosted IBk	20	.80	79.17	79.95
	KStar	50	.79	78.33	79.91
	IBk	20	.79	78.33	79.91
	Boosted J48	100	.78	76.26	84.07
	J48	10	.77	78.11	80.74
	KDE	50	.76	77.97	74.49
	WeightedKDE	10	.73	77.50	71.57
Liver Tissue	KDE	10	1.00	100.00	100.00
	WeightedKDE	10	1.00	100.00	100.00
	KStar	10	1.00	100.00	100.00
	J48	50	1.00	100.00	100.00
	IBk	10	1.00	100.00	100.00
	LibSVM	10	1.00	100.00	100.00
	Boosted IBk	30	1.00	100.00	100.00
	RandomForest	30	1.00	100.00	100.00
	Boosted J48	30	.98	99.67	97.62
Lung Tissue	IBk	10	.90	90.00	90.00
	LibSVM	10	.90	90.00	90.00
	Boosted IBk	10	.90	90.00	90.00
	KDE	20	.86	89.22	86.00
	KStar	20	.85	85.00	84.87
	RandomForest	20	.84	84.93	84.00
	J48	20	.84	85.00	83.53
	Boosted J48	10	.84	85.00	83.53
	WeightedKDE	20	.84	85.66	86.04
Muscle Tissue	RandomForest	10	.82	79.90	85.81
	LibSVM	10	.78	74.58	88.91
	IBk	10	.74	73.93	77.56
	Boosted IBk	10	.72	73.02	75.01
	KStar	10	.71	69.58	78.65
	Boosted J48	10	.71	70.00	73.64
	KDE	10	.69	70.29	73.75
	WeightedKDE	10	.67	67.19	67.29
	J48	10	.62	62.40	61.69
Pancreatic Tissue	KDE	10	.85	85.00	85.00
	WeightedKDE	10	.85	85.00	85.00
	KStar	10	.85	85.00	85.00
	J48	30	.85	85.00	85.00
	IBk	10	.85	85.00	85.00
	LibSVM	10	.85	85.00	85.00
	Boosted IBk	20	.85	85.00	85.00
	RandomForest	10	.85	85.00	85.00

Continued on Next Page...

Table A.5 – Continued

Phenotype	Classifier	Num Genes	F	Sensitivity	Specificity
	Boosted J48	30	.85	85.00	85.00
Prostate Tissue	LibSVM	20	.88	87.50	89.78
	RandomForest	20	.88	87.50	89.78
	IBk	20	.85	83.33	89.32
	Boosted IBk	20	.85	83.33	89.32
	Boosted J48	10	.80	79.08	83.76
	KDE	10	.78	79.35	88.16
	KStar	10	.78	77.50	79.62
	J48	20	.74	76.96	72.33
	WeightedKDE	10	.73	70.69	77.21
Renal Tissue	IBk	10	.95	94.44	94.89
	LibSVM	20	.95	94.44	94.89
	RandomForest	20	.95	94.44	94.89
	KDE	30	.95	94.77	94.55
	J48	10	.95	94.96	94.17
	Boosted J48	20	.94	93.00	95.00
	Boosted IBk	20	.93	91.94	94.75
	WeightedKDE	20	.92	93.43	92.29
	KStar	10	.92	89.89	94.64
Spinal Cord Tissue	LibSVM	10	.85	85.00	85.00
	Boosted IBk	20	.84	83.75	84.96
	KStar	10	.83	84.85	82.33
	IBk	20	.78	77.50	79.66
	RandomForest	10	.75	74.89	77.14
	J48	10	.69	69.93	68.18
	Boosted J48	10	.68	73.29	68.77
	KDE	20	.61	65.57	63.17
	WeightedKDE	10	.59	68.89	59.36
Thymus Tissue	RandomForest	50	.87	86.25	89.73
	LibSVM	30	.85	86.97	86.27
	KStar	30	.85	85.00	85.00
	J48	20	.85	85.00	85.00
	IBk	20	.85	85.00	85.00
	Boosted IBk	50	.85	85.00	85.00
	Boosted J48	10	.85	85.00	85.00
	WeightedKDE	20	.84	84.79	83.00
	KDE	20	.83	84.68	81.33

Bibliography

[1] Andersson A, Ritz C, Lindgren D, Eden P, Lassen C, Heldrup J, Olofsson T, Rade J, Fontes M, Porwit-Macdonald A, Behrendtz M, Hoeglund M, Johansson B, and Fioretos T. Microarray-based classification of a consecutive series of 121 childhood acute leukemias: prediction of leukemic and genetic subtype as well as of minimal residual disease status. *Leukemia*, 21:1198–1203, 2007.

[2] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[3] A. Aronson. Effective mapping of biomedical text to the UMLS metathesaurus: The MetaMap program. In *Proc AMIA 2001*, pages 17–21, 2001.

[4] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–29, May 2000.

[5] T. Barrett, D. B. Troup, S. E. Wilhite, P. Ledoux, D. Rudnev, C. Evangelista, I. F. Kim, A. Soboleva, M. Tomashevsky, and R. Edgar. NCBI GEO: mining tens of millions of expression profiles–database and tools update. *Nucleic Acids Res*, 35(Database issue), January 2007.

[6] Liang-Hua Bin, Larry D. Nielson, Xinqi Liu, Robert J. Mason, and Hong-Bing Shu. Identification of uteroglobin-related protein 1 and macrophage scavenger receptor with collagenous structure as a lung-specific ligand-receptor pair. *The Journal of Immunology*, 171:924–930, 2003.

[7] O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res*, 32(Database issue), January 2004.

[8] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[9] Atul J. Butte and Isaac S. Kohane. Creation and implications of a phenome-genome network. *Nature Biotechnology*, 24(1):55–62, January 2006.

[10] Chih-Chung Chang and Chih-Jen Lin. LIBSVM - a library for support vector machines, 2001. The Weka classifier works with version 2.82 of LIBSVM.

[11] R. Chen, L. Li, and A.J. Butte. AILUN: reannotating gene expression data automatically. *Nat Methods*, 4(11):879, 2007.

[12] Sung-Bae Cho and Hong-Hee Won. Machine learning in DNA microarray analysis for cancer classification. In *APBC '03: Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics 2003*, pages 189–198, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[13] Priscilla M. Clarkson, Joseph M. Devaney, Heather Gordish-Dressman, Paul D. Thompson, Monica J. Hubal, Maria Urso, Thomas B. Price, Theodore J. Angelopoulos, Paul M. Gordon, Niall M. Moyna, Linda S. Pescatello, Paul S. Visich, Robert F. Zoeller, Richard L. Seip, , and Eric P. Hoffman. ACTN3 genotype is associated with increases in muscle strength in response to resistance training in women. *Journal of Applied Physiology*, 99:154–163, 2005.

[14] John G. Cleary and Leonard E. Trigg. K*: An instance-based learner using an entropic distance measure. In *12th International Conference on Machine Learning*, pages 108–114, 1995.

[15] Ramón Díaz-Uriarte and Sara Alvarez de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3, 2006.

[16] Slater DJ, Hilgenfeld E, Rappaport EF, Shah N, Meek RG, Williams WR, Lovett BD, Osheroff N, Autar RS, Ried T, and Felix CA. MLL-SEPTIN6 fusion recurs in novel translocation of chromosomes 3, X, and 11 in infant acute myelomonocytic leukaemia and in t(X;11) in infant acute myeloid leukaemia, and MLL genomic breakpoint in complex MLL-SEPTIN6 rearrangement is a DNA topoisomerase ii cleavage site. *Oncogene*, 21:4706–4714, 2002.

[17] Joel Dudley and Atul J. Butte. Enabling integrative genomic analysis of high-impact human diseases through text mining. In *Pacific Symposium on Biocomputing*, pages 580–591. Morgan Kaufmann, 2008.

[18] R. Edgar, M. Domrachev, and A.E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res*, 30(1):207–10, 2002.

[19] David Edwards. Non-linear normalization and background correction in one-channel cDNA microarray studies. *Bioinformatics*, 19:825–833, 2003.

[20] Yasser EL-Manzalawy. WLSVM, 2005. http://www.cs.iastate.edu/~yasser/wlsvm/.

[21] Nabil A. Elshourbagy, Xiaotong Li, John Terrett, Stephanie VanHorn, Mitchell S. Gross, John E. Adamou, Karen M. Anderson, Christine L. Webb, and Paul G. Lysko. Molecular characterization of a human scavenger receptor, human MARCO. *European Journal of Biochemistry*, 267(3):919–926, 2000.

[22] S. P. Fodor, J. L. Read, M. C. Pirrung, L. Stryer, A. T. Lu, and D. Solas. Light-directed, spatially addressable parallel chemical synthesis. *Science*, 251(4995):767–773, February 1991.

[23] T. S. Furey, N. Christianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Hauessler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[24] J Hiraga, A Katsumi, T Iwasaki, A Abe, H Kiyoi, T Matsushita, T Kinoshita, and T Naoe. Prognostic analysis of aberrant somatic hypermutation of rhoh gene in diffuse large b cell lymphoma. *Leukemia*, 21(8):18461847, 2007.

[25] G. Hripcsak and A. S. Rothschild. Agreement, the F-measure, and Reliability in Information Retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.

[26] Kyu Baek Hwang, Sek Won Kong, Steve A. Greenberg, and Peter J. Park. Combining gene expression data from different generations of oligonucleotide arrays. *BMC Bioinformatics*, 5:159, 2004.

[27] Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In Derek H. Sleeman and Peter Edwards, editors, *Ninth International Workshop on Machine Learning*, pages 249–256. Morgan Kaufmann, 1992.

[28] Isaac S. Kohane, Atul J. Butte, and Alvin Kho. *Microarrays for an Integrative Genomics.* MIT Press, Cambridge, MA, USA, 2002.

[29] Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In Francesco Bergadano and Luc De Raedt, editors, *European Conference on Machine Learning*, pages 171–182. Springer, 1994.

[30] Matthew C. Kostek, Yi-Wen Chen, Daniel J. Cuthbertson, Rongye Shi, Mark J. Fedele, Karyn A. Esser, and Michael J. Rennie. Gene expression responses over 24 h to lengthening and shortening contractions in human muscle: major changes in CSRP3, MUSTN1, SIX1, and FBXO32. *Physiological Genomics*, 31:42–52, 2007.

[31] Pierre Legendre and Daniel Borcard. Statistical comparison of univariate tests of homogeneity of variances. Submitted for publication, 2007.

[32] Leping Li, Clarice R. Weinberg, Thomas A. Darden, and Lee G. Pedersen. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, 17(12):1131–1142, 2001.

[33] NCBI. NCBI website, December 2007. http://www.ncbi.nlm.nih.gov/.

[34] Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[35] Shyamsundar R, Kim YH, Higgins JP, Montgomery K, Sethuraman A Jorden M, van de Rijn M, Botstein D, Brown PO, and Pollack JR. A DNA microarray survey of gene expression in normal human tissues. *Genome Biology*, 6, 2005.

[36] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2007. ISBN 3-900051-07-0.

[37] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc Natl Acad Sci U S A*, 98(26):15149–15154, December 2001.

[38] Marko Robnik-Sikonja and Igor Kononenko. An adaptation of relief for attribute estimation in regression. In Douglas H. Fisher, editor, *Fourteenth International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, 1997.

[39] Mark Schena, Dari Shalon, Ronald W. Davis, and Patrick O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–470, October 1995.

[40] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics).* Wiley-Interscience, September 1992.

[41] E. Segal, N. Friedman, D. Koller, and A. Regev. A module map showing conditional activity of expression modules in cancer. *Nature Genetics*, 36(3):1090–1098, September 2004.

[42] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall/CRC, April 1986.

[43] A Traverse-Glehen, A Verney, L Baseggio, P Felman, E Callet-Bauchu, C Thieblemont, M Ffrench, J-P Magaud, B Coiffier, F Berger, and G Salles. Analysis of BCL-6, CD95, PIM1, RHO/TTF and PAX5 mutations in splenic and nodal marginal zone B-cell lymphomas suggests a particular b-cell origin. *Leukemia*, 21(8):18211824, 2007.

[44] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. CAPTCHA: Using hard ai problems for security. In *EUROCRYPT*, pages 294–311, 2003.

[45] Patrick Warnat, Roland Eils, and Benedikt Brors. Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, 6(1):265, 2005.

[46] Ge X, Yamamoto S, Tsutsumi S, Midorikawa Y, Ihara S, Wang SM, and Aburatani H. Interpreting expression profiles of cancers by genome-wide survey of breadth of expression in normal tissues. *Genomics*, 86:127–141, 2005.

[47] Kun Yang, Jianzhong Li, and Hong Gao. The impact of sample imbalance on identifying differentially expressed genes. *BMC Bioinformatics*, 7:S8, 2006.