Distance Measures in Bioinformatics

A Thesis

Submitted to the Faculty

of

Drexel University

by

Feiyu Xiong

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy in Electrical Engineering

January 2015

Table of Contents

List of Tables

List of Figures

Abstract

Distance Measures in Bioinformatics

Feiyu Xiong

Advisor: Drs. Moshe Kam and Leonid Hrebien, Ph.D.

Many bioinformatics applications rely on the computation of similarities between objects. Distance and similarity measures applied to vectors of characteristics are essential to problems such as classification, clustering and information retrieval.

This study explores the usefulness of distance and similarity measures in several bioinformatics applications. These applications are in two categories.

(1) Estimation of the adverse reaction severity of unknown pharmaceutical treatments, based on the severity of known treatments, in order to provide guidance for testing of the unknown treatments in clinical trials.

(2) Classification of cancer tissue types and estimation of cancer stages, based on high-dimensional microarray data, in order to support clinical decisions making.

To address the first category, we studied several clustering and classification approaches for binary severity estimation of Cytokine Release Syndrome (CRS). We developed a Severity Estimation using Distance Metric Learning (SE-DML) approach to get graded severity estimation. With binary estimation we were able to identify treatments that caused the most severe response and then built prediction models for CRS. Using the SE-DML approach, we evaluated four known data sets and showed that SE-DML outperformed other widely used methods on these data sets.

For the second category, we presented Kernelized Information-Theoretic Metric Learning (KITML) algorithms that optimize distance metrics and effectively handle high-dimensional data. This learned metric by KITML is used to improve the performance of k-nearest neighbor classification for cancer tissue microarray data. We

evaluated our approach on fourteen (14) cancer microarray data sets and compared our results with other state-of-the-art approaches. We achieved the best overall performance for the classification task. In addition we tested the KITML algorithm in estimating the severity stages of cancer samples, with accurate results.

1. Introduction

1.1 Motivation and Overview

Machine learning tasks involve the comparison of data samples in term of some distance/similarity measures. These measures are essential to classification, clustering and information retrieval tasks [4]. In bioinformatics, the concept of similarity is fundamental to the study of macromolecular structures, genomes, proteomes and metabolic pathways. For example, to determine whether a test treatment will have similar adverse reaction as a known treatment, it is common to measure the similarity between vectors of characteristics of samples of both treatments.

Let $\mathbf{X}$ be a set of data points. A distance/similarity measure on $\mathbf{X}$ is a function $d : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$. For all x, y, z in $\mathbf{X}$, this function is required to satisfy the following conditions [5]:

- $d(x, y) \geq 0$ (non-negativity)
- $d(x, y) = 0$, if and only if $x = y$ (coincidence)
- $d(x, y) = d(y, x)$ (symmetry)
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality).

There are many distance/similarity measures in use, including the Euclidean distance, Mahalanobis distance and Pearson correlation. How to choose and use these measures is important in practical applications. For example, studies show that simple nearest neighbor methods work if an appropriate distance measure is chosen [6, 7, 8, 9]. Clustering algorithms such as K-means clustering also rely on the pairwise distance measurements between samples [10] and the right choice of metric makes significant difference in the accuracy of the analyses.

In this study, we develop distance/similarity measures to address two kinds of problems. The first problem type is severity estimation, the estimation of disease states or adverse-reactions to a treatment (drug, regiment, behavior modification, etc.). There are many reasons to study severity estimation including the need to understand the stage of a condition/disease; to match a treatment to the severity at which a condition is manifested; and to track the progression of a condition/disease. Although researchers have developed diagnostic scores for predicting disease states and clinical outcomes [11, 12], the process of determining the scores is time consuming and expensive [13, 14]. To address this issue, we first perform binary severity estimation, e.g., determining whether or not a treatment will have a similar adverse-reaction severity to that of a known treatment. Next we apply a Severity Estimation using Distance Metric Learning approach (SE-DML). This is a generalized approach that provides quantitative severity determination that is applicable for several areas in bioinformatics. Our binary severity estimation is evaluated on Cytokine Release Syndrome (CRS) data. The SE-DML approach is evaluated on several data sets, including CRS data, Cardiotocography (CTG) data, and two Quantitative Structure Activity Relationship (QSAR) data sets.

The second problem we studied is cancer tissue classification using high-dimensional molecular profiling data. Recent advances in molecular profiling technologies have enabled researchers to query the expression values of thousands of genes simultaneously. Information derived from such genome-wide molecular profiling is important in the diagnosis and identification of cancer tissue types in patient samples [15, 16]. An important emerging medical application domain for microarray technologies is clinical decision support in the form of diagnosis of disease as well as the prediction of clinical outcomes in response to treatments [16]. When mining molecular signature data, the process of comparing samples through an adaptive distance function is fundamental

but difficult, as such data sets are normally heterogeneous and high dimensional. In this thesis, we present Kernelized Information-Theoretic Metric Learning (KITML) algorithms that optimize a distance function to tackle the cancer tissue classification problem. We study two applications of KITML using high-dimensional cancer molecular profiling data. (1) for sample-level cancer tissue classification, the learned metric is used to improve the performance of k-nearest neighbor classification. (2) for estimating the severity level or stage of a group of samples, we propose a set-based ranking approach to extend KITML.

1.2 Cytokine Release Syndrome (CRS)

1.2.1 Overview

Monoclonal antibodies (mAbs) are widely used in anti-inflammatory and tumor therapy, but can cause a variety of adverse side effects [17]. One of these is Cytokine Release Syndrome (CRS), which is characterized by the systemic release of several inflammatory mediators which set off a cascade release of cytokines [18]. Symptoms of CRS can include fatigue, headache, urticaria, pruritus, bronchospasm, dyspnea, sensation of tongue or throat swelling, rhinitis, nausea, vomiting, flushing, fever, chills, hypotension, tachycardia and asthenia [19]. Some CRS reactions are mild to moderate in severity and can be controlled by slowing the infusion rate of the mAb or by administering anti-inflammatory drugs [18]. However in a 2006 phase I clinical trial using Anti-CD28 SA mAb (TGN 1412), the reactions were much more severe and six healthy volunteers developed severe CRS within 90 minutes of receiving a dose of Anti-CD28 SA [20].

Prior to the 2006 trial, Anti-CD28 SA was tested on non-human primates and rodents to determine the potential for CRS [21]. Although release of cytokines has been observed in animal models, rarely has it progressed to clinically relevant lev-

els [22, 23, 24]. Differences in expression of target molecules, regulatory T cells, cytokines required for inflammatory response, and cell surface receptors among humans, rodents and non-human primates [25, 26, 27, 28] all indicate that it may not be appropriate to use animal models to predict CRS in humans.

As a result of this, to further understand CRS in humans, an in vitro assay using human whole blood was developed and tested by Walker et al. [18]. This assay was designed to support First-In-Human readiness of mAb treatments assessing the potential for mAbs to release of cytokines similar to Anti-CD28 SA reaction. The studies reported in this thesis use results from this assay for further analysis of CRS using several machine learning approaches.

1.2.2 CRS Problem Definition

The onset of CRS is an important consideration in drug development. Researchers have applied different machine learning approaches to CRS data from different assays [29, 30, 31]. However, the analysis of data has been limited to 1-3 cytokines at a time and simultaneous multi-dimensional comparisons across a greater number of cytokines is not common [18]. In this thesis, we apply three (3) machine learning approaches in combination to multi-dimensional data (12 cytokines) obtained from Walker's in vitro assay [18]. These machine learning approaches are (i) Hierarchical Cluster Analysis (HCA); (ii) Principal Component Analysis (PCA) followed by K-means clustering; and (iii) Decision Tree Classification (DTC). We try to assess the potential of mAb-based therapeutics to produce cytokine release similar to that induced by Anti-CD28 SA. In addition, we apply distance metric learning algorithms to develop a severity estimation approach that is used to give a more graded severity levels for different mAb treatments.

1.3 Cardiotocography

1.3.1 Overview

Cardiotocography (CTG), also known as electronic fetal monitoring, is the most widely used tool for fetal surveillance during pregnancy. CTG monitors changes in fetal heart rate (FHR) and uterine contractions (UC) during pregnancy [32], and identifies the occur of fetal hypoxia (short of oxygen). Fetal hypoxia may result in long term disability or even death during delivery [33]. Therefore, efficient and effective diagnosis on fetal hypoxia is an important issue.

In obstetrics, CTG provides measurements through either external or internal methods. In the external method, the FHR and UC are detected by two transducers placed on the mother's abdomen. A Doppler ultrasound traducer provides FHR information and a pressure transducer provides UC data which is recorded on a paper strip [32]. In the internal method, a catheter is placed in the uterus after a specific amount of dilation has taken place and provides a more accurate and consistent transmission of the FHR and UC than external monitoring because factors such as movement do not affect measurement [34]. CTG shows fetal development and health information, especially the maturation status of autonomous nervous system [34].

1.3.2 Cardiotocography Problem Definition

Cardiotocography (CTG) is used to evaluate fetal well-being during delivery. In general, average of FHR, change of FHR, acceleration and deceleration of FHR and fetal movement are essential parameters on medical diagnosis of fetal hypoxia [35]. Many researchers have been working on different methods to interpret the CTG data for fetal hypoxia in order to help physicians make clinical decisions [36]. Our problem here is to analyze a CTG data set consisting of 2126 samples from University of California-Irvine (UCI) machine learning repository [37]. The data set was classified

by three expert obstetricians and consensus classification label was assigned to each sample indicating the status of fetal hypoxia. The goal is to use distance metric learning approach to build estimation models and to determine the severity of fetal hypoxia based on the features of the samples.

1.4 Quantitative Structure Activity Relationship (QSAR)

1.4.1 Overview

Quantitative Structure Activity Relationships (QSAR) describe the interaction of chemical compounds with biological systems making it possible to predict the activities/properties of a given compound as a function of its molecular descriptors. These relationships are essential to toxicological investigations in the development of pharmaceutical compounds. Biological reactions to new compounds are often inferred from properties of similar materials whose hazards are already known [38]. During the development of new pharmaceutical compounds, such chemicals need to be evaluated in different biological media where both in vitro and in vivo testing is very costly and time consuming [39]. In addition, current trends are toward improved understanding of the chemical mechanisms of toxicological endpoints and consolidation of toxicological data into databases [38].

1.4.2 Quantitative Structure Activity Relationship (QSAR) Model Development

The construction of QSAR models is a two step process. The first step is generating the description of the molecular structure. The second step is multivariate analysis for correlating molecular descriptors with observed activities/properties. The model development process is shown in the Figure 1.1, as described by Nantasenamat et al. [2]. The process starts with observation of the molecular descriptors of the

data, which are its physiochemical properties. These include electronic, geometrical, hydrophobic, lipophilicity, solubility, steric, quantum chemical, and topological properties [2]. Multivariate analysis in this modeling process is the application of machine learning techniques to discover the relationships between molecular descriptors and the biological/chemical properties of interest.

Figure 1.1: Schematic overview of the quantitative structure activity relationship model development process [2]

1.4.3 QSAR Problem Definition

Quantitative structure activity relationships are widely used in drug development. The molecular descriptors of compounds are from experimental results such as bioactivity assays. A computational QSAR model is built from such compounds so that the model learns/captures the structural properties of the compounds that are causally related to their bioactivity [40]. Therefore, QSAR models are commonly formulated as supervised machine learning problems and researchers have applied different supervised classification approaches such as support vector machine [41, 42] and artificial neural network [43, 44] to analyze QSAR data. In this thesis, we try to determine the severity of the two families of chemical compounds –Pyrimidines and Triazines [3]– based on their structural properties. We proposed our distance metric learning approach for severity estimation which uses a numerical value (between 0 to 1) to represent the level of severity for the chemical compounds.

1.5 Applications of High-dimensional Molecular Profiling Data to Cancer Tissue Classification

1.5.1 Overview

With the advancement of genome-wide monitoring technologies, molecular expression data have become widely used for diagnosing cancer using tumor or blood samples. When mining molecular signature data, the process of comparing samples through an adaptive distance function is fundamental but difficult, as such data sets are normally heterogeneous and high dimensional. In this thesis, we focus on applying distance metric learning algorithms on the molecular signatures of patient samples from microarray analysis as well as reducing the computational load when dealing with high dimensional molecular expression data.

1.5.2 High-dimensional Molecular Profiling Data Problem Definition

Machine learning techniques such as classification and clustering are used for analysis and interpretation of data obtained from molecular profiling measurements [16, 45]. These data are characterized by a high number of measured variables (m genes) over a relatively small number of observations (n samples). The number of genes in a single sample is typically in the thousands and the number of samples is typically in the hundreds, so the number of feature variables (genes) greatly exceeds the number of samples. This situation ($m >> n$) has "high dimensionality" [46] and makes the application of machine learning techniques challenging. For example, recent studies have tried to tackle the "high-dimensionality" issue when predicting the existence of cancer using molecular expressions through sparse-learning based approaches [47]. As molecular signature data become available for more and more patient samples (e.g. from the national project The Cancer Genome Atlas (TCGA) [48]), measuring the similarity among patient samples becomes critical for mining such signature

data. Such similarity measures can be used for molecular signature-based retrieval of similar cancer patient cases when treating a target patient.

In this thesis, we have designed an accurate cancer classification algorithm based on an extension of "Information-Theoretic Metric learning (ITML)" techniques [49, 50] that is able to provide good assessments of patient similarity, where previous attempts [45, 47] did not fully succeed due to the "curse-of-dimensionality" [46]. Having been studied over the past few years [10, 10, 49, 51, 52], distance metric learning has been applied to practical areas like image recognition [51] and information retrieval [53]. This thesis presents two novel extensions of metric learning for the tasks of sample-level tissue classification and group-level cancer stage determination. The issue of "small sample, large feature" is addressed through "kernelizing" the learned metric from ITML or Kernelized Information-Theoretic Metric Learning (KITML). By learning a nonlinear transformation in the input space implicitly through kernelization, KITML permits efficient optimization and improved learning of a distance metric. Our two applications of KITML using high-dimensional molecular profiling data are (1) improving the performance of K-Nearest Neighbor (KNN) classification for cancer tissue classification and (2) estimating the severity level or stage of a group of samples.

1.6 Thesis Structure

In Chapter 2, we introduce binary severity estimation for CRS using 1)Hierarchical Clustering Analysis (HCA); 2) Principal Component Analysis (PCA) followed by K-means clustering [54]; and 3) Decision Tree Classification (DTC) [55] to determine whether a test treatment will have a similar adverse-reaction severity to that of Anti-CD28 SA. In Chapter 3 we go beyond binary severity estimation using distance metric learning algorithms which allow us to determine the range of the response

severities. Here we studied CRS, CTG and two QSAR data sets whose dimensionality was relatively low. The specific algorithm used in Chapter 2 is ITML. In order to apply the approach to high dimensional data, which would require high computational cost, we used, instead, a kernel function to make the distance metric learning more computationally efficient. In Chapter 4, we apply this Kernelized Information-Theoretic metric Learning (KITML) algorithm to high dimensional microarray data sets for cancer issue classification. The results for each analysis are given, and the advantages and drawbacks for each approach are discussed in Chapter 5. The last chapter presents the conclusion and future work of this thesis.

2. Binary Severity Estimation for Cytokine Release Syndrome

Binary severity estimation, applied to data of the in vitro assay developed by Walker et al. [18], determines whether or not the severity of CRS due to a test treatment is similar to that of Anti-CD28 SA. Walker's in vitro assay is described first. Several machine learning algorithms were used here for binary severity estimation of the assay data. They are: 1)Hierarchical Clustering Analysis (HCA); 2) Principal Component Analysis (PCA) followed by K-means clustering [54]; and 3) Decision Tree Classification (DTC) [55]. A comparison of the utility of these approaches for the analysis of the assay is also presented.

2.1 In Vitro Assay Description and Data Set

For Walker's assay, blood was drawn aseptically under informed consent[1] by venipuncture using a 21-gauge needle from 44 normal human volunteers into BD Heparin Vacutainer (San Jose, CA) tubes. Cultures were set up within two (2) hours of blood collection. Previous reports on these types of assays suggest the need to immobilize Anti-CD28 for maximal cytokine production [21]. For this purpose, Protein A coated polystyrene beads were selected. Beads were coated with a saturating amount of mAb and then distributed to a 96-well culture dish. Each well contained 1×107 beads/well along with $200\mu l$ of 1:10 diluted whole blood in RPMI 1640 media.

The cultures were incubated at 37 °C for 48 hours. Following incubation, wells were resuspended and centrifuges at 2500 rpm for 5 min. Supernatant was removed, transferred to shipping plates, and supernatant plates were frozen at −80 °C. Plates were shipped on dry ice for multiplex analysis.

We used the assay to test the stimulation of human blood from different donors

[1]Quorum Review IRB approved protocol #NOCOMPOUNDNAP1001

Table 2.1: List of cytokines release measured in the assay for binary severity estimation

Cytokine	Mean (pg/ml)	Median(pg/ml)	Maximum (pg/ml)
IL-1β	240.8	140.4	1932.3
IL-2	12.0	2.3	406.2
IL-4	2.2	0.5	45.2
IL-6	4897.6	1620.7	39195.5
IL-10	6.3	2.7	100.8
IL-12(p70)	8.3	5.4	55.5
IL-17	22.1	0.1	263.2
IL-18	13.0	10.6	71.1
IFN-γ	4661.2	10.4	90400.57
TNF-α(monometric)	433.8	178.1	3729.62
TNF-α(trimetric)	294.4	131.8	1809.5

where the application of a given treatment (monoclonal antibody (mAb)) on blood from a particular donor constituted a sample. The concentrations of the 11 cytokines shown in Table 2.1 were measured for each sample. These concentrations were measured in triplicate by multiplex enzyme-linked immunosorbent assay (ELISA) using SearchLight^{TM} technology from Aushon Biosystems (Billerica, MA). Data were reported in pg/ml for each sample and each cytokine. To allow calculation of mean values and graphic analysis, all concentrations below the level of quantitation were set to 0.1 [18]. The mean, median, and maximum values of each cytokine for all the samples are also shown in Table 2.1.

The 7 mAbs and 2 controls used in our study are described in Table 2.2, which shows the target, manufacturer, number of samples, expected results and class for each mAb. The "Expected Results" column in Table 2 is based on the clinical literature (Tocilizumab and Palivizumab) and on the mechanism of action of the research grade mAb being similar to a compound that has clinical results (for Anti-CD28 SA, Anti-CD80, Anti-CD22, Anti-IL-1, or Anti-IL-5) [56]. The "class" column is based on the expected reaction where severe CRS is caused by Anti-CD28 SA; no infusion reactions

have been reported for the remaining treatments. The data were thus grouped into two categories, "CD28" and "Safe." The "CD28" class contained samples only from cultures treated with Anti-CD28 SA. The "Safe" class contained mAbs that are not likely to cause CRS or an infusion reaction, and controls.

Table 2.2: List of mAbs and controls used in our CRS detection study

mAb/clone no.	Target	Manufacturer	Samples	Expected Results	Class
Anti-CD28 SA / ANC28.1/5D10	CD28	Ancell	152	Severe CRS	CD28
Anti-CD80 / 2D10	CD80	Abcam	8	No CRS or Infusion Reactions	Safe
Anti-CD22 / LT22	CD22	Abcam	8	No CRS or Infusion reactions	Safe
Anti-IL-1β / 2805	IL-1β	R&D Systems	8	No CRS or Infusion reactions	Safe
Anti-IL-5 / QS-5	IL-5	Abcam	8	No CRS or Infusion reactions	Safe
Tocilizumab	IL-6 Receptor	Roche	8	No CRS or Infusion Reactions	Safe
Palivizumab	RSV Fusion	Medimmune	8	No CRS or Infusion Reactions	Safe
PBS	(Control)	-	80	No CRS or Infusion reactions	Safe
AutoPlasma	(Control)	-	152	No CRS or Infusion reactions	Safe

The dataset analyzed in this thesis contains a total of 432 samples that were measured through 11 runs of the assay. The information for each run is shown in Table 2.3, including donors in each run, treatments used in each run, number of samples per treatment, and total number of samples for each run. The sizes of sample sets corresponding to different treatments are uneven, an observation that would affect the performance of subsequent analyses.

Table 2.3: Donor information for 11 runs in the data set

Run ID	Donor IDs	Treatments Used	Sample number per Treatment	Total Samples
Run 1	donor #5, donor #9, donor #30, donor #36	PBS AutoPlasma Anti-CD28 SA CD80 CD22	4 4 4 4 4	20
Run 2	donor #1, donor #4, donor #27, donor #29	PBS AutoPlasma Anti-CD28 SA CD80 CD22	4 4 4 4 4	20
Run 3	donor #2, donor #8, donor #10, donor #21, donor #22, donor #23, donor #25, donor #40	PBS AutoPlasma Anti-CD28 SA IL-1β IL-5	8 8 8 8 8	40
Run 4	donor #1, donor #5, donor #7, donor #9, donor #10, donor #12, donor #19, donor #25	PBS AutoPlasma Anti-CD28 SA	8 8 8	24
Run 5	donor #1, donor #5, donor #10, donor #25, donor #36, donor #37, donor #39, donor #40	PBS AutoPlasma Anti-CD28 SA	8 16 16	40
Run 6	donor #1, donor #7, donor #10, donor #13, donor #19, donor #25, donor #37, donor #39	PBS AutoPlasma Anti-CD28 SA	8 24 24	56
Run 7	donor #6, donor #12, donor #20, donor #21, donor #28, donor #34, donor #37, donor #38	PBS AutoPlasma Anti-CD28 SA	8 8 8	24
Run 8	donor #13, donor #14 donor #17, donor #19, donor #20, donor #24, donor #26, donor #41	PBS AutoPlasma Anti-CD28 SA Tocilizumab Palivizumab	8 8 8 8 8	40
Run 9	donor #11, donor #19, donor #15, donor #31, donor #32, donor #33, donor #41, donor #43	PBS AutoPlasma Anti-CD28 SA	8 24 24	56
Run 10	donor #11, donor #19, donor #15, donor #31, donor #32, donor #33, donor #41, donor #43	PBS AutoPlasma Anti-CD28 SA	8 24 24	56
Run 11	donor #3, donor #16, donor #17, donor #18, donor #35, donor #41, donor #42, donor #44	PBS AutoPlasma Anti-CD28 SA	8 24 24	56

2.2 Hierarchical Clustering Analysis

The first algorithm we used to analyze the treatments of the in vitro assay is agglomerative HCA, implemented in Matlab® 2012a software (The Mathworks, Natick, MA). Agglomerative HCA was applied to means of the cytokine samples from each one of the Table 2.2 treatments. It is a "bottom up" approach which first considers each treatment as being in its own cluster and then merges pairs of clusters by their distances from each other. The process repeats until all treatments are within one cluster. Each treatment was evaluated by using an unweighted group mean with Euclidean distance as the similarity measurement. The Euclidean distance, d_{ab}, between two means, m_a and m_b of treatments a and b is defined as $d_{ab} = \sqrt{(m_a - m_b)(m_b - m_a)}$.

2.3 Principal Component Analysis

Principal Component Analysis (PCA) is an algorithm commonly used to reduce the number of attributes used to represent a set of data. PCA transforms the original data (which may be given as a function of correlated variables) into linearly uncorrelated attributes, by projecting the original data onto orthogonal components such that the variance of the projected data is maximized [54]. These orthogonal components are obtained by using singular value decomposition of the covariance matrix Σ associated with the data [57]. The covariance matrix of a vector of random variables X, is defined as [57]:

$$Cov(X) = E[(X - E[X])(X - E[X])^T]. \tag{2.1}$$

We can consider the samples for each attribute as a column vector of random variables. Hence, we can assemble a matrix M where each row represents one sample and each column is the difference between one of the attributes and its expected

value. For our data, there are 11 columns representing 11 cytokine attributes, so M is of the form

$$M = [(X_1 - E[X_1]) \vdots (X_2 - E[X_2]) \vdots \cdots \cdots \vdots (X_{11} - E[X_{11}])]. \tag{2.2}$$

The covariance matrix can be estimated from the matrix M as the sample covariance matrix Σ,

$$\Sigma = \frac{1}{n} M^T M. \tag{2.3}$$

The eigenvectors of the covariance matrix Σ are known to characterize the orthogonal components (Principal Components). The eigenvalues of Σ are equal to the variances associated with each Principal Component [54]. PCA was applied to reduce the number of attributes used to represent the data. We calculated the variance associated with each of the Principal Components and chose the components with the largest variances. We ignored the Principal Components that accounted for small amount of variance.

The best results from PCA are obtained for datasets whose attributes have similar dynamic ranges [54]. In our case, the data vary greatly, as seen in column 4 of Table 2.1, so we applied a logarithmic transformation on the data before proceeding with the analysis. This transformation may be used when the attributes show a linear or nearly-linear relationship between the standard deviation and the mean for each treatment [58][59], which our data do exhibit (as shown in Chapter 5).

2.4 K-means Clustering

K-means clustering, which we applied to the data after PCA, assigns the n observations in a dataset into k clusters. Each observation is assigned to the cluster

whose mean is the nearest to that observation. The standard K-means clustering algorithm is based on alternating two procedures [60]. The first procedure is the assignment step, which assigns each observation to the cluster whose mean yields the least within-cluster sum of distances. The second procedure is the calculation of new cluster means based on the assignments. The process stops when no reassignment of an observation to a cluster would minimize the within-cluster sum of distances between samples and the mean of each cluster. The Euclidean measure was used here to calculate the distance between observations.

2.5 Decision Tree Classification (DTC)

The C4.5 DTC algorithm [55] implemented in the Weka 3.6.6 software [61] (University of Waikato, Hamilton, New Zealand) was also used to analyze the dataset. This is a supervised machine learning algorithm, meaning that a correctly-labeled data set is required to "train" the algorithm before the algorithm can be applied to unknown data. Each observation in the dataset is defined by a collection of measured attributes (in our case cytokine levels) and a corresponding group or class label. The algorithm defines a set of rules that assign each observation to a corresponding class. The input to the algorithm is the collection of attributes for a given observation, and the output is the assigned class for that observation.

The DTC algorithm uses training data to infer rules that describe the correspondence between input attributes and the classes into which they are associated. The algorithm applies a "divide and conquer" approach, resulting in an iterative process that starts by analyzing each attribute from the training data separately from the others. It calculates the information gain for each attribute with respect to the possible class outcomes present in the training data [55]. The attribute with the highest information gain is denoted as the "root node." This attribute is used to make the

first separation of the data samples in a process called "branching" that assigns a "branch" to each sample according to the attribute value present in the sample.

After the first branching operation, the samples belonging to a specific branch may be associated with a class if all the samples in the branch have the same class label. In this case, the algorithm is said to have researched a "leaf node" denoted by the label associated with the corresponding class. If the samples in the branch have different class labels, the sample are analyzed further to find the attribute with the highest information gain. This process is repeated until all observations in the dataset are assigned a class.

Since the attributes in our dataset have continuous values, it was necessary for the algorithm to define thresholds that separate the possible attribute values into discrete groups which can then be associated with a corresponding class. Each threshold is chosen by iteratively calculating information gains for certain threshold candidates in the data from the attribute being analyzed (the method for generating threshold candidates is described in [62], Section 6.1). The candidate with the highest information gain is used to split the values of the attributes and build the Decision Tree.

The rules created by the algorithm are displayed using a tree structure consisting of test nodes and branches. Each node represents the testing of a rule applied to a certain attribute. The branches represent the possible outcomes from the test, and point to either a class label or to another node for further testing. The top node in the tree is the "root node," which represents the testing of the most relevant attribute obtained by the algorithm. The class assigned for a given sample is denoted by a "leaf node," which is located at the bottom of the tree and contains the label assigned to a given sample.

The DTC algorithm analyzes all the attributes in the training set and selects the best attribute that maximizes the information gain as the root node. This attribute

is considered the most relevant for classification among all attributes [55]. The second most relevant attribute is found by removing the attribute in the root node and applying the algorithm to the remaining attributes [63]. The new root node is considered the second most relevant attribute for classification. This process is continued until all attributes were considered.

2.6 The DTC Model Construction (Definition of Training Data Sets and Test Data Set)

The DTC model is constructed using the data set shown in Table 2.3. In order to build and validate the model, we divided the data into training data and test data. We defined two training data sets using some or all of the 264 samples from 32 donors in runs 1-8 shown in Table 2.3. The remaining 168 samples from 15 donors in runs 9-11, were reserved as test data set. Before applying the test data from runs 9-11 we ran cross validations with the two training data sets to determine the accuracy of our models.

Training data set 1, consisting of 216 samples shown in Table 2.4, was assembled using only the control samples for the "Safe" class and the Anti-CD28 SA samples for the "CD28" class. The use of these data for the training set is expected to maximize the difference between samples from the two classes. For training data set 2, consisting of the 264 samples shown in Table 2.4, we used additional mAbs in the "Safe" class. The "Safe" class included Tocilizumab [64] and Palivizumab [65], two mAbs that were tested clinically and have shown no CRS reactions, and four (4) research-grade compounds that were not tested clinically but have the same target as mAbs that were tested clinically. The four research-grade compounds are assumed - like their clinical counterparts - not to cause CRS reactions. They were: 1) CD80 which may be similar to Galiximab [66]; 2) CD22 which is similar to Epratuzumab [66]; 3) IL-1β which is

similar to Canakinumab [67]; and 4) IL-5 which is similar to Mepolizumab [68]. The controls, PBS and AutoPlasma, were also included in the "Safe" class. The test data set was selected from the 168 samples in runs 9-11 shown in Table 2.3. Since there are 3 common donors (donor #17, donor #19 and donor #41) between runs 1-8 and runs 9-11, the 42 samples of these three donors were removed from the test data set, leaving 126 samples in the test data set, as shown in Table 2.6. The test data samples therefore came from different donors than those used in training data sets 1 and 2, and could be used to assess the ability of the DTC models to classify new data.

Table 2.4: Training data set 1 for DTC

Class	MAbs and controls	# samples
Safe	PBS(56) AutoPlasma(80)	136
CD28	Anti-CD28 SA(80)	80

Table 2.5: Training data set 2 for DTC

Class	MAbs and controls	# samples
Safe	PBS(56) AutoPlasma(80) Tocilizumab (8) Palivizumab(8) CD80(8) CD22(8) IL-1β(8) IL-5(8)	184
CD28	Anti-CD28 SA(80)	80

Table 2.6: Test data set for DTC

Class	MAbs and controls	# samples
Safe	PBS(18) AutoPlasma(54)	72
CD28	Anti-CD28 SA(54)	54

2.7 Cross Validation Method for Estimating DTC Accuracy

In order to estimate the classification accuracy of the DTC model on new data with unknown class labels, the following cross validation methodology was used: the entire training data set was split randomly into two sets: one set was used for training (2/3 of the data) and one set was used for cross validation (1/3 of the data). To prevent samples from the same donor from appearing in both sets, we grouped the samples by donor, and then randomly selected donors whose samples numbers added up to one third of the total number of samples. The samples from the rest of the donors were used for training the DTC model. This process was repeated multiple times until samples from all the donors were considered equally often for both training and validation. The classification accuracy estimate was defined as the average classification accuracy for all the training/testing set partitions.

3. Severity Estimation using Distance Metric Learning

In many circumstances, the binary determination of severity may be of limited value, and we may be interested in a multi-class and more graded determination of the level of severity. For such determination we propose the Severity Estimation using Distance Metric Learning (SE-DML) approach. SE-DML uses distance metric learning algorithms to develop multi-level of severity assessment. The four data sets used in the evaluation are Cytokine Release Syndrome (CRS), Cardiotocography (CTG), Pyrimidines and Triazines data sets.

3.1 Problem Formulation

The problem of estimating the disease's severity can be divided into two stages: (1) choosing the most medically relevant set of features describing the disease of interest and (2) combining these variables in a functional form (model) which is able to provide the most accurate severity estimation for the disease [12]. Focusing on the stage (2), here we proposes to tackle SE-DML. More specifically, we have observed that in most cases of biomedical severity estimation in practice, the reference data (i.e. the sample groups with known severity) normally include only positive (e.g. least severe disease state) and negative controls (e.g. most severe disease state). The reason is that in biomedical experiments such as blood assay, clinical trials and animal testing, many researchers utilize and label positive and negative controls to verify the success of their experiments. Thus SE-DML aims to solve the following problem.

- We are given a data set with multiple samples groups associated with different severity levels of a disease. Some sample group severity levels are known (positive and negative control groups) and some are unknown. The main goal is to

estimate the severity of unknown sample groups based on their relationship to the known ones.

Samples in the same group should match to the same level of severity. For example, a "group" could describe a certain disease state.

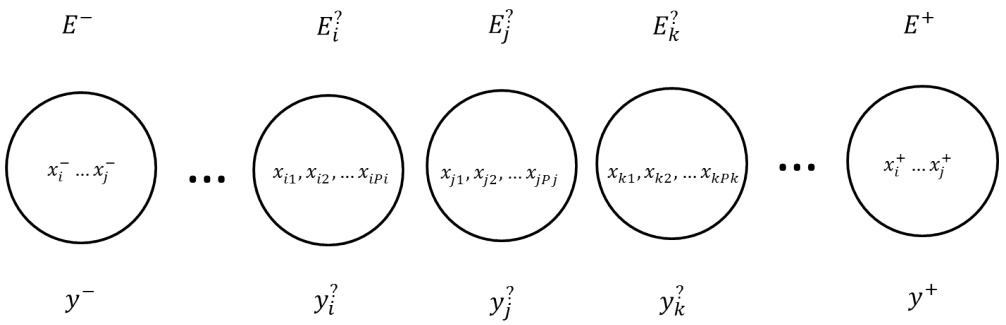


Figure 3.1: Problem formulation: the severity levels of positive control $\mathbf{E}^+$ and negative control $\mathbf{E}^-$ are known. The severity level $y_i^?$ of an unknown sample group $\mathbf{E}_i^?$ is estimated based on its distances to the two controls.

Our setup includes a data set of m-dimensional samples about a certain biomedical condition. These samples belong to n sample groups $\{\mathbf{E}_1, \dots, \mathbf{E}_n\}$, where each sample group $\mathbf{E} \in \mathbb{R}^{p_E \times m}$ contains p_E samples $\{\mathbf{x}_1, \dots, \mathbf{x}_{p_E}\}$ and corresponds to a severity level y_i of this biomedical condition. We assume that the severity levels y_i are numerical values between 0 and 1, with 0 being the least severe and 1 being the most severe. Among these n sample groups, some have known severity levels. As we mentioned above, in most cases, the sample groups with known severity are positive and negative controls. Here we define $y^+ = 1$ for positive control $\mathbf{E}^+$, whereas $y^- = 0$ for negative control $\mathbf{E}^-$. The objective is to estimate the severity level $y_i^?$ of an unknown sample group $\mathbf{E}_i^?$ based on its distances to $\mathbf{E}^+$ and $\mathbf{E}^-$. The problem definition of SE-DML is illustrated in Figure 3.1.

3.2 Basic Distance Metric Learning

Metric learning methods try to learn a Mahalanobis distance defined in expression (3.1), where A is a positive semi-definite m by m matrix of parameters learned from data. The learning process usually relies on pairwise constraints between sample points as training signals: (1) equivalent constraints (equation 3.2), which state that a given pair of data points are semantically similar and should be close together in the learned metric; and (2) inequivalent constraints (equation 3.3), which indicate that the given pairs of samples are semantically dissimilar and should not be close together in the learned metric [69].

$$d_A(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T A (\mathbf{x}_i - \mathbf{x}_j)} \tag{3.1}$$

$$\mathbf{S} = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \; and \; \mathbf{x}_j \; are \; similar\} \tag{3.2}$$

$$\mathbf{D} = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \; and \; \mathbf{x}_j \; are \; dissimilar\} \tag{3.3}$$

A commonly used formulation of distance metric learning [10] converts the above constraints to a convex programming task to learn the parameter matrix A:

$$\begin{aligned} \min_{A \in \mathbb{R}^{m \times m}} \quad & \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{S}} d_A(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}} d_A(\mathbf{x}_i, \mathbf{x}_j) \geq 1, \;\; \text{and} \;\; A \succeq 0. \end{aligned} \tag{3.4}$$

The positive semi-definite constraint $A \succeq 0$ is required to guarantee the learned distance between any two points (parameterized by A) cannot be negative and satisfies the triangle inequality.

For our targeted task handling a set of sample groups mapping to a range of severity levels, it is natural to think that one can calculate the distances between samples

with unknown severity to samples with known severity, in order to estimate the unknown severity. But the commonly used Euclidean distance metric may not capture the fact that samples from the positive control E^+ should be far from samples from the negative control E^-. The basic idea of distance metric learning is maximizing the distances between dissimilar sample groups, and minimizing the distances between samples in the same group or among similar groups. Specifically, the learned metric based on positive control and negative control should give a maximum distance $d(E^+, E^-)$ between these two controls. The distances between a sample group $E_i^?$ with unknown severity level and two controls can then be measured based on this learned metric. These distances should be proportional to $d(E^+, E^-)$ and can be combined to locate the position of this unknown group between the two controls, where the position indicates the severity level.

3.3 Overall Framework

The objective of SE-DML approach is to estimate the severity levels of n unknown sample groups $\{\mathbf{E}_1^?, \ldots, \mathbf{E}_n^?\}$ based on positive control $\mathbf{E}^+$ and negative control $\mathbf{E}^-$, which are known beforehand. The set of equivalence constrains S (equation 3.2) consists of pairs of samples within $\mathbf{E}^+$ or $\mathbf{E}^-$. The set of inequivalent constrains D (equation 3.3) consists of pairs of samples from different controls – one sample from $\mathbf{E}^+$ and one sample from $\mathbf{E}^-$. A Mahalanobis distance metric is then learned based on these constrains using the distance metric learning method described in Chapter 3.4. Based on the learned metric, the distances of the unknown groups to the controls are calculated and will be transformed to severity levels y as described in Chapter 3.5.

3.4 Information-Theoretic Metric Learning (ITML)

Given a distance metric parameterized by A, a corresponding multivariate Gaussian distribution could be expressed for describing samples where A^{-1} is the covariance matrix of the distribution, i.e.,

$$Pr(x|A) = \frac{1}{(2\pi)^{m/2}|A|^{1/2}} \exp(-\frac{1}{2}x^T A^{-1}x). \tag{3.5}$$

Considering the Euclidean distance (i.e., distance metric with identity matrix $A_0 = I$) works well as a baseline empirically, we regularize the learned metric matrix A with A_0. Probabilistically, this equals to minimize the distance between the two corresponding Gaussian distributions, denoted by $Pr(x|A)$ and $Pr(x|A_0)$. Typically, Kullback-Leibler (KL) divergence [70] is used to measure the distance between two distributions, thus the distance between $Pr(x|A)$ and $Pr(x|A_0)$ is given by,

$$\begin{aligned} d(A_0||A) &= KL(Pr(x|A_0)||Pr(x|A)) \\ &= \int Pr(x|A_0) \log \frac{Pr(x|A_0)}{Pr(x|A)} d\mathbf{x}. \end{aligned} \tag{3.6}$$

Then the log determinate (LogDet) formulation is used to simplify the $d(A_0||A)$ in a closed form:

$$d(A_0||A) = \frac{1}{2}(tr(A^{-1}A_0) + log|A| - log|A_0| - m), \tag{3.7}$$

where m is the dimensionality of the data. Suppose the means of the Gaussian distribution is 0, the proof for equation 3.7 is as follow,

$$
\begin{aligned}
& d(A_0||A) \qquad (3.8)\\
&= KL(Pr(x|A_0)||Pr(x|A)) \\
&= \int Pr(x|A_0) \log \frac{Pr(x|A_0)}{Pr(x|A)} d\mathbf{x} \\
&= \int [\log(Pr(x|A_0)) - log(Pr(x|A))] Pr(x|A_0) dx \\
&= \int [\frac{1}{2} \log \frac{|A|}{|A_0|} + \frac{1}{2} x^T A_0^{-1} x + \frac{1}{2} x^T A^{-1} x] Pr(x|A_0) dx \\
&= \frac{1}{2} \log \frac{|A|}{|A_0|} - \frac{1}{2} tr\{E(x^T x) A_0^{-1} + \frac{1}{2} E(x^T x) A^{-1}\} \\
&= \frac{1}{2} \log \frac{|A|}{|A_0|} - \frac{1}{2} tr\{I_m\} + \frac{1}{2} tr\{A^{-1} A_0\} \\
&= \frac{1}{2} (tr(A^{-1} A_0) + log|A| - log|A_0| - m)
\end{aligned}
$$

Based on the above formulation, Davis et al. proposed ITML [49][50] to tackle metric learning by minimizing the LogDet divergence (equation 3.7) plus side constraints (equivalent or inequivalent). The constraints used in ITML are similar to those described in Chapter 3.2, in which for two similar samples, their learned distance is constrained to be smaller than a given upper bound, i.e., $d_A(\mathbf{x}_i, \mathbf{x}_j) \leq u$ for a parameter u, and, for two samples that are known to be dissimilar, $d_A(\mathbf{x}_i, \mathbf{x}_j) \geq l$ for a parameter l. The objective is to learn a distance metric parameterized by parameter matrix A. To solve this optimization, ITML uses the so-called Bregman projections for solving a strictly convex optimization with respect to multiple linear inequality constraints. Using this simple first-order technique developed in [71], ITML repeatedly computes Bregman projections of the current solution onto a single constraint via the following update

$$A_{t+1} = A_t + \beta A_t (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T A_t, \qquad (3.9)$$

where β is the projection parameter (Lagrange multiplier) corresponding to the current constraint. It is positive for similar pairs and negative for dissimilar pairs.

3.5 Severity Estimation for a Sample Group

After learning a distance metric, we can calculate the distances between a sample $\mathbf{x}_i^?$ (with an unknown severity level $y_{x_i}^?$) to the positive control $\mathbf{E}^+$ and negative control $\mathbf{E}^-$. Thus the distance between $\mathbf{x}_i^?$ and $\mathbf{E}^+$ is defined as :

$$d_A(\mathbf{x}_i^?, \mathbf{E}^+) = (\frac{\sum\limits_{x_k \in \mathbf{E}^+} \mathbf{x}_k^+}{|\mathbf{E}^+|} - \mathbf{x}_i^?)^T A(\frac{\sum\limits_{x_k \in \mathbf{E}^+} \mathbf{x}_k^+}{|\mathbf{E}^+|} - \mathbf{x}_i^?). \tag{3.10}$$

Similarly, the distance between $\mathbf{x}_i^?$ and $\mathbf{E}^-$ is defined as :

$$d_A(\mathbf{x}_i^?, \mathbf{E}^-) = (\frac{\sum\limits_{x_k \in \mathbf{E}^-} \mathbf{x}_k^-}{|\mathbf{E}^-|} - \mathbf{x}_i^?)^T A(\frac{\sum\limits_{x_k \in \mathbf{E}^-} \mathbf{x}_k^-}{|\mathbf{E}^-|} - \mathbf{x}_i^?). \tag{3.11}$$

These two distances are used together to determine the severity level $y_{x_i}^?$ (equation 3.12) for a sample $\mathbf{x}_i^?$. If $y_{x_i}^?$ is closer to 0, the severity level of $\mathbf{x}_i^?$ is more similar to that of the negative control. If $y_{x_i}^?$ is close to 1, the severity level of $\mathbf{x}_i^?$ is more similar to that of the positive control.

$$y_{x_i}^? = \frac{d_A(\mathbf{x}_i^?, \mathbf{E}^-)}{(d_A(\mathbf{x}_i^?, \mathbf{E}^+) + d_A(\mathbf{x}_i^?, \mathbf{E}^-))}. \tag{3.12}$$

The severity $y_i^?$ of $\mathbf{E}_i^?$ is then defined as

$$y_i^? = \frac{\sum\limits_{\mathbf{x}_i^? \in \mathbf{E}_i^?} y_{x_i}^?}{|\mathbf{E}_i^?|}. \tag{3.13}$$

Table 3.1: List of cytokines release measured in the assay for SE-DML

Cytokine	Mean (pg/ml)	Median(pg/ml)	Maximum (pg/ml)
IL-1β	457.68	222.37	4550.80
IL-2	27.07	2.34	2636.76
IL-4	1.69	0.38	45.20
IL-6	7875.14	2591.21	88002.05
IL-8	24106.95	13859.18	427144.20
IL-10	15.09	3.50	636.68
IL-12(p70)	8.57	5.88	91.37
IL-17	25.11	0.10	2410.40
IL-18	14.36	11.42	131.41
IFN-γ	3222.47	36.68	90400.57
TNF-α(monometric)	455.00	267.08	3729.62
TNF-α(trimetric)	306.93	173.67	1819.08

3.6 Cytokine Release Syndrome Data Set

This CRS data set used here is generated by the same in intro assay as described in Chapter 2.1 with more treatments. The assay measured the concentrations of the 12 cytokines shown in Table 3.1. Data were reported in pg/ml for each sample and each cytokine. To allow calculation of mean values and graphic analysis, all concentrations below the level of quantitation were set to 0.1 [18].

The data set from the in vitro assay contains a total of 30 treatments listed in Table 3.2. Each treatment has a different number of samples indicated in the parenthesis. There are a total of 711 samples in the data set. For each sample, the 12 cytokines described in Table 3.1 are measured as features. The 30 treatments have been roughly classified into 5 groups based on the severity descriptions of CRS found in clinical literature, which ranges from the most severe CRS caused by anti-CD28 SA to no reaction at all [56][72]. The 5 groups are negative control($\mathbf{E}^{-}$), safe($\mathbf{E}_1^{?}$), middle($\mathbf{E}_2^{?}$), severe($\mathbf{E}_3^{?}$) and positive control($\mathbf{E}^{+}$). The negative control group has no CRS reaction at all. The safe group contains treatments not likely to cause CRS or an infusion reaction. The middle group contains treatments that could potentially

Table 3.2: List of treatments (mAbs) used in the CRS data set for SE-DML. Each treatment has a different number of samples indicated in the parenthesis

Negative Control($\mathbf{E}^-$)	Safe($\mathbf{E}_1^?$)	Middle($\mathbf{E}_2^?$)	Severe($\mathbf{E}_3^?$)	Positive Control($\mathbf{E}^+$)
AutoPlasma (152)	Adalimumab (16)	Anti-VEGF (8)	CD2 (16)	Anti-CD28 SA (152)
PBS (80)	Alemtuzumab (8)	Basiliximab (8)	CD28-Biol (8)	LPS (80)
	CD11a (8)	IL-12 (8)	CD28-SB (8)	
	CD22 (8)	KV1D261-1.003 (8)	CD20 (23)	
	CD80 (8)	KV1D261-20.001 (8)	CD3 (16)	
	Cetuximab (8)	Panitumumab (8)	CD3/CD28 (8)	
	IL-1b (8)		CD4(8)	
	IL-5 (8)		CD40 (8)	
	IL-6 (8)			
	IL-6-B-E8 (8)			
	Palivizumab (8)			
	Tocilizumab (8)			

cause infusion reactions, but milder than the reactions caused by the treatments in severe-CRS class. The severe-CRS group contains treatments that will cause severe CRS. The positive control group will cause the most severe CRS.

3.6.1 Evaluation Setup

To reduce the differences in the dynamic ranges of the 12 features, the data are z-score transformed first. The z-score transformation is defined as $z = \frac{x-\mu}{\sigma}$, where x is a raw data sample, μ is the means of the sample population and σ is the standard deviation of the sample population. The distance metric parameterized by A is learned based on the positive controls ($\mathbf{E}^+$) and negative controls ($\mathbf{E}^-$) in Table 3.2. The constrained sample pairs are formulated by the samples within the two control groups. The lower and upper bounds of the right hand side of the constraint (l and u) described in Chapter 3.4 are the 5^{th} and 95^{th} percentiles of the observed distribution of distances between pair of points within two control groups, respectively.

3.7 Cardiotocography Data Set

Cardiotocography (CTG) is the most widely used tool for fetal surveillance. The data set contains 2126 fetal CTG samples that were classified by three expert obste-

Table 3.3: Number of samples in each class of CTG data set

Class	Number of Samples
Normal	1655
Suspect	295
Pathologic	176

tricians and a consensus classification label assigned to each of them [37][73]. The classification labels are based on the severity of fetal abnormal states: normal, suspect and pathologic. The number of samples for each class are shown in Table 3.3. The 21 diagnostic features for each sample are shown in Table 3.4.

3.7.1 Evaluation Setup

The CTG data set is normalized through each feature by z-score transformation. The z-score transformation is defined as $z = \frac{x-\mu}{\sigma}$, where x is a raw data sample, μ is the means of the sample population and σ is the standard deviation of the sample population. To evaluate proposed SE-DML approach, a 10-fold cross validation strategy was used. Since there are only 3 classes in the data set, in each iteration of the 10-fold cross validation, 90% of the normal class samples and pathologic samples were used to formulate negative controls $\mathbf{E}^-$ and positive controls $\mathbf{E}^+$ which were used to learn distance metric parameterized by A. The constrained sample pairs are formulated by the samples within $\mathbf{E}^-$ and $\mathbf{E}^+$. The lower and upper bounds of the right hand side of the constraint (l and u) described in Section 3.4 are the 5^{th} and 95^{th} percentiles of the observed distribution of distances between pair of points within positive and negative controls data, respectively. The remaining 10% samples of these two classes and the entire suspect class samples $\{\mathbf{E}_1^?, \mathbf{E}_2^?, \mathbf{E}_3^?\}$ are used to test the performance. The average results of the 10 iterations are used as the final evaluation results.

Table 3.4: The 21 diagnostic features of CTG data set

Index	Feature
1	LB - FHR baseline (beats per minute)
2	AC - number of accelerations per second
3	FM - number of fetal movements per second
4	UC - number of uterine contractions per second
5	DL - number of light decelerations per second
6	DS - number of severe decelerations per second
7	DP - number of prolongued decelerations per second
8	ASTV - percentage of time with abnormal short term variability
9	MSTV - mean value of short term variability
10	ALTV - percentage of time with abnormal long variability
11	MLTV - mean value of long term variability
12	Width - width of FHR histogram
13	Min - minimum of FHR histogram
14	Max - Maximum of FHR histogram
15	Nmax - number of histogram peaks
16	Nzeros - number of histogram zeros
17	Mode - histogram mode
18	Mean - histogram mean
19	Median - histogram median
20	Variance - histogram variance
21	Tendency - histogram tendency

3.8 Quantitative Structure Activity Relationship Data Sets

Quantitative Structure Activity Relationships (QSAR) relate to the interaction of chemical compounds with biological systems. These relationships are essential to toxicological investigation in the development of pharmaceutical compounds. Our severity estimation approach is used to predict the toxicity of two families of chemical compounds, Pyrimidines and Triazines, based on their QSAR data sets [3], where each compound has 5 levels of severity as class labels.

In forming a QSAR for a series of chemical compounds, we consider the compounds to have a common structure onto which substituent groups are added [1].

Table 3.5: Physico-chemical attributes of substituent in Pyrimidines and Triazines [1]

Attribute name	Notation
Polarity	PL
Size	SZ
Flexibility	FL
Hydrogen-bond donor	HD
Hydrogen-bond acceptor	HA
π bond acceptor	ΠA
π bond donor	ΠD
Polarizability	PO
δ effect	δ
Branching	BR

Substituent groups are groups of atoms substituted in place of a hydrogen atom on the parent chain of a hydrocarbon and represent physico-chemical attributes. The physico-chemical attributes of the substituent in Pyrimidines and Triazines are shown in Table 3.5. These physico-chemical attributes in the Table 3.5 are considered as features that related to the two compounds' toxicity. The SE-DML approach developed here was used to capture the relationships between features and compounds' toxicity. In Pyrimidines there are three possible regions for a substituent as shown in Figure 3.2, and with nine features for each region (Branching was not used), each sample in the Pyrimidines data set has an feature vector of 27 elements. In additional, each sample has a real value activity class label denoted as the severity level. For example, a sample in the Pyrimidines set with a CI substituted at position R3, OCH_3 group substituted at position R4, and a CI group substituted at position R5, is represented by the feature vector shown in Figure 3.3. The first 9 features are the physico-chemical attributes of CI substituent at position R3 (The attributes in Table 3.5 excluding the last attribute Branching). The second 9 features are the physico-chemical attributes of OCH_3 position R4. The third 9 features are the physico-chemical attributes of CI position R5. The last number, 2, indicates the

severity level [3]. For the Triazines data set, compounds have six possible regions for a substituent and there are 10 features for each regions. So each sample has a total of 60 features.

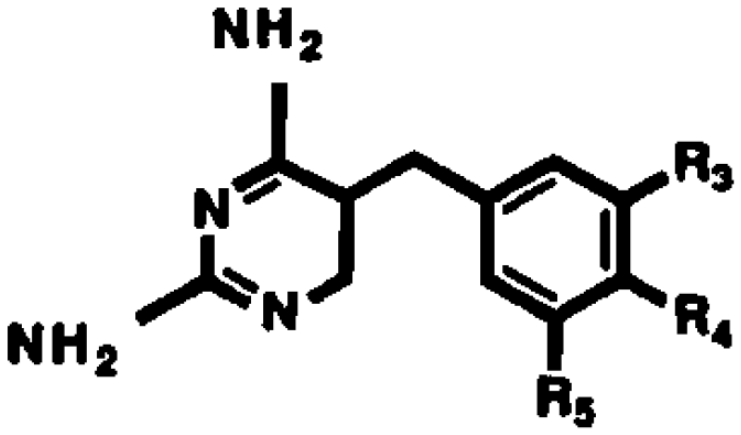


Figure 3.2: Structure of the Pyrimidines where substitutions can occur at positions R3, R4 and R5 [1].

3.8.1 Evaluation Setup

The activity values that indicate the toxicity levels for both Pyrimidines and Triazines data sets were discretized into five intervals [3]. These five intervals are denoted as five sample groups: $\{\mathbf{E}^-, \mathbf{E}_1^?, \mathbf{E}_2^?, \mathbf{E}_3^?, \mathbf{E}^+\}$, with severity levels $y^- < y_1^? < y_2^? < y_3^? < y^+$. Each data set has been randomly separated into control and test partitions. The control partition is used to learn a distance metric, where, $\mathbf{E}^-$ is negative control and $\mathbf{E}^+$ is positive control. In the test partition, the $\mathbf{E}_1^?$, $\mathbf{E}_2^?$ and $\mathbf{E}_3^?$ sample groups are used for severity estimation in order to evaluate our SE-DML

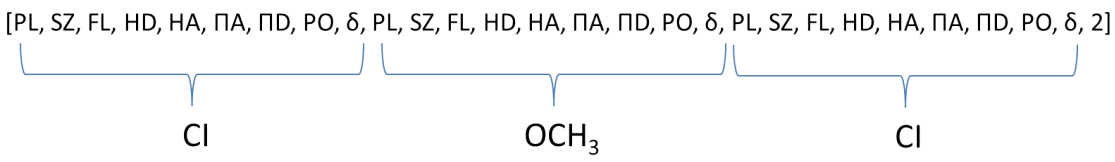


Figure 3.3: Feature vector for a sample in Pyrimidines data set. The first 9 features are the physico-chemical attributes of CI substituent at position R3 (The attributes in Table 3.5 excluding the last attribute Branching). The second 9 features are the physico-chemical attributes of OCH_3 position R4. The third 9 features are the physico-chemical attributes of CI position R5. The last number, 2, indicates the severity level [3].

approach. The partitioning was repeated 20 times with replacement[1] and the number of samples for both control and test partitions are shown in Table 3.6. The average severity levels of $\mathbf{E}_1^?, \mathbf{E}_2^?, \mathbf{E}_3^?$ are used as the final results.

Table 3.6: Characteristic of the two QSARs data sets: the number of features and number of samples for the two partitions

Data Sets	Number of Samples in Control Partition	Number of Samples in Test Partition	Number of features
Pyrimidines (74)	50	24	27
Triazines (186)	100	86	60

[1]The data sets and the partitions generated are available to download at http://www.gatsby.ucl.ac.uk/ chuwei/ordinalregression.html

3.9 Algorithm Comparison

The evaluation setup for all four datasets (Cytokine Release Syndrome (CRS), Cardiotocography (CTG), Pyrimidines and Triazines data sets) are shown in Table 3.7. We implement the following five approaches to compare their ability to estimate the severity on these four data sets:

1. Severity Estimation using Distance Metric Learning (SE-DML) where we use Information-Theoretic Metric Learning (ITML) as the metric learning algorithm;

2. Euclidean distance under the same framework of SE-DML; we use this approach to compare how learned distance metric improve the performance of severity estimation over commonly used distance metric;

3. Large Margin Nearest Neighbors (LMNN), another state-of-the-art metric learning algorithm [52]. We use LMNN under the same framework of SE-DML;

4. Linear Regression where we use $\mathbf{E}^+$ and $\mathbf{E}^-$ with severity level 1 and 0, respectively, to build the regression model to predict severity levels of individual samples in each test class $\{\mathbf{E}_1^?, \mathbf{E}_2^?$ and $\mathbf{E}_3^?\}$;

5. Support Vector Regression, using the same setup as linear regression, implemented by libsvm v3.18 with radial basis function kernel function [74] on Matlab® 2012a software (The Mathworks, Natick, MA).

3.10 Evaluation Criteria

We use two evaluation criteria for comparing the five approaches described above. The first is the order of severity levels of each test group. Second, in order to measure how well each sample's estimated severity level lies within its group, we use a

Table 3.7: Evaluation setup of four data sets. The positive control $\mathbf{E}^{+}$ and negative control $\mathbf{E}^{-}$ are used to learn distance metric. The three middle groups $\mathbf{E}_1^?$, $\mathbf{E}_2^?$ and $\mathbf{E}_3^?$ are used as test data.

	Levels of Severity				
Data Sets	$\mathbf{E}^{-}$	$\mathbf{E}_1^?$	$\mathbf{E}_2^?$	$\mathbf{E}_3^?$	$\mathbf{E}^{+}$
CRS	AutoPlasma & PBS	Safe treatments	Middle treatments	Severe-CRS treatments	Anti-CD28 SA & LPS
CTG	90% Normal	10% Normal	100% Suspect	10% Pathologic	90% Pathologic
Pyrimidines	Level 1	Level 2	Level 3	Level 4	Level 5
Triazines	Level 1	Level 2	Level 3	Level 4	Level 5

silhouette coefficient [75], which contrasts the average distance of a sample to other samples in the same cluster with the average distance to samples in other clusters. The silhouette coefficient is defined in equation 3.14, where a_i is the average distance from the i^{th} sample to the other samples in the same cluster and b_i is the minimum average distance of the i^{th} sample to samples in a different cluster which is the closest to the cluster of i^{th} sample. This coefficient has a value between -1 and +1 where a higher value, closer to +1, indicates that the sample is well-matched to its own group, and poorly-matched to other groups.

$$s_i = \frac{b_i - a_i}{max(a_i, b_i)} \tag{3.14}$$

4. Application of High-dimensional Molecular Profiling Data to Cancer Tissue Classification

Cancer molecular profiling data are typically high-dimensional and make machine metric learning quite challenging. When given a data set $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}]$, where each $\mathbf{x}_i \in \mathbb{R}^m$,Information-Theoretic Metric Learning (ITML) introduced in previous chapter will learn a distance metric parameterized by a $m \times m$ matrix A. If the dataset is high-dimensional, where the feature number m is relatively large in gene microarray data set, ITML needs to estimate m^2 parameters in A which is certainly not ideal considering the computational load.

4.1 Kernelized Information-Theoretic Metric Learning (KITML) for High-Dimensional Data

The commonly formulation of distance metric learning [10] is a convex programming task that tries to learn the parameter matrix A:

$$\begin{aligned} \min_{A \in \mathbb{R}^{m \times m}} \quad & \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{S}} d_A(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}} d_A(\mathbf{x}_i, \mathbf{x}_j) \geq 1, \text{ and } A \succeq 0. \end{aligned} \tag{4.1}$$

The positive semi-definite constraint $A \succeq 0$ is required to guarantee the learned distance between any two points (parameterized by A) cannot be negative and satisfies the triangle inequality.

ITML solves the metric learning problem as minimizing the relative entropy between two multivariate Gaussians distribution under side constraints. Two samples are similar if the Mahalanobis distance between them is smaller than a given upper

bound, i.e., $d_A(\mathbf{x}_i, \mathbf{x}_j) \leq u$ for a relatively small value of u. Similarly, two samples are dissimilar if $d_A(\mathbf{x}_i, \mathbf{x}_j) \geq l$ for a relatively large l. The objective is to learn a Mahalanobis distance parameterized by A which should be as close as possible to a prior distance function A_0, e.g. Euclidean distance. The closeness of the solution to the prior is measured by the Kullback-Leibler (KL) divergence [70]:

$$d(A_0||A) = \int Pr(x|A_0) \log \frac{Pr(x|A_0)}{Pr(x|A)} d\mathbf{x}. \tag{4.2}$$

The optimization problem for ITML can be solved by the connection between KL-divergence and the LogDet divergence. Therefore, the optimization problem can be expressed as following [49, 50],

$$\begin{aligned} \min_{A} \quad & d(A||A_0) \\ \text{s.t.} \quad & tr(A(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T) \leq u, (i,j) \in S, \\ & tr(A(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T) \geq l, (i,j) \in D, \\ & A \succeq 0. \end{aligned} \tag{4.3}$$

Given a microarray data set $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}]$, each $\mathbf{x}_i \in \mathbb{R}^m$ is a data vector with m features. ITML will try to learn a distance metric parameterized by $m \times m$ matrix A. Since m is relatively large in microarray data set, ITML will be slow under this situation. To adapt ITML for datasets with $n \ll m$, we employ the kernel function and present the Kernelized Information-Theoretic Metric Learning (KITML) for learning a kernel matrix $K = X^T AX$. Under this formulation, we only need to estimate $n \times n$ parameters in the matrix K which is much smaller than $m \times m$ parameters in the original A matrix. The distance between two points based on K

can be denoted as

$$d_A(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{e}_i - \mathbf{e}_j)^T K (\mathbf{e}_i - \mathbf{e}_j), \tag{4.4}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are the unit basis vectors in which only the entry i or j is 1 and the rest are 0.

The optimization problem is to search for K that satisfies the similar/dissimilar side constraints as well as minimizing the KL divergence. Similarly, A_0 is transformed to kernelized $K_0 = X^T A_0 X$ for the regularization distribution.

$$\begin{aligned} \min_{A} \quad & d(K_0||K) \\ \text{s.t.} \quad & (\mathbf{e}_i - \mathbf{e}_j)^T K (\mathbf{e}_i - \mathbf{e}_j) \leq u, (i,j) \in S, \\ & (\mathbf{e}_i - \mathbf{e}_j)^T K (\mathbf{e}_i - \mathbf{e}_j) \geq l, (i,j) \in D, \\ & K \succeq 0. \end{aligned} \tag{4.5}$$

The parameters, upper bound u and lower bound l are determined by the distribution of the data. The optimization is performed through Bregman projections [71], where in each iteration, a constraint $(i,j) \in S$ or $(i,j) \in D$ is picked to update the matrix K. The Bregman projection update is similar to equation 3.9 and could be denoted as,

$$K_{t+1} = K_t + \beta K_t (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T K_t, \tag{4.6}$$

where β is the same as that in equation 3.9.

4.2 Calculating Distance in KITML for High-dimensional Microarray Data

When using KITML to improve the classification cancer microarray data, the distance metric parameterized by kernel matrix K is learned by certain samples that are considered as training data. During the testing phase, we need to calculate distances among samples that might not be covered by the kernel K, thus, we could not use equation 4.4 directly. Through a few steps of derivation and a theorem from [51], we can construct A in a closed-form from K in the following manner.

$$A = \alpha I + XTX^T, \tag{4.7}$$

$$\text{where } T = K_0^{-1}(K - \alpha K_0)K_0^{-1}, \tag{4.8}$$

Here α is suggested to be 1 [51]. We can calculate the distance between any two "high-dimensional" sample points using an *implicit* representation of A through kernel evaluation as, namely

$$\begin{aligned} d_A(\mathbf{x}_i, \mathbf{x}_j)^2 &= (\mathbf{x}_i - \mathbf{x}_j)^T(I + XTX^T)(\mathbf{x}_i - \mathbf{x}_j) \\ &= (\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j) + (\mathbf{x}_i - \mathbf{x}_j)^T XTX^T(\mathbf{x}_i - \mathbf{x}_j). \end{aligned} \tag{4.9}$$

Instead of learning m^2 parameters in A (m is the number of features), only n^2 parameters (n is the number of samples) need to be learned by using the above kernel formulation. KITML thus permits efficient optimization and low storage through equation 4.6. At the same time, equations 4.8 and 4.9 make the evaluation of the learned distance metric (i.e. calculating distances) efficient as well.

4.3 Sample-level Tissue Classification with K-Nearest Neighbor (KNN) KITML

K-Nearest Neighbor (KNN) classification is a commonly used classification algorithm for cancer data. For high-dimensional molecular signature data, when using metrics like Euclidean distance, KNN is often inferior to more sophisticated approaches such as Support Vector Machines [16]. In this thesis, we use KITML to actively learn a distance metric to improve the performance of KNN-driven cancer classification. KITML also reduce the heavy computation burden of distance metric learning through kernelization. The process works as following:

1. Compute the distances of a test sample $\mathbf{x}_i$ to the labeled training samples $\mathbf{y}_i$ using equation 4.9.

2. Order the training samples by increasing distances from the test sample.

3. Use cross validation to find the optimal number of nearest neighbors, k, based on training samples.

4. Use majority vote (or inverse distance weighted average based on k nearest neighbors) to determine the class of the test sample $\mathbf{x}_i$.

4.3.1 Other Algorithms compared with KITML

We compared KITML performance with several state-of-the-art algorithms.

1. **KNN Classification with distance metric learned by ITML*** Directly learning distance metric from high-dimensional data set by ITML is quite slow. Our data sets originally contain between 1000 and 4000 features so we first use a variance feature selection process to obtain a reduced feature set of size 100. The metric learning process and KNN classification are based on these 100

features with the highest variance. We denote this ITML algorithm using the 100 selected features as ITML*.

2. **K-Nearest Neighbor (KNN) Classification with Euclidean Distance** Here we use Euclidean distance as a baseline to illustrate that KITML can improve KNN classification.

3. **Multi-class Support Vector Machine (SVM) with Linear Kernel** SVM often achieves superior classification performance compared to other learning algorithms across most domains and tasks [16]. SVM maps data to a higher dimensional space via a kernel function and then solves an optimization problem to find the maximum-margin hyperplane that separates training samples. The test samples are classified based on their separation by hyperplane. In this thesis, we compare SVMs implemented by libsvm v3.18 [74] using a linear kernel $K(x, y) = x^T y$, where x and y are samples with gene expression values.

4. **Multi-class Support Vector Machine (SVM) with Radial Basis Function Kernel** We compare SVM using a radial basis function (RBF) kernel $K(x, y) = \exp(-\gamma|x - y|^2)$, where x and y are samples with gene expression values and γ is kernel parameter. It was also implemented by libsvm v3.18 [74].

5. **Large Margin Nearest Neighbor Classification** Large Margin Nearest Neighbor (LMNN) [52] is a popular metric learning algorithm that learns a Mahalanobis distance metric for KNN classification from labeled examples. It aims at improving KNN classification by exploiting the local structure of the data. In this algorithm, the distance metric is trained with the goal that k nearest neighbors always belong to the same class while examples from different classes are separated by a large margin. The algorithm attempts to increase the number of training examples with this property by learning a linear trans-

formation of the input space that precedes KNN classification using Euclidean distances [52]. The loss function minimized by the linear transformation consists of two terms. The first term penalizes large distances between examples in the same class, while the second term penalizes small distances between examples with different class labels.

6. **Decision Tree Classification (DTC)** The DTC algorithm applies a divide and conquer approach, resulting in an iterative process that starts by analyzing each feature from the training data. It calculates the information gain for each feature with respect to the possible class outcomes present in the training data [55]. The C4.5 DTC algorithm implemented in the Weka 3.6.6 software [61] (University of Waikato, Hamilton, New Zealand) was used here to analyze the data.

7. **Random Forest** Random forest is an ensemble learning method for classification and regression using multiple decision tree models. Each model is built with a different subset of the same training data, with replacement. The remaining training data are used to estimate error and to determine the importance of each variable. We used the Random Forest algorithm implemented in the Weka 3.6.6 software [61] (University of Waikato, Hamilton, New Zealand) to classify the data.

4.3.2 High-dimensional Microarray Data Sets

Fourteen publicly available microarray data sets in Table 4.1 were used to evaluate our KITML approach. These data sets were obtained using two microarray technologies: single-channel Affymetric chips (6 sets) and double-channel cDNA chips (8 sets). For each data set, the total number of samples, number of features, number of classes, number of samples in each class, type of microarray chip, and tissue type are pro-

Table 4.1: Sample-level cancer tissue classification data set description

Dataset Name	Total Samples	Num of Features	Num of Classes	Num of Samples in Each Class	Tissue
Alizadeh [78]	42	1095	2	21, 21	Blood
Bittner [79]	38	2201	2	19, 19	Skin
Bredel [80]	50	1739	3	31, 14, 5	Brain
Garber [81]	66	4553	4	17, 40,4, 5	Lung
Golub-v1 [82]	72	1877	2	47, 25	Bone marrow
Golub-v2 [82]	72	1877	3	38, 9, 25	Bone marrow
Gordon [83]	181	1626	2	31, 150	Lung
Nutt [84]	28	1070	2	14,14	Brain
Pomeroy [85]	42	1379	5	10, 10, 10, 4, 8	Brain
Su [86]	174	1571	10	26, 8, 26, 23,12, 11, 7, 27, 6, 28	Multi-tissue
Tomlins-v1 [87]	104	2315	5	27, 20, 32, 13, 12	Prostate
Tomlins-v2 [87]	92	1288	4	27, 20, 32, 13	Prostate
Yeoh-v1 [88]	248	2526	2	43, 205	Bone marrow
Yeoh-v2 [88]	248	2526	6	15, 27, 64, 20, 79, 43	Bone marrow

vided in Table 4.1. The 14 data sets had 2-10 distinct classes, 28-248 samples, and 1095-4553 features. These data sets have been used for clustering analysis in previous studies [76, 77]. The clustering algorithms that have been applied to these data sets are k-means clustering, mixture model clustering, spectral clustering, etc.. Most of these data sets have not been explored for classification purposes, so we applied different classification algorithms to these 14 data sets. The results presented in this thesis are not comparable with analyses provides in the past studies since their analyses used clustering algorithms and our's used supervised classification algorithms.

4.3.3 Evaluation Setup

The experimental setup was designed to obtain reliable performance estimates and avoid over-fitting, we used two loops. The inner loop was used to determine the best parameters of the classifier using cross-validation sets. The outer loop was used to estimate the performance of the classifiers built using the parameters found by the inner loop. The test data sets used in the outer loop were independent from the cross-validation sets. The outer loop uses a 10-fold cross-validation and the inner loop uses a 4-fold cross-validation. We ran each of the 14 data sets through both our KITML and the six test algorithms five(5) times and averaged the classification results.

4.3.4 KITML Setting

To construct constrained pairs, we consider the pairs of samples in the same class to be similar and pairs of samples in different classes to be dissimilar. A total of $20C^2$ constrained pairs were randomly chosen in the learning process, where C is the number of classes in each data set. The lower and upper bounds of the right hand side of the constraints (l and u) in equation 4.5 are the 5^{th} and 95^{th} percentiles of the observed distribution of distances between pairs of samples within each data set.

4.3.5 Performance Metrics

We used two classification performance metrics. The first is accuracy, which is easy to interpret and simplifies statistical testing. Accuracy is defined as $Acc = \frac{TP+TN}{TP+TN+FP+FN}$, where TP is the total number of true positives, TN is the total number of true negatives, FP is the total number of false positives, and FN is the total number of false negatives. Accuracy is sensitive to the prior class probabilities but does not fully describe the actual difficulty of the decision problem for highly unbalanced distributions [16]. The second metric is macro-averaged F1 (F-measure). The F-measure is a weighted combination of precision and recall. Precision is defined as the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved [89]. Recall is defined as the ratio of the number of relevant records retrieved to the total number of relevant records in the data set [89]. F-measure is defined as:

$$F = \frac{(\beta^2 + 1)P_{macro}R_{macro}}{\beta^2 P_{macro} + R_{macro}}, \tag{4.10}$$

where β is typically set to 1. The multi-class precision and recall is define as:

$$P_{macro} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{TP_i}{TP_i + FP_i}, \tag{4.11}$$

$$R_{macro} = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{TP_i}{TP_i + FN_i}, \tag{4.12}$$

where TP_i is the number of true positives for class i, FP_i is the number of false positives for class i, FN_i is the number of false negatives for class i, and C is the class.

4.3.6 Statistical Comparison among Classifiers

Statistical comparison is used to verify that the differences in accuracy between algorithms are non-random. Since we have only 14 datasets we cannot assume that the difference between results are normally distributed [90]. For this reason, we have used the Wilcoxon signed-ranks test [91], which is a non-parametric alternative to paired t-test. The Wilcoxon signed-ranks test ranks the difference in performance of two classifiers for each data set, ignoring the signs, and compares the ranks for positive and negative differences.

Let N be the number of pairs, $x_{1,i}$ and $x_{2,i}$ are the pairs of observation where $i = 1, 2, \ldots, N$. The Wilcoxon signed-ranks test works as follows.

1. The null hypothesis H_0 is that the median difference between pairs of observation is zero. The alternative hypothesis H_1 is that the median difference is not zero.

2. The absolute values of the differences between pairs of observations $|x_{1,i} - x_{2,i}|$ and $|sgn(x_{1,i} - x_{2,i})|$ are calculated, where sgn is the sign function.

3. The smallest absolute difference pair gets a rank of 1, then next larger difference pair gets a rank of 2, etc. The pairs with zero difference are excluded from the test. The pairs with the same difference receive a rank as the average of the ranks they span.

4. The test statistic W is denoted as

$$W = |\sum_{i=1}^{N_r} sgn(x_{1,i} - x_{2,i})R_i|, \tag{4.13}$$

where Nr is the remaining pairs excludes zero difference pairs, and R_i is the rank.

5. For $N_r \geq 10$, since the sampling distribution of W coverages to a normal distribution, a z-score can be calculated as $z = \frac{W-0.5}{\delta_w}$, $\delta_w = \sqrt{\frac{N_r(N_r+1)(2N_r+1)}{6}}$. If $z > z_{critical}$, we reject H_0.

6. For $N_r < 10$, W is compared to a critical value from a reference table, if $W > W_{critical}$, we reject H_0.

4.4 Group-level Severity/Stage Estimation with Set-ranking KITML

Another important task for molecular profiling based cancer diagnosis is the ability to further quantify/classify samples like blood or tumor samples into subtypes which have distinct biomedical properties and result in varied prognoses. For instance, samples of "blood cancers" –Diffuse Large B-Cell Lymphomas (DLBCLs)– are indistinguishable based on histological methods yet are clinically heterogeneous. Some patients respond well and exhibit prolonged survival while some do not [92]. It has been shown that using expression profiling techniques to stratify DLBCLs to two subtypes is necessary for better classification [92]. For most cases of disease severity/stage estimation in practice, the reference data normally include only positive (e.g., most severe disease state) and negative controls (e.g., least severe disease state). The reason is that in many experiments using a blood assay or in clinical trials only positive and negative controls were labeled to verify the success of the studied techniques.

It may be advantageous to design more advanced computational methods for categorizing the subtypes of cancer samples through molecular expression representation. Here we propose a set-based ranking method using metrics learned from KITML for severity estimation. Normally given a data set with multiple sample groups associated with different severity levels of a type of cancer, we assume that the severity levels of the control groups are known; the severity levels of the remaining sample groups are unknown. The goal is to estimate the severity levels of these unknown sample groups based on their relationship to the known control groups.

The basic idea of set-based KITML is to maximize the distances between dissimilar sample groups, and minimize the distances between samples in the same group or among similar groups. Specifically, the learned metric based on positive control and negative control should give a maximum distance $d(E^+, E^-)$ between these two controls. The distances, between a sample group $E^?$ with unknown severity level and two controls, can then be measured using this learned distance. These distances should be proportional to $d(E^+, E^-)$ and can be combined to locate the position of the unknown sample group between the two controls, where the position indicates the severity level.

Under the high-dimensional setting, using the parameter T learned through equation 4.8, we can calculate the distance measure between any data samples. Therefore, we define and calculate distances between an unknown severity sample $\mathbf{x}_i^?$ (within $\mathbf{E}^?$) to $\mathbf{E}^+$, and to $\mathbf{E}^-$. The distance between $\mathbf{x}_i^?$ and $\mathbf{E}^+$ is defined as

$$d_A(\mathbf{x}_i^?, \mathbf{E}^+) = (\frac{\sum\limits_{x_k \in \mathbf{E}^+} \mathbf{x}_k^+}{|\mathbf{E}^+|} - \mathbf{x}_i^?)^T (I + XTX^T)(\frac{\sum\limits_{x_k \in \mathbf{E}^+} \mathbf{x}_k^+}{|\mathbf{E}^+|} - \mathbf{x}_i^?). \tag{4.14}$$

Similarly, the distance between $\mathbf{x}_i^?$ and $\mathbf{E}^-$ is defined as

$$d_A(\mathbf{x}_i^?, \mathbf{E}^-) = (\frac{\sum\limits_{x_k \in \mathbf{E}^-} \mathbf{x}_k^-}{|\mathbf{E}^-|} - \mathbf{x}_i^?)^T (I + XTX^T)(\frac{\sum\limits_{x_k \in \mathbf{E}^-} \mathbf{x}_k^-}{|\mathbf{E}^-|} - \mathbf{x}_i^?). \tag{4.15}$$

These two distances are then used to determine the predicted severity level $y_{x_i}^?$ of $\mathbf{x}_i^?$ (equation 4.16), which indicate the predicted severity level is the ratio between the distance of $\mathbf{x}_i^?$ to negative control and the distance of $\mathbf{x}_i^?$ to positive control. When $y_{x_i}^?$ is close to 0, the severity of $\mathbf{x}_i^?$ is similar to that of the negative controls. On the other hand, if $y_{x_i}^?$ is close to 1, the severity of $\mathbf{x}_i^?$ is similar to that of the positive controls.

$$y_{x_i}^? = \frac{d_A(\mathbf{x}_i^?, \mathbf{E}^-)}{(d_A(\mathbf{x}_i^?, \mathbf{E}^+) + d_A(\mathbf{x}_i^?, \mathbf{E}^-))}. \tag{4.16}$$

The severity $y^?$ of $\mathbf{E}^?$ is then defined as

$$y^? = \frac{\sum\limits_{\mathbf{x}_i^? \in \mathbf{E}^?} y_{x_i}^?}{|\mathbf{E}^?|}. \tag{4.17}$$

4.4.1 High-dimensional Molecular Profiling Data Data Sets

We used three microarray datasets from bladder, prostate and ovarian multi-stage cancer patient studies (Table 4.2) [93]. (1) The bladder dataset contains gene expression data of human bladder tumor samples from a clinical specimen bank. There are 20 Ta (stage 1) samples, 11 T1 (stage 2) samples and 9 T2+ (stage 3) samples, which contain a total of 7129 genes. After pre-processing according to [93], we removed genes having missing data, leaving 3036 genes for our analysis. (2) The prostate cancer data set was created in an attempt to characterize gene expression profiles of specific Gleason patterns. The dataset contains gene expression data of 11 Gleason pattern three (stage 1) samples, 12 Gleason pattern four (stage 2) samples and 8 Gleason pattern five (stage 3) samples. After removing the data with missing

Table 4.2: Estimating severity of sample subgroups data set description

Dataset Name	Total Samples	Num of Features	Num of Features after Pre-processing	Staging	Num of Sample in Each Stage
Bjladder [95]	40	7129	3036	Ta, T1, T2+	20, 11, 9
Prostate [96]	31	15488	9491	Gleason patterns 3,4,5	11, 12, 9
Ovary [94]	37	22283	18091	T1, T2, T3	18,5,14

values, 9491 genes were left for analysis. (3) The ovary data set is from genetically engineered mouse models which are used to demonstrate the mutations of certain signaling pathways in woman and mouse ovarian endometriod adenocarcinomas [94]. There are 18 T1 (stage 1) samples, 5 T2 (stage 2) samples and 14 T3 (stage 3) samples. After pre-processing, 18091 genes were left for our analysis.

4.4.2 KITML Setting

Since there are 3 stages in each data set, around 75% of the stage 1 samples are used as negative control $\mathbf{E}^-$ and around 75% of the stage 3 samples are used as positive control $\mathbf{E}^+$. These controls are used to learn distance metric. The remaining 25% of both stage 1 samples $\mathbf{E}_1^?$ and stage 3 samples $\mathbf{E}_3^?$, and all of the stage 2 samples $\mathbf{E}_2^?$, are then used as test groups to evaluate the learned metric. This process was repeated 4 times and averaged. For all three data sets, the constrained sample pairs used were formulated by the samples within negative controls $\mathbf{E}^-$ and positive controls $\mathbf{E}^+$. The lower and upper bounds of the right hand side of the constraint (l and u) in equation 4.5 are the 5^{th} and 95^{th} percentiles of the observed distribution of distances between pairs of samples within each data set.

5. Distance Measures Application Results and Discussion

5.1 Binary Severity Detection Results

The results obtained using three binary severity estimation approaches we have studied(Chapter 2) – Hierarchical Clustering Analysis (HCA), Principal Component Analysis (PCA) followed by K-means clustering and Decision Tree Classification (DTC) – are consistent in distinguishing the severity of CRS between Anti-CD28 SA from that of other mAbs.

5.1.1 Hierarchical Clustering Analysis

Hierarchical Cluster Analysis was applied to the data shown in Table 2.4 and Table 2.5. The resulting dendrograms are shown in Figure 5.1 and Figure 5.2. The dendrograms represent the cluster hierarchy of the dataset; the horizontal axis is the standardized Euclidean distance between pairs of clusters. Both dengrograms show that the distances between the Anti-CD28 SA cluster and all other mAbs clusters are quite large, indicating that the average response for Anti-CD28 SA is very different from that of the other mAbs. Figure 5.1 shows the HCA for Anti-CD28 SA and controls only, indicating clear separation between the means of both sets of data. Figure 5.2 shows the HCA for Anti-CD28 SA, the controls, and the 6 mAbs from the "Safe" class. Again, we see clear separation between Anti-CD28 SA and the other treatments. In addition, the dendrogram in Figure 5.2 appears to separate the controls and mAbs in the "Safe" class into two separate clusters with Tocilizumab, IL-5, Palivizumab, CD22, and CD80 in one cluster and PBS, AutoPlasma and IL-1β in the other cluster. However, the method does not provide direct explanation as to why these treatments from the "Safe" class form two separate clusters and how the

treatments within each cluster are related.

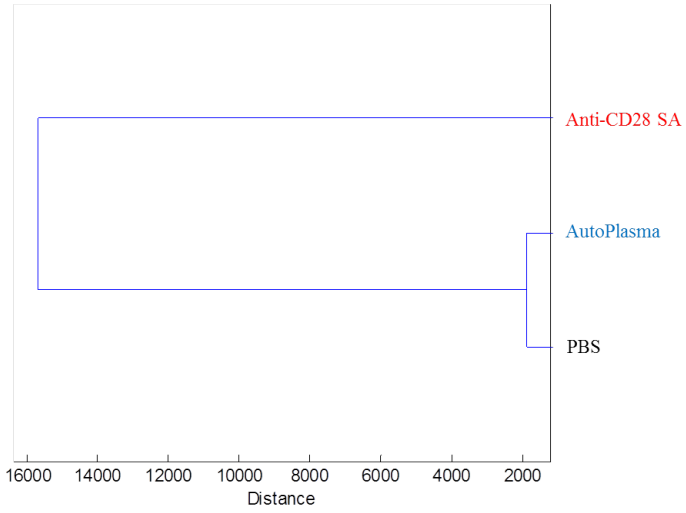


Figure 5.1: HCA dendrogram for Anti-CD28 SA, AutoPlasma and PBS

5.1.2 Principal Components Analysis (PCA) with K-means Clustering

Principal Components Analysis (PCA) was applied in order to reveal the internal structure of the data in a way that best explains the variance in the data. It is well documented that in order to be successful, the attributes subject to PCA need to have similar dynamic ranges [54, 58]. However, the attributes from our dataset have a large variation in their dynamic ranges, as shown in column 4 of Table 2.1. As shown in Figure 5.3, our attributes also exhibit a nearly linear relationship between the mean and the standard deviation for samples of different treatments. This property allows us to reduce the differences in the dynamic ranges of the attributes by using the logarithmic transformation [58, 59, 60]. This transformation replaces each sample

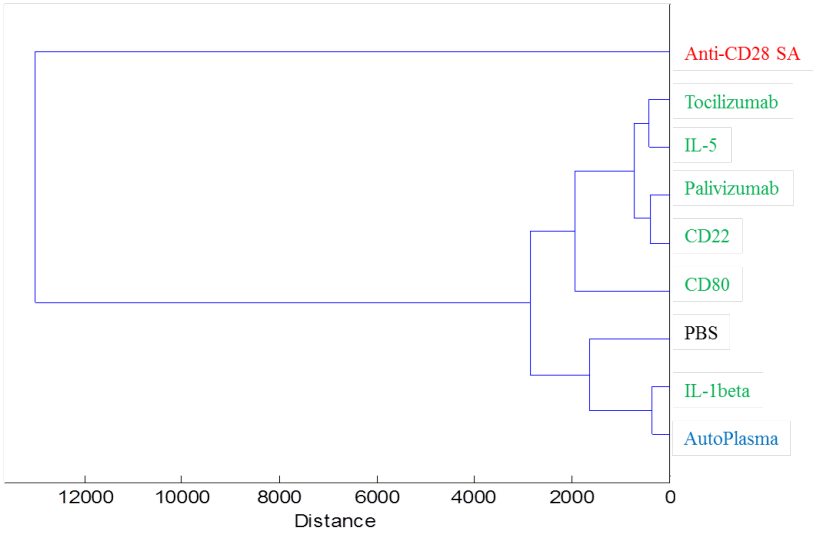


Figure 5.2: HCA dendrogram for all the treatments

data value by its logarithm (base 10) before applying PCA.

We applied PCA to the samples from Anti-CD28 SA and the controls in Table 2.4 and sorted the Principal Components in descending order by the amount of variance associated with them. The variances associated with each of the Principal Components are listed in Table 5.1, where we see that the first three Principal Components account for more than 90% of the variance of the data. This observation suggests that we can possibly ignore the rest of the principal components, with little impact on the underlying structure of the data. We then represented the data graphically using a three-dimensional scatter plot against the three principle components (Figure 5.4(a)).

Principal Components Analysis (PCA) helps with visualizing but does not cluster the data points in the new coordinate system. In order to identify sample populations in the new attribute space, we used K-means clustering. The technique requires that we specify the number of clusters into which observations will be assigned before the clustering process starts. We used the previous HCA to determine the number of

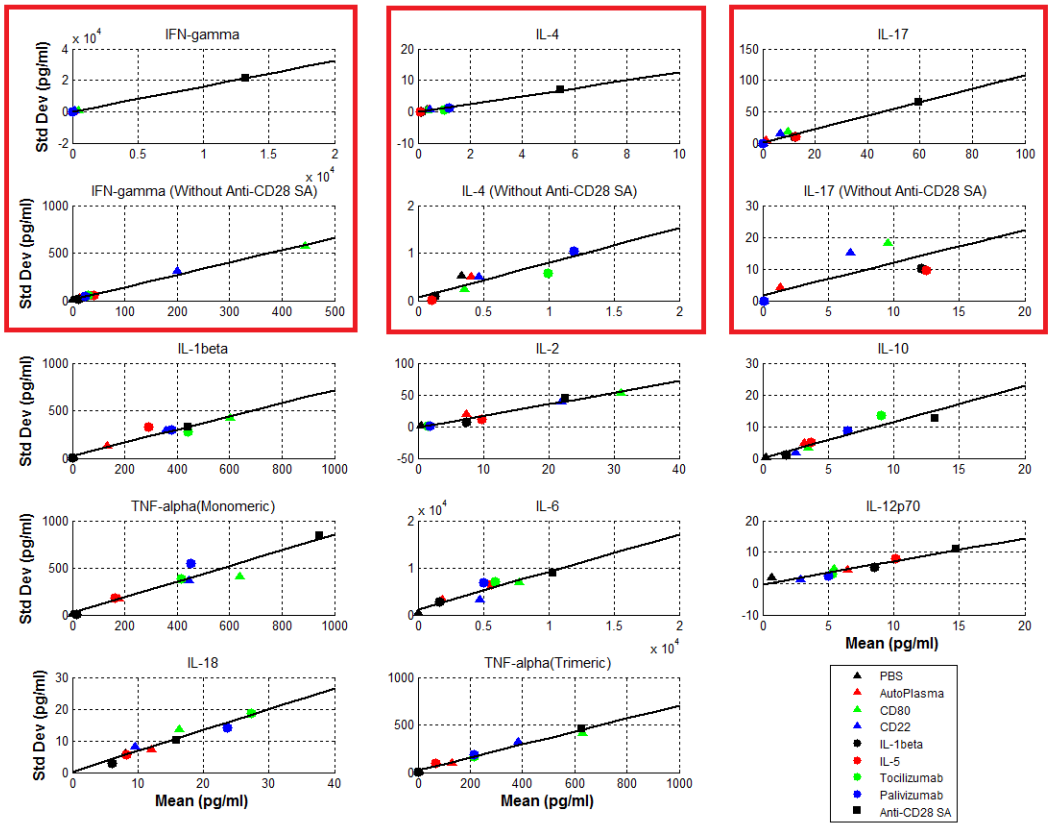


Figure 5.3: Standard deviation vs. means for all cytokines showing the least squares linear approximation for all points (IFN-γ, IL-4 and IL-17 were plotted with and without Anti-CD28 SA to confirm the linear relationships)

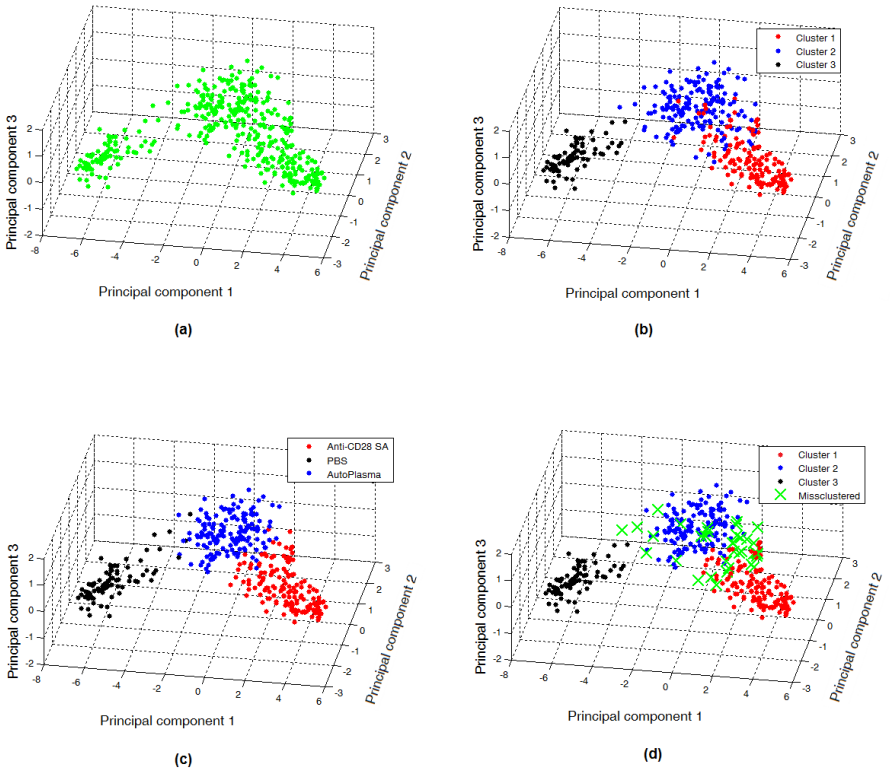


Figure 5.4: (a) Graphical representation of the data using the first three principal components of PCA (b) K-means clustering results based on the first three principal components (c) Visual representation of the data using the known labels to identify populations after applying PCA (d) K-means clustering showing misclassified samples

clusters, which in this case is 3. Figure 5.4(b) shows the samples corresponding to each cluster in different colors, and Table 5.2 shows the number of samples in each cluster and the label associated with them. It is notable that Cluster 1 consists mostly of Anti-CD28 SA samples. Clusters 2 and 3 are dominated by AutoPlasma and PBS samples, respectively. These results confirm that samples from Anti-CD28 SA and the two controls can be separated from each other using this method.

Table 5.1: Variance associated with principal components for Anti-CD28 SA, PBS and AutoPlasma

Variance included in 11 Principal Components (PC)			
PC 1	PC 2	PC 3	PC 4-11
77.60%	10.90%	3.10%	8.40%
91.60%			8.40%

Table 5.2: The number of samples for each treatment in each cluster found by K-means clustering on the data using the first three principal components

	CD28	AutoPlasma	PBS
Cluster 1	132	10	0
Cluster 2	20	142	5
Cluster 3	0	0	75

Applying the known class labels of the samples to visualize the data as shown in Figure 5.4(c), we see that each sample population, Anti-CD28 SA and the two controls, is located in a different region within the three-dimensional space. Using PCA we are thus able to graphically represent the differences in the cytokine responses for each sample, and can observe how samples from a given population are grouped

in a specific region. These observations are consistent with the dendrogram from Figure 5.1 in which the differences between the mean response of Anti-CD28 SA and the controls are highlighted. Using PCA we can not only corroborate this separation, but also see the differences between samples. Finally, using the information from the labels associated with each sample and the results obtained from K-means clustering, we can identify and visualize the misclassified samples. Figure 5.4(d) shows the misclassified samples. Not surprisingly, they are located on the boundaries of the established cluster.

Next, we repeated the analysis (PCA with K-means clustering) using the samples from all treatments shown in Table 2.5. This analysis would attempt to classify all treatment samples, not just the Anti-CD28 SA, AutoPlasma, PBS sample set. The variances associated with each of the Principal Components are listed in Table 5.3. Again, the first three principal components accounted for more than 90% of the variance of the data; we used these three principal components to create a three-dimensional scatter plot of the samples from Anti-CD28 SA, "Safe" treatments, and controls (Figure 5.5(a)).

Table 5.3: Variance associated with principal components for all treatments

Variance included in 11 Principal Components (PC)			
PC 1	PC 2	PC 3	PC 4-11
75.20%	11.90%	3.20%	9.70%
90.30%			9.70%

After applying PCA, we applied K-means clustering to the results of PCA, considering the first three Principal Components. As before, we used HCA to guide us as to the number of clusters. The dendrogram in Figure 5.2 suggested 3 clusters, which

we have used to run K-means clustering on the transformed data. The clustering results are shown in Figure 5.5(b) and are detailed further in the Table 5.4. The table shows the three clusters and their constituents. The first cluster consists mostly of Anti-CD28 samples; the second cluster consists mostly of samples from AutoPlasma, and the third cluster consists mostly of samples from PBS.

Table 5.4: The number of samples for each treatment in each cluster found by K-means on the data using the first three principal components

	CD28	AutoPlasma	PBS	CD80	CD22	IL-1	IL-5	Tocilizumab	Palivizumab
Cluster 1	132	12	0	2	2	2	4	0	0
Cluster 2	20	140	4	6	6	0	4	8	8
Cluster 3	0	0	76	0	0	6	0	0	0

The samples from CD28 and the two controls (AutoPlasma and PBS) can again be distinguished from one another using this technique. Most samples from the "Safe" treatments are clustered with the controls, indicating the relative similarity of response between the controls and the safe treatments. The small number of the "Safe" and control samples makes it impractical to cluster these samples separately. The results of this analysis are consistent with the results from HCA and the observations made on PCA with labeled data. When we use the known labels of the samples to visualize the data after PCA is applied (as shown in Figure 5.5(c)), we see that most of the samples from the "Safe" treatments were placed closer to the control samples than to the Anti-CD28 SA samples. This observation is consistent with the dendrogram obtained from the HCA analysis in Figure 5.2.

5.1.3 Decision Tree Classification (DTC)

Using HCA and PCA with K-means clustering we can automatically separate samples from Anti-CD28 SA and other treatments, and visualize the data in a three-

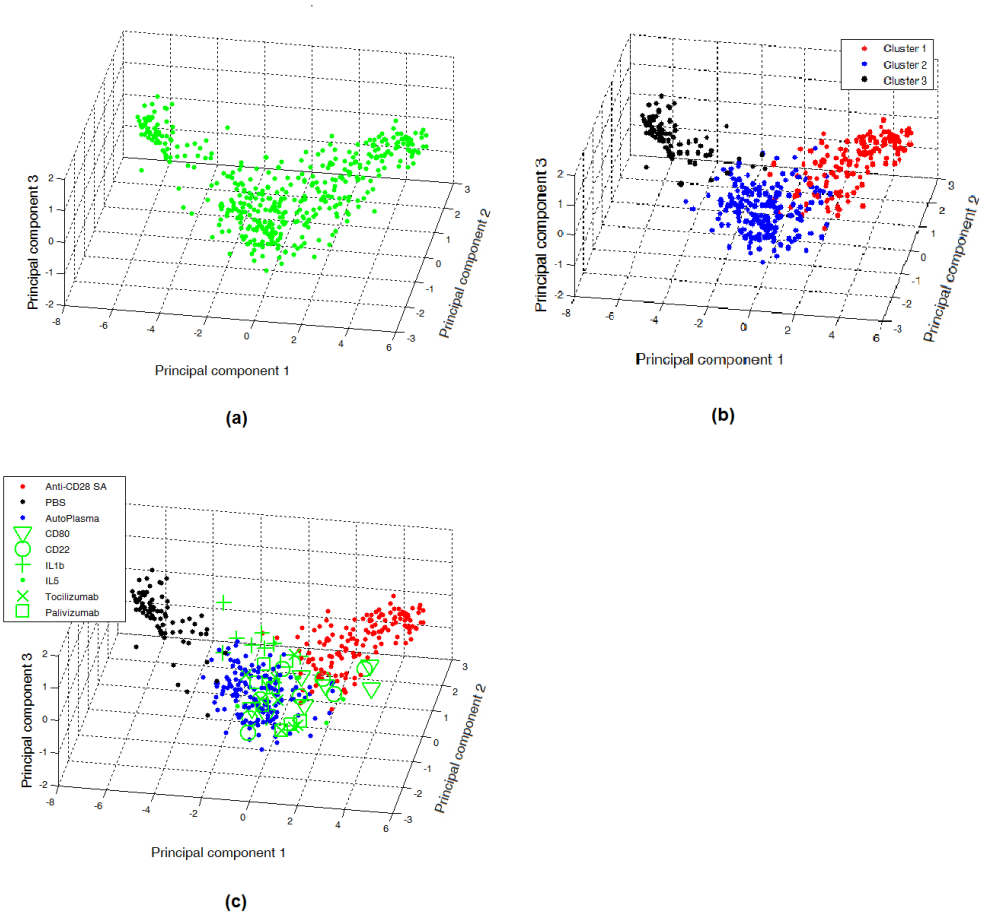
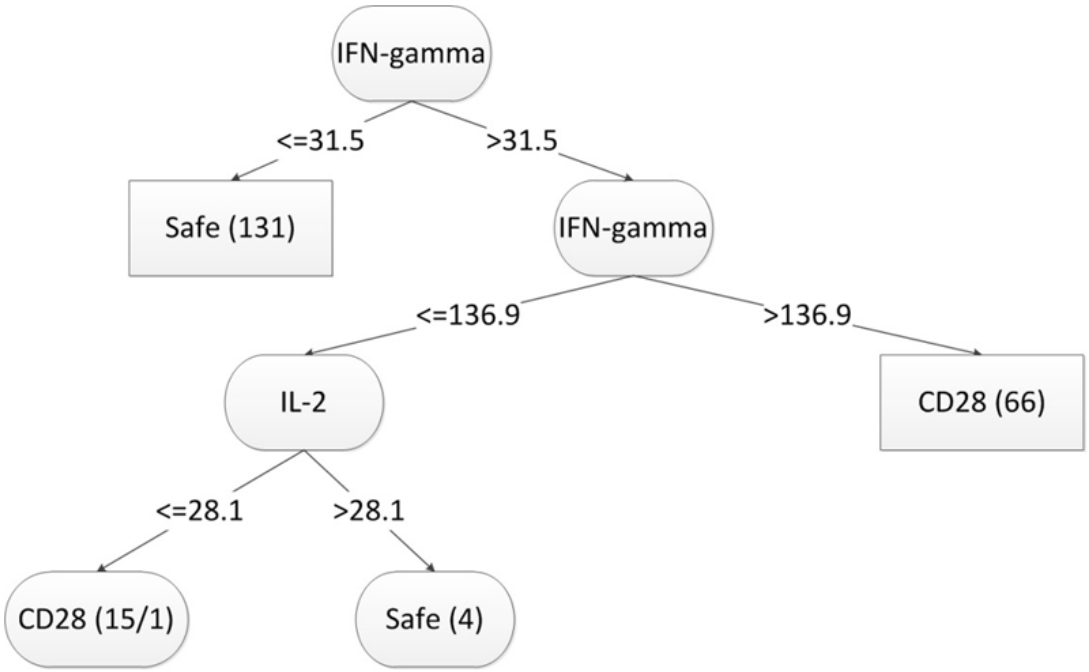


Figure 5.5: (a) Data (all treatments) representation based on principal components after selecting the three first Principal Components (b) K-means clustering results based on the first three principal components (c) Representation of the data based on the labels known for each sample

dimensional space. However we still have not determined how the different cytokine releases led to this separation. To add this dimension to the study, we have employed Decision Tree Classification (DTC). The two training data sets described in Tables 2.4 and 2.5 were used to train the DTC algorithm. Training data set 1 included 136 samples in the "Safe" class and 80 samples in the "CD28" class, for a total of 216 samples (Table 2.4). The cytokine that best separated the two classes among the 11 measured cytokines was selected first by the DTC, and the selection process was then repeated with the next best cytokine, etc. The tree model shown in Figure 5.6(a) was built using training data set 1. As can be seen from the root node, the cytokine which produced the first split of data was IFN-γ. The 131 samples in this data set with values of IFN-$\gamma \leq$ 31.5pg/ml were classified into the "Safe" class while the 66 samples in this data set with values of IFN-γ greater than 136.9pg/ml were classified into the "CD28" class. Samples with values of IFN-γ between 31.5pg/ml and 136.9pg/ml required further analysis using the IL-2 node to identify the corresponding classes. Values of IL-2 $\leq$ 28.1pg/ml were in the "CD28" class, and values of IL-2 greater than 28.1pg/ml were in the "Safe" class. On the left branch of IL-2 node, the notation "(15/1)" indicates that of the 15 samples assigned to the "CD28" class, one (1) sample was misclassified. The accuracy of this tree model in classifying new data correctly is estimated by the cross validation procedure described in Chapter 2.7 (In addition, 10-fold cross validation was also used, and the results are discussed in Appendix A). A square Confusion Matrix (CM) is used here to record the performance of the DTC model. Each column of the CM represents the instances of the predicted class. Each row represents the instances of the actual class. $CM_{ij}$is the number of samples in the true class i which were assigned to predicted class j. Ideally CM is a diagonal matrix.

Cross validation was performed 100 times. The average accuracy after 100 times

was 95.5%; the corresponding Confusion matrix is shown in Figure 5.6(b). The false alarm or false positive rate of the DTC model was 5.1% and the misdetection or false negative rate was 3.8%. The classification accuracy obtained from the cross validation method is high, indicating that the developed model was reliable in this case in terms of its ability to classify samples from unknown donors.



(a) Decision Tree model

		Classifier Outcome		Classifier accuracy (95.5%)
		Safe	CD28	
Known	Safe	129	7	False alarm/False positive 5.1%
	CD28	3	77	Misdetection/False negative 3.8%

(b) Confusion Matrix for cross validation

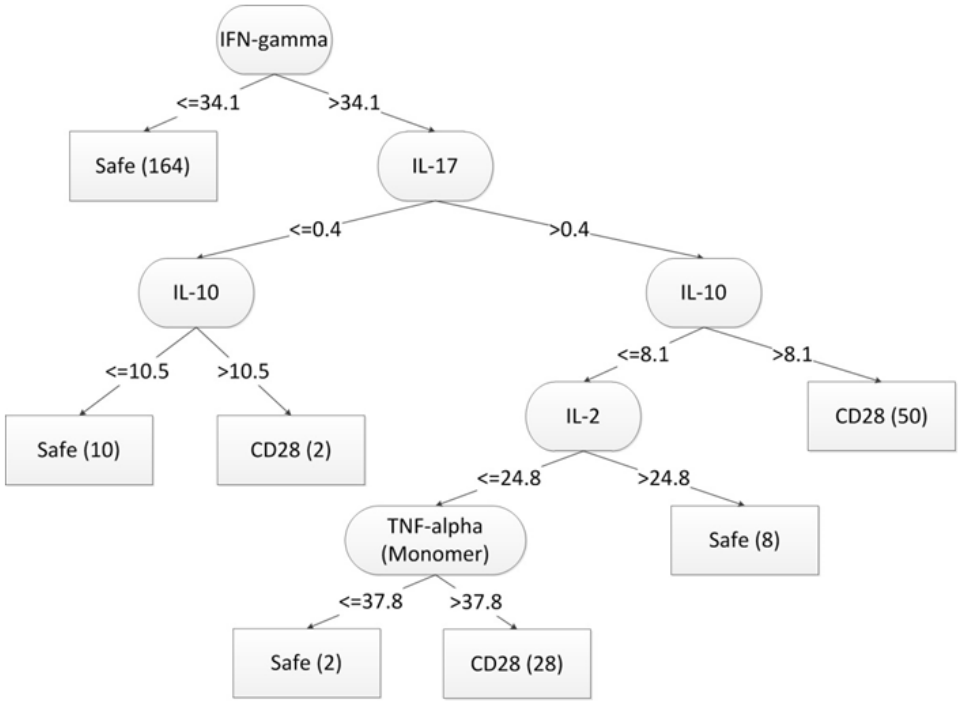
Figure 5.6: (a) Decision Tree model using training data set 1 with 11 cytokines (b) The confusion matrix shows the performance of the cross validation

Training data set 2 included 184 samples from the "Safe" class and 80 samples from the "CD28" class, for a total of 264 samples (Table 2.5). The tree model and its Confusion Matrix are shown in Figure 5.7. The average accuracy of the classification for this tree model was 93.1% for 100 iterations of the random donor splitting method. The Confusion Matrix is shown in Figure 5.7 (b). The classification rules can be inferred from the model in the same manner as for training data set 1. Here we see that IL-17, IL-10 and TNF-α (monomeric) are used in addition to IFN-γ and IL-2 to classify the samples.

The DTC model generated using training data set 2 is shown in Figure 5.7(a), where we observe that the root node was IFN-γ. It produced the first data split identifying 164 samples in the "Safe" class with values of IFN-γ $\leq$34.1pg/ml and 100 samples with values of IFN-γ $\leq$34.1pg/ml that required further classification. The cytokine IL-17 was used as the second cytokine for separation, suggesting that it is relevant for classification, although IL-17 release is not commonly measured in CRS assays (e.g., [56][72]).

Both DTC models identified IFN-γ as the most relevant cytokine for classification. Next, we have removed IFN-γ samples from the dataset, and ran our models with the remaining 10 cytokines. The results are shown in Figure 5.8, where both models identify IL-17 as the root node. The resulting tree structures are more complex than the previous analysis with IFN-γ as the root node; however, the classification accuracies are still relatively high at 90.7% and 95.1% for training datasets 1 and 2, respectively. These results indicate that IL-17 may also be relevant for classification of these cytokine data.

In order to assess the ability of the DTC models (shown in Figure 5.6(a) and Figure 5.7(a)) to analyze new data, we used test sets for the models that did not included any of the data in the training set. Specifically, we used the (hitherto unseen)

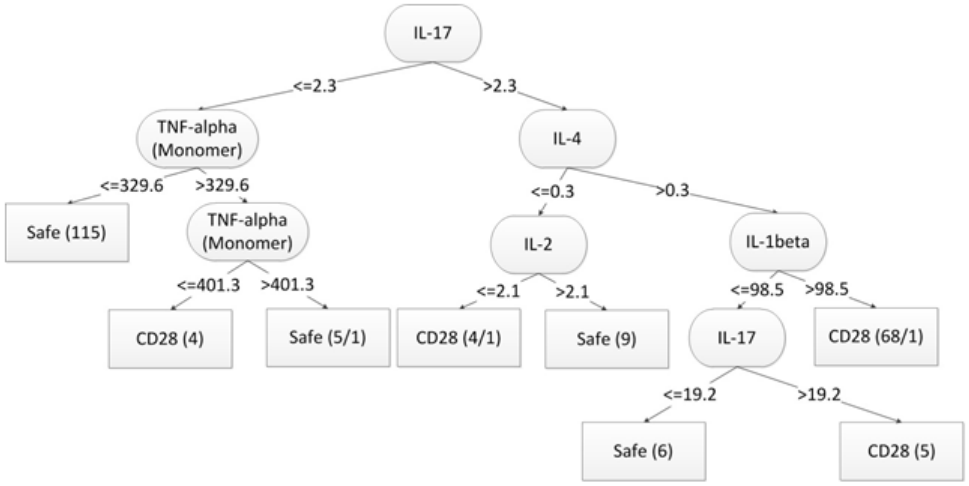


(a) Decision Tree model

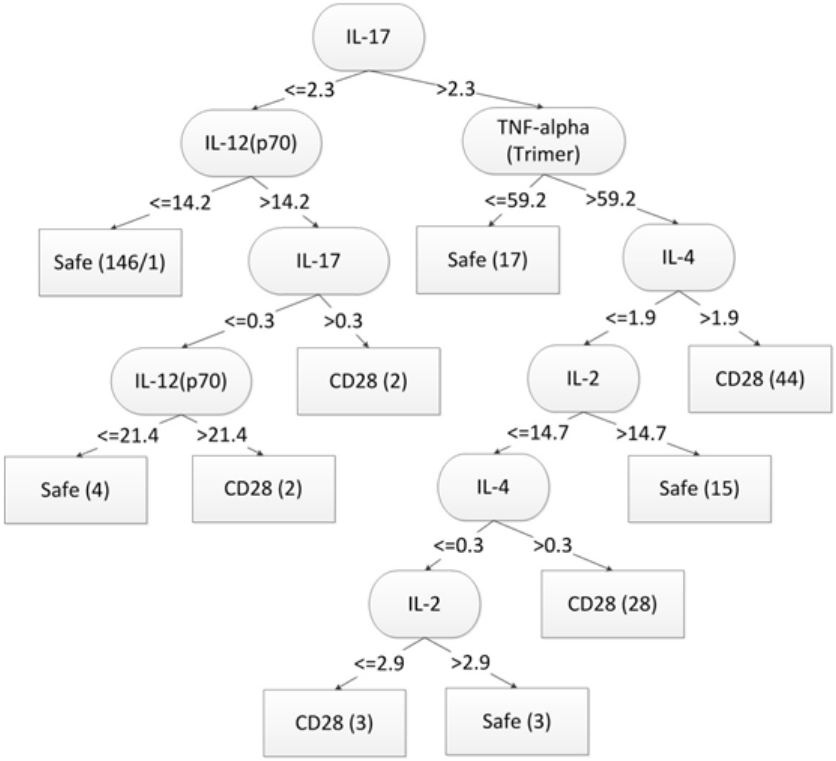
		Classifier Outcome		Classifier accuracy (93.1%)
		Safe	CD28	
Known	Safe	173	11	False alarm/False positive 6.0%
	CD28	7	73	Misdetection/False negative 8.75%

(b) Confusion Matrix for cross validation

Figure 5.7: (a) Decision Tree model using training data set 2 with 11 cytokines (b)The confusion matrix shows the performance of the cross validation



(a) Decision Tree model using Training data set 1



(b) Decision Tree model using Training data set 2

Figure 5.8: In order to verify the importance of IL-17, IFN-γ has been removed from training data set 1 and 2. DTC was applied to the remaining 10 cytokines and the two tree models are: (a) Tree model corresponding to training data set 1 (b) Tree model corresponding to training data set 2. Both two tree models show IL-17 as the root node.

test data described in Table 2.6. The accuracy of the two models in Figure 5.6(a) and Figure 5.7(a) are provided in Table 5.5 and Table 5.6, respectively. The accuracy percentages are 92.9% for the DTC in Figure 5.6(a) and 93.7% for the DTC in Figure 5.7(a).

Table 5.5: Test results for the tree model in Figure 5.6(a)

		Classifier Outcome		Test accuracy (92.9%)
		Safe	CD28	
Known	Safe	63	9	False alarm/False positive 12.5%
	CD28	0	54	Misdetection/False negative 0%

Table 5.6: Test results for the tree model in Figure 5.7(a)

		Classifier Outcome		Test accuracy (92.9%)
		Safe	CD28	
Known	Safe	72	0	False alarm/False positive 12.5%
	CD28	8	46	Misdetection/False negative 0%

5.1.4 Discussion

The results obtained from the analysis of our dataset, using the three machine learning approaches, are consistent in discriminating between Anti-CD28 SA and other mAbs with respect to CRS. However, each approach provides different information about the studied data set.

HCA highlights the differences between the treatments using the means of their cytokine response. It also provides a simple cluster hierarchy of the treatments, e.g.,

in Figure 5.2 we identified three clusters corresponding to Anti-CD28 SA, controls, and the other treatments in the "Safe" class. These findings are consistent with the results from Walker et al. [18] and give guidance on how many clusters may exist in the data. On the other hand, HCA does not allow classification of individual samples from the dataset since it uses the average response for each mAb as the basis for the measure of "distance" in order to find the cluster hierarchy. Using HCA it is not clear whether some cytokines have more discriminatory relevance than others.

PCA was used in this study to build scatter plots to visualize the data sample-by-sample. The first three principal components that account for most of the variance were used to visualize the data, showing where each sample is located in the suggested principal components space. We used K-means clustering to further analyze the PCA results[1] seeking three clusters, which was suggested by HCA. Samples from Anti-CD28 SA were placed far away from the other samples, and samples from the treatments in the "Safe" class were closer to PBS and AutoPlasma than they were to Anti-CD28 SA. However, PCA and K-means clustering provided, in addition to the information provided by HCA, a method of clustering actual samples rather than clustering the means of samples. The resulting classification can also be used to assign unlabeled samples to a particular cluster, based on the distance of the sample from the centroid of the cluster. The principal components used in this approach do not have a direct biological interpretation.

DTC fills this biological interpretation gap. As with HCA and PCA, DTC can also distinguish Anti-CD28 SA from all other treatments and controls, but in addition it provides information about which cytokines are most relevant to the separating of

[1]We applied K-means clustering to both raw data and the outcomes of PCA on the raw data. The clustering error rates were comparable (8%-10% on average in both cases). However, using PCA gave a clear low-dimensional representation of the data. Furthermore, we could visualize the results of K-means clustering using the 3 principal components from PCA, which could not be done directly using the 11-dimensional raw data.

Anti-CD28 SA. The rules created by DTC specified thresholds that separated cytokine release levels into groups. These groups can then be associated with the "Safe" and the "CD28" classes. For example, in the DTC model in Figure 5.6(a), when the value of IFN-γ is greater than 136.9pg/ml, the model classifies 66 samples into the "CD28" class. The value "136.9pg/ml" is a threshold chosen by iteratively calculating information gains for threshold candidates associated with each cytokine and using the value with the highest information gain (in [62], Section 6.1).

In addition to ease of interpretation, DTC shows other advantages in analyzing biological data. First, once the tree model is constructed from a training data set, it can easily be used to classify unknown samples as we demonstrated using the test data set in Table 2.6. Our results also show that DTC provided high classification accuracy, and required relatively little effort from users for data preparation. Similarity conclusions were reported in [97], which compared several machine learning algorithms in cancer tissue classification , and in [98], which studied the use of Decision Tree based classification of uncertain data.

Our study needs to be viewed in the context of several other investigations of cytokine release through machine learning approaches [29, 99, 100, 101, 102, 103, 104, 105]. Collectively these studies point to the usefulness of applying the three machine learning approaches studied here (HCA, PCA, K-means clustering and DTC) to mAb induced cytokine release data from a variety of assays and conditions. In order to apply these approaches to a new assay, the following requirements should be met. First, there should be some labeled reference data from mAbs with known CRS potential. In some cases we may know the CRS potential of all the mAb and controls. In other cases, when new mAbs with unknown CRS potential are studied, we could measure their similarity to the known mAbs or controls to infer the CRS potential of the new mAbs. Second, enough samples should be collected to get useful

and consistent results. This has been done, in our case, by starting with a "reasonable" number of samples (5-10), categorizing the samples using the machine learning approaches, then repeating with about ten more samples each time. Once the results started stabilizing (which in our case was at about 120 samples), we concluded that enough samples have been used (see Appendix D).

Prior studies have shown that CRS is mainly related to high levels of IFN-γ, TNF-α, IL-6, IL-2, IL-8, and IL-10 [56][72]. The analysis of TGN 1412 treated patients shows that TNF-α was increased at 1 hour after infusion and IFN-γ, IL-2, IL-6 and IL-10 were increased after 4 hours [106]. In addition, recent work using mathematical modeling suggests that there is a cause and effect relationship among some of the cytokines [107]. Serum collected from the TGN 1412 trial showed that INF-γ and TNF-α are the first cytokines to be produced and from the mathematical modeling INF-γ is thought to subsequently induce IL-10 and IL-6, whereas, TNF-γ is thought to induce IL-8 and IL-10. In fact, our DTC models identified IFN-γ, TNF–α and IL-10 as important cytokines in CRS, corroborating these findings.

Our DTC analysis suggested IL-17 as a potentially important cytokine in classifying treatments as "Safe" class or "CD28" class. IL-17 has not been identified so far in the literature as an important cytokine in the context of CRS. Literature reports of cytokine analysis of the TGN 1412 trial did not measure IL-17. In the published literature, this cytokine has not been measured in assays developed for detecting mAb-induced CRS. Since IL-17 is a pro-inflammatory cytokine, it should contribute to CRS since it has been shown to stimulate a highly pro-inflammatory gene signature [108]. It induces NF-Kβ [109], a transcription factor historically known for fast-acting pro-inflammatory cellular responses [110]. IL-17 is also known to induce IL-6, IL-8, G-CSF, and prostaglandin E2 production by mesenchymal cells and cause accumulation of neutrophils in the blood and tissues [111]. This IL-6 production could act

on acute phase proteins and lead to inflammation. Moreover, IL-17 has been shown to increase the effects of TNF-α partially by increasing Tumor Necrosis Factor Receptor 2 (TNFR2) or Lipopolysaccharide-Induced CXC Chemokine (LIX) [111, 112], possibly contributing to the CRS cascade, which is thought to start with TNF-α and IFN-γ. IL-17 is thought to be important in chronic inflammatory conditions such as autoimmune diseases, transplantation, and infections [113, 114]. In addition to IL-17 being pro-inflammatory in nature, IL-17 has been shown to be involved in other forms of cytokine storm including those induced by bacteria and transplant [115, 116]. Therefore, it is not surprising that our DTC model was able to identify IL-17 as an important cytokine in detecting CRS even though this correlation has not been noted so far in the literature.

To conclude, we used different approaches to analyze data from an in vitro assay that uses human blood to assess the potential of CRS from different mAbs. All of the approaches were able to identify the treatment that caused the most severe cytokine response. Additionally, PCA and K-means clustering allowed classifying treatments sample by sample and visualizing them in a low dimensional space. DTC models showed the relative importance of various cytokines such as IFN-γ, TNF-α and IL-10 to CRS. The combined use of the techniques provided a more comprehensive view of the data and better-informed processes for selection of parameters and thresholds.

5.2 Severity Estimation using Distance Metric Learning Results

5.2.1 Cytokine Release Syndrome Data Set Results

The analyses of Cytokine Release Syndrome (CRS) data set by Severity Estimation using Distance Metric Learning (SE-DML) have two steps: (1) group severity level estimation; (2) individual severity level estimation.

The three groups – safe($\mathbf{E}_1^?$), middle($\mathbf{E}_2^?$) and severe($\mathbf{E}_3^?$) listed in Table 3.2 – are

tested using the distance metric learned from positive controls ($\mathbf{E}^+$: Anti-CD28 SA and LPS) and negative controls ($\mathbf{E}^-$: PBS and AutoPlasma). The severity level of each group estimated by SE-DML is a normalized value between 0 and 1 showing in Table 5.7. These levels have an order of $y_1 < y_2 < y_3$, indicating that these levels matched their group labels. These results illustrate that the SE-DML approach can correctly estimate the severity of the treatment groups.

Table 5.7: Severity levels for three treatment groups in CRS data set

Class	Severity level y_i
Safe ($\mathbf{E}_1^?$)	0.081±0.0143
Middle ($\mathbf{E}_2^?$)	0.121±0.0257
Severe ($\mathbf{E}_3^?$)	0.482± 0.1106

The 26 treatments listed in Table 3.2 are also estimated using the distance metric learned from positive controls ($\mathbf{E}^+$: Anti-CD28 SA and LPS) and negative controls ($\mathbf{E}^-$: PBS and AutoPlasma). The ideal results would be the 8 treatments in the severe group have higher values than that of the treatments in the other two groups. All the 6 treatments in the middle group have severity levels right in between that of the treatments in the other two treatments. The 11 treatments in the safe group have the lowest severity levels. But the real results are much more complex than the ideal condition. Several treatments have severity levels mixed with treatments within other groups. The severity levels of the 26 treatments shown in Figure 5.9 are sorted in a descending order from the top to the bottom, with higher treatments indicating more severe CRS reactions. Almost all the treatments in the severe group have a higher severity level than treatments in the other two groups except CD40, which stays in the middle of Figure 5.9. Two safe treatments, IL-6, and CD80 and one from

the middle class (Anti-VEGF) are mixed with treatments in the severe group. The middle group treatments are mostly mixed with safe treatments but have relative higher severity levels than safe treatments.

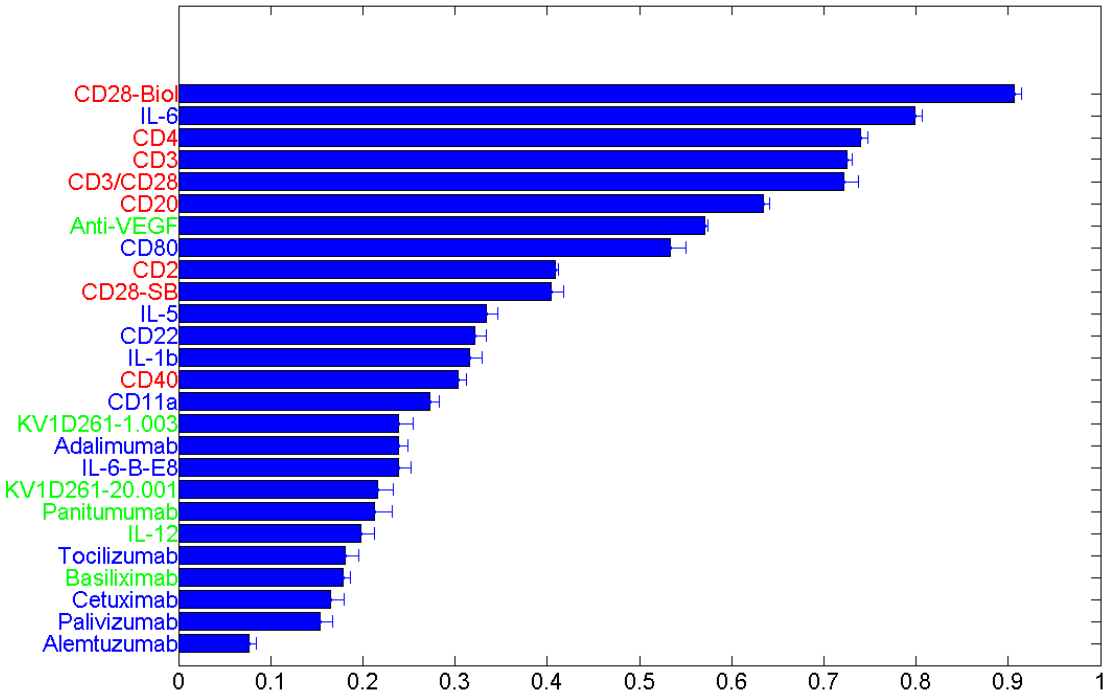


Figure 5.9: Severity levels of 26 test treatments in CRS data set. The standard deviation of the estimation is shown as error bar in the figure. Red treatments are in severe-CRS group, green treatments are in middle group and blue treatments are in safe class.

5.2.2 Cardiotocography Data Results

Based on the data set described in Chapter 3.7, the learned A matrix is a 21 by 21 positive definite matrix. The severity levels over 10 iterations are shown in Table 5.8. They indicate that the severity level of $\mathbf{E}_1^?$ is almost the same as the

negative control $\mathbf{E}^-$ and the severity level of $\mathbf{E}_3^?$ is very close to the positive control $\mathbf{E}^+$. The severity levels of $\mathbf{E}_1^?$ and $\mathbf{E}_3^?$ illustrate that the proposed approach works here because $\mathbf{E}_1^?$ and $\mathbf{E}^-$ are from the same class in the CTG data set, as well as $\mathbf{E}_3^?$ and $\mathbf{E}^+$. Moreover, without any prior information about $\mathbf{E}_2^?$, the proposed approach still ranked the severity of $\mathbf{E}_2^?$ to the right position – between $\mathbf{E}_1^?$ and $\mathbf{E}_3^?$.

Table 5.8: Average severity levels of three classes in test data and their standard deviations over 10 fold cross validation

Class	Severity levels y_i
Normal ($\mathbf{E}_1^?$)	0.003±0.002
Suspect ($\mathbf{E}_2^?$)	0.251±0.058
Pathologic ($\mathbf{E}_3^?$)	0.905±0.070

5.2.3 Quantitative Structure Activity Relationship (QSAR) Results

Based on the data set described in Chapter 3.8, the learned A matrices are (1) a 27 by 27 positive definite matrix and (2) a 60 by 60 positive definite matrix respectively according to the different number of attributes. The average severity levels of the two data sets over 20 times are shown in Table 5.9. In both data sets, $\mathbf{E}_1^?$ is the closest to $\mathbf{E}^-$ and it has the lowest severity level among the three sample groups; $\mathbf{E}_3^?$ is the closest to $\mathbf{E}^+$ and it has the highest severity level among the sample groups; The severity levels of $\mathbf{E}^-$ falls right in between the other two sample groups. These severity levels indicate that with only two extreme sample groups, the SE-DML approach was able to rank the severity of three middle sample groups correctly in this case.

Table 5.9: Average severity levels of test data in the two QSARs data sets and their standard deviations

Sample Group	Severity levels y_i of Pyrimidines	Severity levels y_i of Triazines
$\mathbf{E}_1^?$	0.134± 0.063	0.720±0.123
$\mathbf{E}_2^?$	0.437± 0.193	0.755±0.096
$\mathbf{E}_3^?$	0.659± 0.220	0.902±0.044

5.2.4 Algorithm Comparison

The estimated severity levels of the four data sets by the 5 approaches are shown in Figure 5.10. The four bar charts (row-wise) represent results of the four data sets. In each chart, there are five sets of bars representing the severity levels (y_1, y_2 and y_3) of the three test groups $\mathbf{E}_1^?$, $\mathbf{E}_2^?$ and $\mathbf{E}_3^?$ estimated by the 5 approaches. The blue bar is the severity level y_1 of $\mathbf{E}_1^?$, the green bar is the severity levels y_2 of $\mathbf{E}_2^?$, and the red bar is the severity levels y_3 of $\mathbf{E}_3^?$. Each y_i is the mean of sample severity levels within $\mathbf{E}_i^?$. The standard deviation is shown as the error bar in the figure.

For each set of bars, we ignore the absolute difference between y_1, y_2 and y_3 but only evaluate relative order of these three severity levels. We consider the relative order of $y_1 < y_2 < y_3$ as the correct severity estimation since it matches the true group label. For CRS data set, only SE-DML can correctly estimate the relative order among the severity levels of the three test groups. For CTG data set, all the 5 approaches correctly estimate the relative ordering. For Pyrimidines data set, linear regression fails to distinguish the severity levels of $\mathbf{E}_1^?$ and $\mathbf{E}_2^?$ and the results show a large standard deviation, indicating it can not estimate the severity of Pyrimidines data set robustly. For Triazines data set, the SE-DML approach is the only one that can identify correctly the relative order of $y_1 < y_2 < y_3$.

According to the relative orders among the predicted group-level severity levels, our SE-DML approach has achieved the best performance among all the 5 approaches.

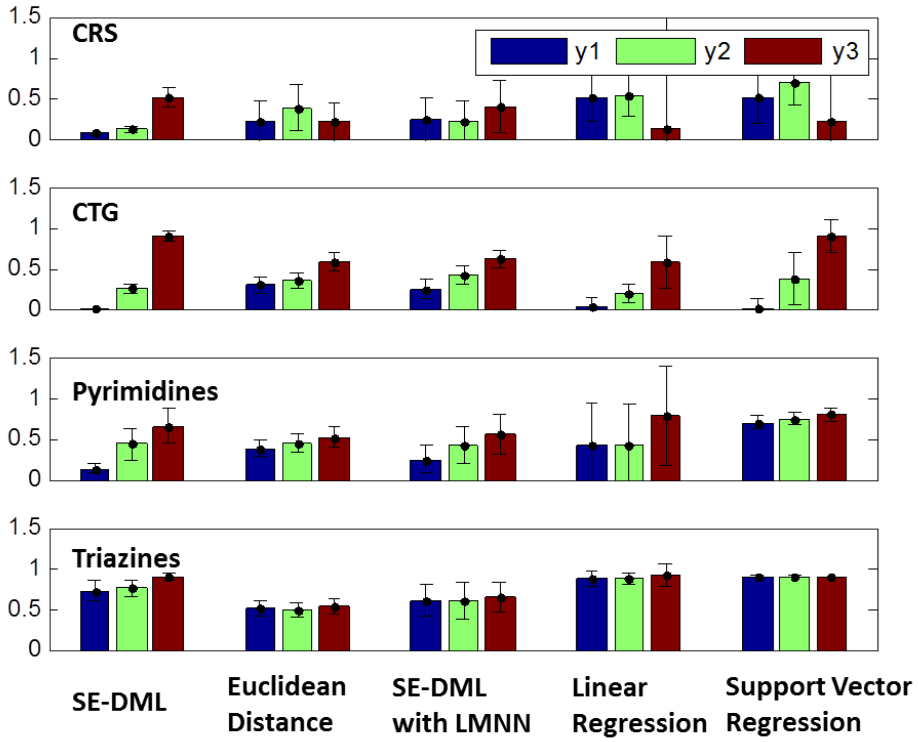


Figure 5.10: Severity estimation results of the 5 approaches on four data sets. Each bar chart presents the estimated severity levels of one data set.

However, this approach could not capture the sample-level severity estimation since only using average severity level representing a group of samples may not be sufficient enough to illustrate the effectiveness of SE-DML approach. We need to evaluate how well the individual sample's severity level matches to other samples within this sample's group. Silhouette coefficients provide a numerical measure about this evaluation. For each data set, there are three test groups giving three clusters of individual sample's severity levels. Higher silhouette coefficient of a sample indicates its severity level is well-matched to its own group, when compared to severity levels of samples in other groups. The average silhouette coefficient of the 5 approaches for four data sets are listed in Table 5.10. The bold number in each row indicates the best silhouette

coefficient of each data set. The SE-DML approach has the best silhouette coefficient in 3 out of the 4 data sets.

Table 5.10: Silhouette coefficients of the 5 approaches

	SE-DML	**SE-DML with Euclidean Distance**	**SE-DML with LMNN**	**Linear Regression**	**Support Vector Regression**
CRS	-0.1114	**-0.0821**	-0.1376	-0.3654	-0.1015
CTG	**0.7362**	-0.0271	0.2987	0.2905	0.6657
Pyrimidines	**0.1293**	0.0063	0.0905	-0.0753	0.0116
Triazines	**0.1186**	-0.0135	0.0302	-0.2814	0.1128

5.3 High-dimensional Cancer Tissue Data Classification Results

5.3.1 Sample-level Cancer Tissue Classification Results

Cross Validation Accuracy of Variations of Neighbor Sizes in K-nearest Neighbor (KNN) Classification

In KNN classification, nearest neighbor size k is a user-defined parameter and the choice of k is very critical to the classification performance. The optimal value of k is based on the cross validation accuracy of training samples. Generally, the larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct [117]. Figure 5.11 shows the cross validation accuracy of KNN KITML and KNN with Euclidean distance as a function of different neighbor sizes (from $k = 1$ to 11) for all 14 data sets listed in Table 4.1. The purpose of this analysis is finding the k value that leads to the highest classification accuracy for training data. Then when testing KNN KITML and KNN with Euclidean distance, this optimal value of k will be used in the classification. For example, for Alizadeh

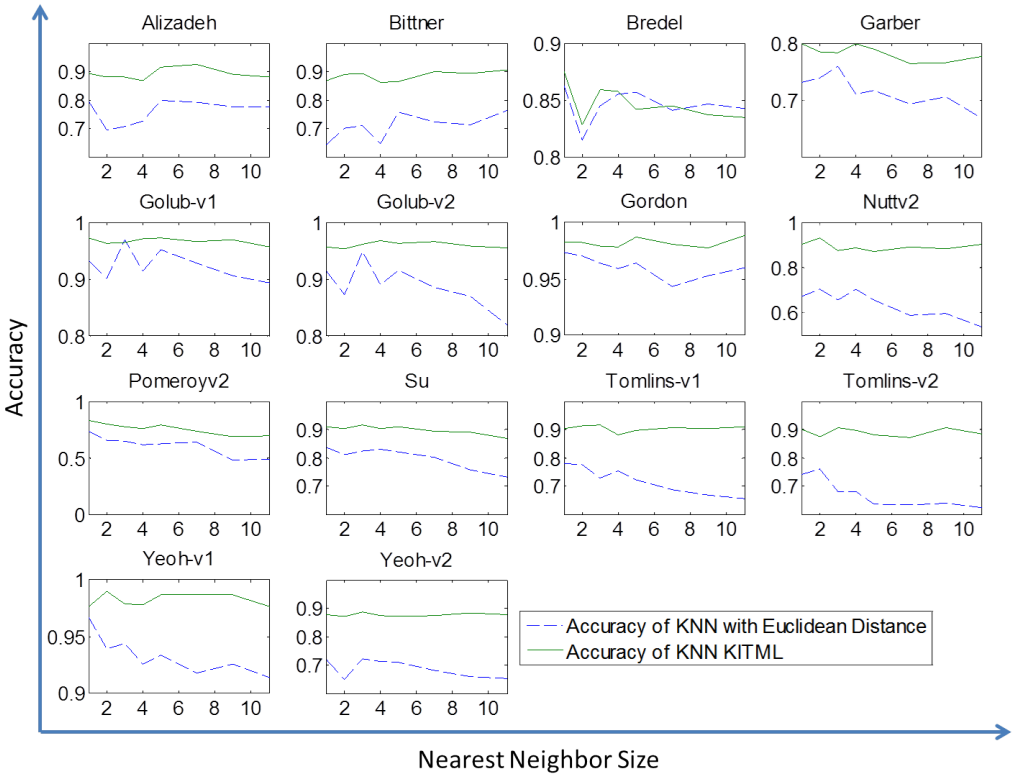


Figure 5.11: Comparing cross validation classification accuracy when varying neighbor size k

data set, when $k = 7$ KNN KITML has the highest cross validation accuracy and when $k = 5$ KNN with Euclidean distance has the highest cross validation accuracy. Moreover, from Figure 5.11, we can see that the cross validation accuracy of KNN KITML does not have an obvious trend when k is getting larger. For KNN with Euclidean Distance, it can be seen that except Alizadel, Bitterner and Bredel data sets, the cross validation accuracy decreases when k is getting larger. From all these, we can see the importance of k in the performance of KNN algorithm.

Overall Accuracy and Macro-average F1

The results of the algorithm comparison with accuracy and macro-average F1 as the performance metrics are shown in Table 5.11 and Table 5.12, respectively. For Bredel data set, the best accuracy is obtained by KNN KITML, which is 0.8760. In contrast, the Macro-averaged F1 obtained by KNN KITML is 0.7707 where there is an obvious performance decrease. This decrease can be explained by the difficulties of this classification task – the Bredel data set has unbalanced classes. It has three classes with the smallest class having a prior probability of only 10% (5 out of 50 samples). Similarity, Garber data set also has unbalance class distribution. The KNN KITML results on Garber data set are 0.8121 for accuracy and 0.6118 for macro-averaged F1. The reason is that Garber data set has an unbalanced class distribution of $\{17, 40, 4, 5\}$.

Overall, the KNN KITML has the best average performance among all 8 classification algorithms for both accuracy and Macro-averaged F1. Specifically, KNN KITML achieved best performance in accuracy in 9 out of 14 data sets (Table 5.11). LMNN has better classification in Golub-v1, Nutt, Tomlins-v2 and Yeoh-v1 data sets. For Golub-v2 data set, KNN KITML and LMNN achieved the same accuracy. KNN ITML* achieve the best accuracy for Bittner data set. For Golub-v1 data set, KNN ITML* and LMNN have the same best accuracy. KNN KITML also generated best macro-average F1 in 9 out of 14 data sets (Table 5.12). For Nutt, Tomlins-v2 and Yeoh-v1 data sets, LMNN outperformed KNN KITML in terms of Macro-averaged F1. For Garber data set, Random Forest outperformed KNN KITML in terms of Macro-averaged F1. KNN ITML* has the best Macro-averaged F1 for Bittner data set.

Table 5.11: Classification accuracies

Dataset	KNN KITML	KNN ITML*	KNN Euclidean	SVM Linear	SVM RBF	DTC	Random Forest	LMNN
Alizadeh	**0.9381±0.0319**	0.8714±0.0494	0.8048±0.0516	0.9238±0.0310	0.9238±0.0261	0.7333±0.0516	0.7048±0.0516	0.9286±0.0337
Bittner	0.9158±0.0220	**0.9368±0.0300**	0.7737±0.0634	0.8157±0.0263	0.8158±0.0671	0.5947±0.0758	0.6316±0.0758	0.8289±0.0558
Bredel	**0.8760±0.0167**	0.8503± 0.0231	0.8720±0.0179	0.8480±0.0110	0.8520±0.0110	0.3720±0.1432	0.4400±0.1432	0.8640±0.0089
Garber	**0.8121±0.0083**	0.7612± 0.0930	0.7606±0.0225	0.8090±0.0314	0.7909±0.0166	0.6727±0.0509	0.7424±0.0509	0.8106±0.0233
Golub-v1	0.9861±0.0170	**0.9861±0.0098**	0.9722±0.0000	0.9777±0.0124	0.9722±0.0098	0.8583±0.0116	0.8694±0.0116	**0.9861±0.0098**
Golub-v2	**0.9722±0.0098**	0.9694±0.0228	0.9583±0.0139	0.9277±0.0116	0.9278±0.0062	0.8111±0.0602	0.7944±0.0602	**0.9722±0.0098**
Gordon	**0.9923±0.0030**	0.9901± 0.0082	0.9834±0.0000	0.9901±0.0046	0.9901±0.0025	0.9337±0.0039	0.9370±0.0039	0.9862±0.0039
Nutt	0.9357±0.0391	0.9000±0.0299	0.7143±0.0000	0.9143±0.0196	0.8714±0.0319	0.8429±0.0542	0.8071±0.0542	**0.9643±0.0048**
Pomeroy	**0.8333±0.0292**	0.7810±0.0391	0.7524±0.0213	0.7667±0.0199	0.7810±0.0106	0.4000±0.0353	0.3476±0.0353	0.7819±0.0292
Su	**0.9195±0.0081**	0.8644 ± 0.0126	0.8414±0.0051	0.8966±0.0057	0.8931±0.0077	0.4253±0.0379	0.4448±0.0379	0.8621±0.0081
Tomlins-v1	**0.9192±0.0146**	0.8712± 0.0587	0.7904±0.0080	0.9115±0.0185	0.9000±0.0221	0.5115±0.0554	0.4827±0.0554	0.8798±0.0204
Tomlins-v2	0.9109±0.0209	0.9025 ±0.0186	0.7717±0.0133	0.9000±0.0049	0.8826±0.0179	0.6109±0.0597	0.5891±0.0597	**0.9261±0.0161**
Yeoh-v1	0.9935±0.0022	0.9895±0.0036	0.9774±0.0036	0.9839±0.0057	0.9774±0.0036	0.9927±0.0018	0.9919±0.0018	**0.9960±0.0002**
Yeoh-v2	**0.8871±0.0075**	0.8419±0.0087	0.7250±0.0157	0.8411±0.0061	0.8387±0.0128	0.4984±0.0240	0.5605±0.0240	0.8750±0.0057
Average	**0.9208**	0.8940	0.7866	0.8933	0.8869	0.6613	0.6674	0.9044

Table 5.12: Classification macro-averaged F1

Dataset	KNN KITML	KNN ITML*	KNN Euclidean	SVM Linear	SVM RBF	DTC	Random Forest	LMNN
Alizadeh	**0.9386±0.0316**	0.8762±0.0199	0.8196±0.0437	0.9240±0.0309	0.9242±0.0260	0.7357±0.0513	0.7042±0.0507	0.9296±0.0351
Bittner	0.9174±0.0219	**0.9371±0.0301**	0.7766±0.0627	0.8183±0.0253	0.8199±0.0692	0.5882±0.0686	0.6215±0.1011	0.8300±0.055
Bredel	**0.7707±0.0288**	0.7442±0.0546	0.7138±0.0352	0.7254±0.0153	0.7238±0.0168	0.4062±0.1466	0.4640±0.0706	0.7560±0.0195
Garber	0.6118±0.0156	0.6005±0.0425	0.5246±0.0301	0.6062±0.0421	0.5796±0.0411	0.5682±0.1329	**0.6819±0.0933**	0.5823±0.0386
Golub-v1	**0.9849±0.0185**	0.9637±0.0107	0.9697±0.0000	0.9758±0.0134	0.9697±0.0106	0.8478±0.0109	0.8609±0.0274	0.9848±0.0107
Golub-v2	**0.9662±0.0076**	0.9589±0.0078	0.9554±0.0229	0.9057±0.0167	0.9047±0.0135	0.8256±0.0447	0.8153±0.0247	0.9637±0.0094
Gordon	**0.9863±0.0053**	0.9823±0.0147	0.9705±0.0000	0.9824±0.0082	0.9824±0.0044	0.8850±0.0068	0.8888±0.0309	0.9755±0.007
Nutt	0.9402±0.0347	0.9086±0.0253	0.7376±0.0000	0.9207±0.0169	0.8714±0.0319	0.8309±0.0457	0.7934±0.0951	**0.9655±0.0065**
Pomeroy	**0.8451±0.0306**	0.8031±0.0556	0.7814±0.0163	0.7708±0.0287	0.7698±0.0140	0.2663±0.0390	0.2716±0.0250	0.7621±0.0285
Su	**0.9123±0.0100**	0.8351±0.0169	0.8165±0.0048	0.8733±0.0041	0.8640±0.0162	0.3995±0.0642	0.4718±0.0297	0.8326±0.0046
Tomlins-v1	**0.9205±0.0186**	0.8768±0.0220	0.8089±0.0085	0.9193±0.0192	0.9065±0.0202	0.5523±0.0317	0.5198±0.0685	0.8770±0.0286
Tomlins-v2	0.9026±0.0232	0.8801±0.0256	0.7776±0.0153	0.8994±0.0045	0.8818±0.0186	0.5693±0.0564	0.5574±0.0341	**0.9215±0.0195**
Yeoh-v1	0.9887±0.0039	0.9516±0.0064	0.9601±0.0065	0.9715±0.0102	0.9599±0.0065	0.9877±0.0030	0.9864±0.0000	**0.9930±0.0256**
Yeoh-v2	**0.8411±0.0204**	0.7798±0.0071	0.6979±0.0082	0.8049±0.0128	0.8047±0.0111	0.4814±0.0146	0.5323±0.0714	0.8264±0.0237
Average	**0.8947**	0.8656	0.8079	0.8641	0.8545	0.6389	0.6550	0.8714

Table 5.13: Comparing accuracy of KNN KITML with 7 other approaches. "Better" indicates KNN KITML achieved higher accuracy than compared approach. "Same" indicates KNN KITML achieved the same accuracy as compared approach. "Worse" indicates KNN KITML achieved lower accuracy than compared approach.

Dataset	KNN ITML*	KNN Euclidean	SVM Linear	SVM RBF	DTC	Random Forest	LMNN
Alizadeh	Better	Better	Better	Better	Better	Better	Better
Bittner	Worse	Better	Better	Better	Better	Better	Better
Bredel	Better	Better	Better	Better	Better	Better	Better
Garber	Better	Better	Better	Better	Better	Better	Better
Golub-v1	Worse	Better	Better	Better	Better	Better	Worse
Golub-v2	Better	Better	Better	Better	Better	Better	Same
Gordon	Better	Better	Better	Better	Better	Better	Better
Nutt	Better	Better	Better	Better	Better	Better	Worse
Pomeroy	Better	Better	Better	Better	Better	Better	Better
Su	Better	Better	Better	Better	Better	Better	Better
Tomlins-v1	Better	Better	Better	Better	Better	Better	Better
Tomlins-v2	Better	Better	Better	Better	Better	Better	Worse
Yeoh-v1	Better	Better	Better	Better	Better	Better	Worse
Yeoh-v2	Better	Better	Better	Better	Better	Better	Better

Table 5.14: Comparing macro-average F1 of KNN KITML with 7 other approaches. "Better" indicates KNN KITML achieved higher macro-average F1 than compared approach. "Worse" indicates KNN KITML achieved lower macro-average F1 than compared approach.

Dataset	KNN ITML*	KNN Euclidean	SVM Linear	SVM RBF	DTC	Random Forest	LMNN
Alizadeh	Better	Better	Better	Better	Better	Better	Better
Bittner	Worse	Better	Better	Better	Better	Better	Better
Bredel	Better	Better	Better	Better	Better	Worse	Better
Garber	Better	Better	Better	Better	Better	Better	Better
Golub-v1	Better	Better	Better	Better	Better	Better	Better
Golub-v2	Better	Better	Better	Better	Better	Better	Better
Gordon	Better	Better	Better	Better	Better	Better	Better
Nutt	Better	Better	Better	Better	Better	Better	Worse
Pomeroy	Better	Better	Better	Better	Better	Better	Better
Su	Better	Better	Better	Better	Better	Better	Better
Tomlins-v1	Better	Better	Better	Better	Better	Better	Better
Tomlins-v2	Better	Better	Better	Better	Better	Better	Worse
Yeoh-v1	Better	Better	Better	Better	Better	Better	Worse
Yeoh-v2	Better	Better	Better	Better	Better	Better	Better

Wilcoxon Signed-Ranks Test Results

Wilcoxon signed-ranks test is used to verify that the differences in accuracy between algorithms are non-random. The Wilcoxon signed-ranks test ranks the difference in performance of two classifiers for each data set, ignoring the signs and compares the ranks for positive and negative differences. The results of right-sided Wilcoxon signed-ranks test are shown in Table 5.15. The p-values of the tests between KITML and the other 6 classification algorithms indicate that KNN KITML achieved better performance than all the other algorithms in terms of accuracy and macro-average F1 at 5% significance level.

Table 5.15: p-values of right-sided Wilcoxon signed-ranks test between KNN KITML and the other 7 classification algorithms

	KNN ITML*	**KNN Euclidean**	**SVM Linear**	**SVM RBF**	**DTC**	**Random Forest**	**LMNN**
p-values for accuracies	0.0012	6.1e-05	6.1e-05	6.1e-05	6.1e-05	6.1e-05	0.03
p-values for macro-averaged F1	6.1e-04	6.1e-05	6.1e-05	6.1e-05	6.1e-05	1.8e-04	0.02

Time Complexity Analysis

The objective of the learning process in KITML is to learn the n by n parameter matrix in the distance metric, where n is the number of samples. Therefore, for each constraint (l or u) defined in equation 4.5, the time complexity is $O(n^2)$. For the entire learning process looping through all the constraints, the time complexity is $O(cn^2)$, where c is the number of constraints defined in Chapter 4.3.4. We further analyzed the execution times for the two best classification algorithms – KNN KITML and LMNN. Our experimental setup was designed to obtain reliable performance estimates and

avoid over-fitting using two loops. The inner loop is used to determine the best parameters of the classifier using cross-validation sets. The outer loop is used to estimate the performance of the classifiers built using the parameters found by the inner loop. In the execution time analysis, we only ran outer loop for each algorithms. For the inner loop we usde default parameters for each algorithms. Figure 5.12 shows the execution time analysis. KNN KITML requires much less time, taking 2-519 seconds to run each data set, while for LMNN this typically exceeded 24 hours to finish calculation. All experiments were executed in Matlab® 2012a software (The Mathworks, Natick, MA) on a Quad core Intel 3.5GHz PC.

5.3.2 Estimating Severity of Sample Subgroups

The estimated severity levels as determined by KITML on the three microarray datasets from bladder, prostate and ovarian multi-stage cancer patient studies (Table 4.2) are shown in Figure 5.13. Each data set contains samples from 3 different cancer stages, indicating 3 different severity levels. Each y_i is the mean of sample severity levels within $\mathbf{E}_i^?$. The standard deviation is shown as the error bar in the figure. We consider the relative order of $y_1 < y_2 < y_3$ as the correct severity estimation since it matches the true group labels and our KITML approach correctly estimated the relative ordering among the severity levels of the three test groups. Notably, without any prior information about $\mathbf{E}_2^?$ in each data set, the proposed approach still can estimate the severity level y_2 of $\mathbf{E}_2^?$ in the right order – between y_1 and y_3.

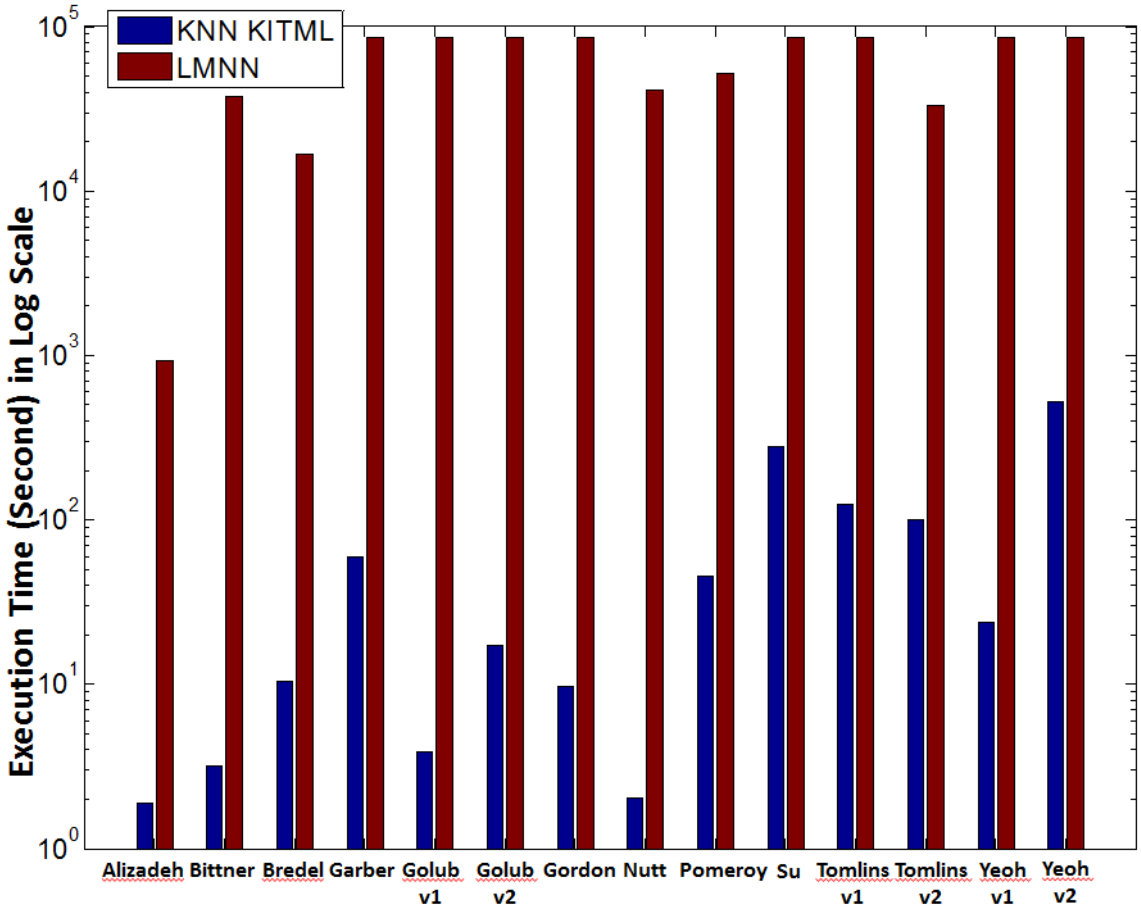


Figure 5.12: Comparing execution time between KNN KITML and LMNN for all 14 data sets. Since Garber, Golub-v1, Golub-v2, Gordon, Su, Tomlins-v1, Yeoh-v1 and Yeoh-v2 need more than 24 hours execution time, we draw their bars using the same longest length in the figure.

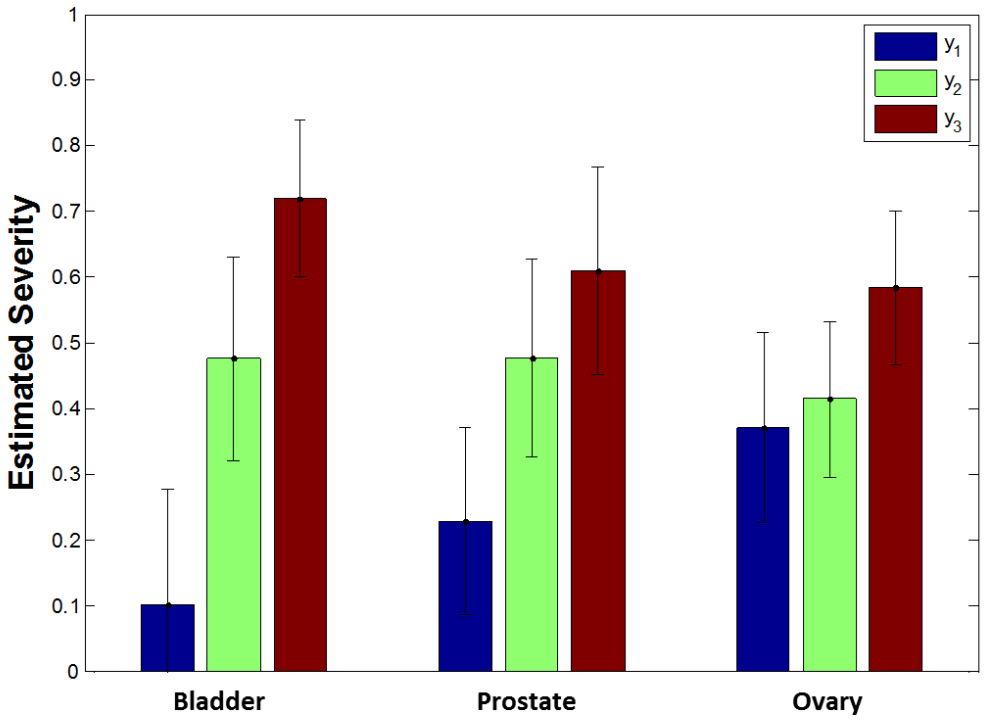


Figure 5.13: Severity estimation results of three high-dimensional data sets. The blue bar is the severity level y_1 of test group $\mathbf{E}_1^?$, the green bar is the severity levels y_2 of test group $\mathbf{E}_2^?$, and the red bar is the severity levels y_3 of test group $\mathbf{E}_3^?$.

6. Conclusions and Future Work

This thesis describes distance measures used to address two kinds of problems: severity estimation and cancer tissue classification. For severity estimation, we first applied several binary severity estimation approaches to Cytokine Release Syndrome (CRS) data. These approaches were (i) Hierarchical Cluster Analysis (HCA); (ii) Principal Component Analysis (PCA) followed by K-means clustering; and (iii) Decision Tree Classification (DTC). All three approaches were able to identify the treatment that caused the most severe cytokine response. HCA was able to provide information about the expected number of clusters in the data. PCA coupled with K-means clustering allowed classification of treatments sample by sample, and visualizing clusters of treatments. DTC models showed the relative importance of various cytokines such as IFN-γ, TNF-α and IL-10 to CRS. The use of these approaches in tandem provided better selection of parameters for one method based on outcomes from another, and an overall improved analysis of the data through complementary approaches. Moreover, the DTC analysis showed in addition that IL-17 may be correlated with CRS reactions. This correlation has not yet been corroborated in the literature.

Next we went beyond binary severity estimation using distance metric learning algorithms which allowed us to determine a more graded severity level for different bioinformatics areas. We use the known severity of both negative controls (least severe) and positive controls (most severe) to define the range of possible severity, and used this information to learn a distance metric from data. This learned metric is used to measure the distances of an unknown disease or reaction from both the negative controls and positive controls and thus to estimate its severity. We evaluated four known data sets which studied the severity of CRS, the severity of fetal hypoxia, and

toxic reactions of chemical compounds. We compared our approach to four public methods from the literature. The results showed that our approach was able to estimate correctly the severity of the disease/reaction better than the other approaches. Regression based approaches and approaches that use other distance metrics were less stable in estimating the corrected results. In the future, we would like to generalize our severity estimation approaches to more data sets and test their ability to estimate the severity in different bioinformatics areas. Currently, the learning distance metric is based on positive and negative controls, however, it is possible that some samples of middle severity are also known beforehand, we would like to incorporate these known samples to improve our learned metric. How to revise the current severity estimation framework to adapt this scenario is left for future work.

The second problem we addressed in this thesis is cancer tissue classification. We used a Kernelized Information-Theoretic Metric Learning (KITML) approach that optimizes a distance function to improve the classification of cancer microarray data and scale to high dimensionality. By learning a nonlinear transformation in the input space implicitly through kernelization, KITML permits efficient optimization, low storage, and improved learning of distance metric. We proposed two applications of KITML using high-dimensional microarray data. (1) For sample-level tissue classification, the learned metric is used to improve the performance of k-nearest neighbor classification. (2) For estimating the severity level or stage of a group of samples, we propose a set-based scheme to identify the stage/severity of different cancer. For the sample-level cancer classification task, we evaluated fourteen cancer gene microarray data sets and compared with six other state-of-the-art approaches. The results show that our approach achieves the best overall performance for the task of molecular expression driven cancer tissue classification. For the group-level cancer stage estimation, we test the proposed set-KITML approach using three multi-stage cancer

microarray data sets, and correctly estimated the stages of sample groups for all three studies.

Currently we use the learned distance metric to estimate severity levels of different biomedical conditions and to classify cancer tissue microarray data. However, the use of learned distance metrics may not be limited to just these applications, e.g., in cancer diagnosis, the learned metric may be used to find similar patient cases to a target patient case using a nearest neighbor search. This may provide a great tool for physicians when determining relevant prognoses or treatment plans. In addition, there are many other possible distance metric learning algorithms, but we only compared some of them. Future work may include a comprehensive comparison with more distance metric learning algorithms to determine the best approach for a given bioinformatics scenario.

Bibliography

[1] R. D. KING, J. D. HIRST, and M. J. E. STERNBERG, "Comparison of artificial intelligence methods for modeling pharmaceutical qsars," *Applied Artificial Intelligence*, vol. 9, no. 2, pp. 213–233, 1995.

[2] C. Nantasenamat, C. Isarankura-Na-Ayudhya, T. Naenna, and V. Prachayasittikul, "A practical overview of quantitative structure-activity relationship," *EXCLI J.*, vol. 8, pp. 74–88, 2009.

[3] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression," *Journal of Machine Learning Research*, vol. 6, p. 2005, 2004.

[4] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007.

[5] M. O'Searcoid, *Elements of Abstract Analysis*. Springer, 2001.

[6] K. Buza, A. Nanopoulos, and L. Schmidt-Thieme, "Fusion of similarity measures for time series classification," in *Hybrid Artificial Intelligent Systems* (E. Corchado, M. Kurzyński, and M. Woźniak, eds.), vol. 6679 of *Lecture Notes in Computer Science*, pp. 253–261, Springer Berlin Heidelberg, 2011.

[7] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *Proc. VLDB Endow.*, vol. 1, pp. 1542–1552, Aug. 2008.

[8] E. Keogh, C. Shelton, and F. Moerchen, "First international workshop and challenge on time series classification," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, (New York, NY, USA), pp. 11:1–11:1, ACM, 2007.

[9] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. II–521–II–527 vol.2, June 2003.

[10] E. Xing, A. Ng, M. Jordan, and S. Russell, *Distance Metric Learning with Application to Clustering with Side-Information*, pp. 505–512. MIT Press, 2003.

[11] C. Ramaker, J. Marinus, A. M. Stiggelbout, and B. J. van Hilten, "Systematic evaluation of rating scales for impairment and disability in parkinson's disease," *Movement Disorders*, vol. 17, no. 5, pp. 867–876, 2002.

[12] M. D. Birkner, S. Kalantri, V. Solao, P. Badam, R. Joshi, A. Goel, M. Pai, and A. E. Hubbard, "Creating diagnostic scores using data-adaptive regression: An application to prediction of 30-day mortality among stroke victims in a rural hospital in india," *Therapeutics and clinical risk management*, vol. 3, no. 3, pp. 475–484, 2007.

[13] A. Tsanas, M. Little, P. McSharry, and L. Ramig, "Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 4, pp. 884–893, 2010.

[14] F. Pedregosa, E. Cauvet, G. Varoquaux, C. Pallier, B. Thirion, and A. Gramfort, "Learning to rank from medical imaging data," in *Machine Learning in Medical Imaging* (F. Wang, D. Shen, P. Yan, and K. Suzuki, eds.), vol. 7588 of *Lecture Notes in Computer Science*, pp. 234–241, Springer Berlin Heidelberg, 2012.

[15] S. Ramaswamy, K. Ross, E. Lander, and T. Golub, "A molecular signature of metastasis in primary solid tumors," *Nature genetics*, vol. 33, no. 1, pp. 49–54, 2002.

[16] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis," *Bioinformatics*, vol. 21, no. 5, pp. 631–643, 2005.

[17] W. J. Pichler, "Adverse side-effects to biological agents," *Allergy*, vol. 61, no. 8, pp. 912–920, 2006.

[18] M. R. Walker, D. A. Makropoulos, R. Achuthanandam, S. V. Arsdell, and P. J. Bugelski, "Development of a human whole blood assay for prediction of cytokine release similar to anti-cd28 superagonists using multiplex cytokine and hierarchical cluster analysis," *International Immunopharmacology*, vol. 11, no. 11, pp. 1697 – 1705, 2011.

[19] *COMMON TOXICITY CRITERIA (CTC), version 2.0*, 1999.

[20] G. P. Sandilands, M. Wilson, C. Huser, L. Jolly, W. A. Sands, and C. McSharry, "Were monocytes responsible for initiating the cytokine storm in the tgn1412 clinical trial tragedy?," *Clinical & Experimental Immunology*, vol. 162, no. 3, pp. 516–527, 2010.

[21] R. Stebbings, L. Findlay, C. Edwards, D. Eastwood, C. Bird, D. North, Y. Mistry, P. Dilger, E. Liefooghe, I. Cludts, B. Fox, G. Tarrant, J. Robinson, T. Meager, C. Dolman, S. J. Thorpe, A. Bristow, M. Wadhwa, R. Thorpe, and S. Poole,

""cytokine storm" in the phase i trial of monoclonal antibody tgn1412: Better understanding the causes to improve preclinical testing of immunotherapeutics," *The Journal of Immunology*, vol. 179, no. 5, pp. 3325–3331, 2007.

[22] D. H. Hsu, J. D. Shi, M. Homola, T. J. Rowell, J. Moran, D. Levitt, B. Druilhet, J. Chinn, C. Bullock, and C. Klingbeil, "A humanized anti-cd3 antibody, hum291, with low mitogenic activity, mediates complete and reversible t-cell depletion in chimpanzees," *Transplantation*, vol. 68, no. 4, pp. 545–54, 1999.

[23] E. A. Hod, C. M. Cadwell, J. S. Liepkalns, J. C. Zimring, S. A. Sokol, D. A. Schirmer, J. Jhang, and S. L. Spitalnik, "Cytokine storm in a mouse model of igg-mediated hemolytic transfusion reactions," *Blood*, vol. 112, no. 3, pp. 891–4, 2008.

[24] C. FERRAN, M. DY, K. SHEEHAN, S. MERITE, R. SCHREIBER, P. LANDAIS, G. GRAU, J. BLUESTONE, J.-F. BACH, and L. CHATENOUD, "Inter-mouse strain differences in the in vivo anti-cd3 induced cytokine release," *Clinical & Experimental Immunology*, vol. 86, no. 3, pp. 537–543, 1991.

[25] A. A. Ansari, A. Mayne, D. Hunt, J. B. Sundstrom, and F. Villinger, "Th1/th2 subset analysis. i. establishment of criteria for subset identification in pbmc samples from nonhuman primates," *Journal of Medical Primatology*, vol. 23, no. 2-3, pp. 102–107, 1994.

[26] D. Eastwood, L. Findlay, S. Poole, C. Bird, M. Wadhwa, M. Moore, C. Burns, R. Thorpe, and R. Stebbings, "Monoclonal antibody tgn1412 trial failure explained by species differences in cd28 expression on cd4+ effector memory t-cells," *British Journal of Pharmacology*, vol. 161, no. 3, pp. 512–526, 2010.

[27] D. H. Nguyen, N. Hurtado-Ziola, P. Gagneux, and A. Varki, "Loss of siglec expression on t lymphocytes during human evolution," *Proceedings of the National Academy of Sciences*, vol. 103, no. 20, pp. 7765–7770, 2006.

[28] R. J. Ober, C. G. Radu, V. Ghetie, and E. S. Ward, "Differences in promiscuity for antibody–fcrn interactions across species: implications for therapeutic antibodies," *International Immunology*, vol. 13, no. 12, pp. 1551–1559, 2001.

[29] J. A. Patel, S. Nair, J. Grady, K. Revai, S. Victor, A. R. Brasier, and T. Chonmaitree, "Systemic cytokine response profiles associated with respiratory virus-induced acute otitis media," *Pediatr Infect Dis J*, vol. 28, no. 5, pp. 407–411 10.1097/INF.0b013e318194b7c6, 2009.

[30] P. Szodoray, P. Alex, V. Dandapani, B. Nakken, J. Pesina, X. Kim, G. L. Wallis, P. C. Wilson, R. Jonsson, and M. Centola, "Apoptotic effect of rituximab on peripheral blood b cells in rheumatoid arthritis," *Scandinavian Journal of Immunology*, vol. 60, no. 1-2, pp. 209–218, 2004.

[31] H. J. van den Ham, W. de Jager, J. W. Bijlsma, B. J. Prakken, and R. J. de Boer, "Differential cytokine profiles in juvenile idiopathic arthritis subtypes revealed by cluster analysis," *Rheumatology*, vol. 48, no. 8, pp. 899–905, 2009. Using Smart Source Parsing Aug; doi: 10.1093/rheumatology/kep125. Epub 2009 May 28.

[32] Z. Alfirevic, D. Devane, and G. M. Gyte, "Continuous cardiotocography (ctg) as a form of electronic fetal monitoring (efm) for fetal assessment during labour," *Cochrane Database Syst Rev*, no. 3, p. Cd006066, 2006.

[33] A. Costa, D. Ayres-de Campos, F. Costa, C. Santos, and J. Bernardes, "Prediction of neonatal acidemia by computer analysis of fetal heart rate and st event signals," *Am J Obstet Gynecol*, vol. 201, no. 5, pp. 464.e1–6, 2009. 1097-6868 Costa, Antonia Ayres-de-Campos, Diogo Costa, Fernanda Santos, Cristina Bernardes, Joao Comparative Study Journal Article Research Support, Non-U.S. Gov't United States Am J Obstet Gynecol. 2009 Nov;201(5):464.e1-6. doi: 10.1016/j.ajog.2009.04.033. Epub 2009 Jun 18.

[34] M.-L. Huang and Y.-Y. Hsu, "Fetal distress prediction using discriminant analysis, decision tree, and artificial neural network," *Journal of Biomedical Science and Engineering*, vol. 5, no. 9, pp. 526–533, 2012.

[35] M. Romano, M. Bracale, M. Cesarelli, M. Campanile, P. Bifulco, M. De Falco, M. Sansone, and A. Di Lieto, "Antepartum cardiotocography: A study of fetal reactivity in frequency domain," *Comput. Biol. Med.*, vol. 36, pp. 619–633, June 2006.

[36] S. Nidhal, M. A. M. Ali1, and H. Najah, "A novel cardiotocography fetal heart rate baseline estimation algorithm," *Scientific Research and Essays*, vol. 5, no. 24, pp. 1002–4010, 18 December, 2010.

[37] K. Bache and M. Lichman, "UCI machine learning repository," 2013.

[38] J. D. McKinney, A. Richard, C. Waller, M. C. Newman, and F. Gerberick, "The practice of structure activity relationships (sar) in toxicology," *Toxicological Sciences*, vol. 56, no. 1, pp. 8–17, 2000.

[39] P. Chauhan and M. Shakya, "Role of physicochemical properties in the estimation of skin permeability: in vitro data assessment by partial least-squares regression," *SAR QSAR Environ Res*, vol. 21, no. 5-6, pp. 481–94, 2010.

[40] X. Ning and G. Karypis, "In silico structure-activity-relationship (sar) models from machine learning: a review," *Drug Development Research*, vol. 72, 2010. impact factor: 1.109.

[41] J. R. Bock and D. A. Gough, "A new method to estimate ligand-receptor energetics," *Mol Cell Proteomics*, vol. 1, no. 11, pp. 904–10, 2002.

[42] L. Jacob and J.-P. Vert, "Protein-ligand interaction prediction: an improved chemogenomics approach," *Bioinformatics*, vol. 24, no. 19, pp. 2149–2156, 2008.

[43] D. Erhan, P.-J. L'Heureux, S. Y. Yue, and Y. Bengio, "Collaborative filtering on a family of biological targets," *Journal of Chemical Information and Modeling*, vol. 46, no. 2, pp. 626–635, 2006. PMID: 16562992.

[44] H. Strombergsson, P. Daniluk, A. Kryshtafovych, K. Fidelis, J. E. Wikberg, G. J. Kleywegt, and T. R. Hvidsten, "Interaction model based on local protein substructures generalizes to the entire structural enzyme-ligand space," *J Chem Inf Model*, vol. 48, no. 11, pp. 2278–88, 2008.

[45] Cancer Genome Atlas Research Network *et al.*, "Comprehensive genomic characterization of squamous cell lung cancers," *Nature*, vol. 489, no. 7417, pp. 519–525, 2012.

[46] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, corrected ed., Aug. 2003.

[47] G. C. Cawley and N. L. Talbot, "Gene selection in cancer classification using sparse logistic regression with bayesian regularization," *Bioinformatics*, vol. 22, no. 19, pp. 2348–2355, 2006.

[48] T. J. Hudson, W. Anderson, A. Aretz, A. D. Barker, C. Bell, R. R. Bernabé, M. Bhan, F. Calvo, I. Eerola, D. S. Gerhard, *et al.*, "International network of cancer genome projects," *Nature*, vol. 464, no. 7291, pp. 993–998, 2010.

[49] J. Davis, B. Kulis, S. Sra, and I. Dhillon, "Information-theoretic metric learning," in *in NIPS 2006 Workshop on Learning to Compare Examples*, 2007.

[50] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, *Information Theoretic Metric Learning*. UT, Austin, http://www.cs.utexas.edu/users/pjain/itml/.

[51] B. Kulis, "Metric learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2013.

[52] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009.

[53] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 1–26, 2012.

[54] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[55] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[56] P. J. Bugelski, R. Achuthanandam, R. J. Capocasale, G. Treacy, and E. Bouman-Thio, "Monoclonal antibody-induced cytokine-release syndrome," *Expert Rev Clin Immunol*, vol. 5, no. 5, pp. 499–521, 2009.

[57] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[58] F. Perveen and Z. Hussain, "Use of statistical techniques in analysis of biological data," *Basic Research Journal of Agricultural Science and Review*, vol. 1, no. 1, pp. 01–10, 2012.

[59] J. M. Bland and D. G. Altman, "Transforming data," *Bmj*, vol. 312, no. 7033, p. 770, 1996. Bland, J M Altman, D G Journal Article England BMJ. 1996 Mar 23;312(7033):770.

[60] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armananzas, G. Santafe, A. Perez, and V. Robles, "Machine learning in bioinformatics," *Brief Bioinform*, vol. 7, no. 1, pp. 86–112, 2006. Larranaga, Pedro Calvo, Borja Santana, Roberto Bielza, Concha Galdiano, Josu Inza, Inaki Lozano, Jose A Armananzas, Ruben Santafe, Guzman Perez, Aritz Robles, Victor Research Support, Non-U.S. Gov't Review England Brief Bioinform. 2006 Mar;7(1):86-112.

[61] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009.

[62] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd ed., 2011.

[63] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *MACHINE LEARNING: PROCEEDINGS OF THE ELEVENTH INTERNATIONAL*, pp. 121–129, Morgan Kaufmann, 1994.

[64] B. Bannwarth and C. Richez, "Clinical safety of tocilizumab in rheumatoid arthritis," *Expert Opin Drug Saf*, vol. 10, no. 1, pp. 123–31, 2011.

[65] X. Sez-Llorens, M. T. Moreno, O. Ramilo, P. J. Snchez, F. H. J. Top, E. M. Connor, and f. t. M.-. S. Group, "Safety and pharmacokinetics of palivizumab therapy in children hospitalized with respiratory syncytial virus infection," *The Pediatric Infectious Disease Journal*, vol. 23, no. 8, pp. 707–712, 2004.

[66] M. A. Fanale and A. Younes, "Monoclonal antibodies in the treatment of non-hodgkin's lymphoma," *Drugs*, vol. 67, no. 3, pp. 333–50, 2007. Fanale, Michelle A Younes, Anas Journal Article Review New Zealand Drugs. 2007;67(3):333-50.

[67] E. Dhimolea, "Canakinumab," *MAbs*, vol. 2, no. 1, pp. 3–13, 2010. 1942-0870 Dhimolea, Eugen Journal Article Review United States MAbs. 2010 Jan-Feb;2(1):3-13. Epub 2010 Jan 15.

[68] F. E. Roufosse, J. E. Kahn, G. J. Gleich, L. B. Schwartz, A. D. Singh, L. J. Rosenwasser, J. A. Denburg, J. Ring, M. E. Rothenberg, J. Sheikh, A. E. Haig, S. A. Mallett, D. N. Templeton, H. G. Ortega, and A. D. Klion, "Long-term safety of mepolizumab for the treatment of hypereosinophilic syndromes," *J Allergy Clin Immunol*, vol. 131, no. 2, pp. 461–7.e1–5, 2013.

[69] L. Yang, "Distance metric learning: A comprehensive survey," 2006.

[70] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. pp. 79–86, 1951.

[71] L. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," {*USSR*} *Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200 – 217, 1967.

[72] M. Walker, D. Makropoulos, R. Achuthanandam, and P. J. Bugelski, "Recent advances in the understanding of drug-mediated infusion reactions and cytokine release syndrome," *Curr Opin Drug Discov Devel*, vol. 13, no. 1, pp. 124–35, 2010.

[73] D. Ayres-de Campos, J. Bernardes, A. Garrido, J. Marques-de Sá, and L. Pereira-Leite, "Sisporto 2.0: A program for automated analysis of cardiotocograms," *The Journal of Maternal-Fetal Medicine*, vol. 9, no. 5, pp. 311–318, 2000.

[74] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[75] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[76] M. de Souto, I. Costa, D. de Araujo, T. Ludermir, and A. Schliep, "Clustering cancer gene expression data: a comparative study," *BMC Bioinformatics*, vol. 9, no. 1, p. 497, 2008.

[77] M. C. P. de Souto, R. B. C. Prudencio, R. G. F. Soares, D. A. S. Araujo, I. G. Costa, T. B. Ludermir, and A. Schliep, "Ranking and selecting clustering algorithms using a meta-learning approach," 2008.

[78] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E.

Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000. 10.1038/35000501.

[79] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, N. Hayward, and J. Trent, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000. 10.1038/35020115.

[80] M. Bredel, C. Bredel, D. Juric, G. R. Harsh, H. Vogel, L. D. Recht, and B. I. Sikic, "Functional network analysis reveals extended gliomagenesis pathway maps and three novel myc-interacting genes in human gliomas," *Cancer Res*, vol. 65, no. 19, pp. 8679–89, 2005. Bredel, Markus Bredel, Claudia Juric, Dejan Harsh, Griffith R Vogel, Hannes Recht, Lawrence D Sikic, Branimir I CA92474/CA/NCI NIH HHS/United States Research Support, N.I.H., Extramural Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, P.H.S. United States Cancer Res. 2005 Oct 1;65(19):8679-89.

[81] M. E. Garber, O. G. Troyanskaya, K. Schluens, S. Petersen, Z. Thaesler, M. Pacyna-Gengelbach, M. van de Rijn, G. D. Rosen, C. M. Perou, R. I. Whyte, R. B. Altman, P. O. Brown, D. Botstein, and I. Petersen, "Diversity of gene expression in adenocarcinoma of the lung," *Proceedings of the National Academy of Sciences*, vol. 98, no. 24, pp. 13784–13789, 2001.

[82] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–7, 1999. Golub, T R Slonim, D K Tamayo, P Huard, C Gaasenbeek, M Mesirov, J P Coller, H Loh, M L Downing, J R Caligiuri, M A Bloomfield, C D Lander, E S Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, P.H.S. United states Science. 1999 Oct 15;286(5439):531-7.

[83] G. J. Gordon, R. V. Jensen, L.-L. Hsiao, S. R. Gullans, J. E. Blumenstock, S. Ramaswamy, W. G. Richards, D. J. Sugarbaker, and R. Bueno, "Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma," *Cancer Research*, vol. 62, no. 17, pp. 4963–4967, 2002.

[84] C. L. Nutt, D. R. Mani, R. A. Betensky, P. Tamayo, J. G. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. E. McLaughlin, T. T. Batchelor, P. M. Black, A. von Deimling, S. L. Pomeroy, T. R. Golub, and D. N. Louis, "Gene expression-based classification of malignant gliomas correlates better with survival than histological classification," *Cancer Research*, vol. 63, no. 7, pp. 1602–1607, 2003.

[85] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub, "Prediction of central nervous system embryonal tumour outcome based on gene expression," *Nature*, vol. 415, no. 6870, pp. 436–442, 2002. 10.1038/415436a.

[86] A. I. Su, J. B. Welsh, L. M. Sapinoso, S. G. Kern, P. Dimitrov, H. Lapp, P. G. Schultz, S. M. Powell, C. A. Moskaluk, H. F. Frierson, and G. M. Hampton, "Molecular classification of human carcinomas by use of gene expression signatures," *Cancer Research*, vol. 61, no. 20, pp. 7388–7393, 2001.

[87] S. A. Tomlins, R. Mehra, D. R. Rhodes, X. Cao, L. Wang, S. M. Dhanasekaran, S. Kalyana-Sundaram, J. T. Wei, M. A. Rubin, K. J. Pienta, R. B. Shah, and A. M. Chinnaiyan, "Integrative molecular concept modeling of prostate cancer progression," *Nat Genet*, vol. 39, no. 1, pp. 41–51, 2007. 10.1038/ng1935.

[88] E.-J. Yeoh, M. E. Ross, S. A. Shurtleff, W. K. Williams, D. Patel, R. Mahfouz, F. G. Behm, S. C. Raimondi, M. V. Relling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C.-H. Pui, W. E. Evans, C. Naeve, L. Wong, and J. R. Downing, "Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling," *Cancer Cell*, vol. 1, no. 2, pp. 133–143, 2002.

[89] R. Jizba, "Measuring search effectiveness," 2007.

[90] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.

[91] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, pp. 80–83, Dec. 1945.

[92] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, *et al.*, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.

[93] C.-K. Chen, "The classification of cancer stage microarray data," *Computer Methods and Programs in Biomedicine*, vol. 108, no. 3, pp. 1070 – 1077, 2012.

[94] R. Wu, N. Hendrix-Lucas, others, E. R. Fearon, and K. R. Cho, "Mouse model of human ovarian endometrioid adenocarcinoma based on somatic defects in the wnt/-catenin and pi3k/pten signaling pathways," *Cancer Cell*, vol. 11, no. 4, pp. 321 – 333, 2007.

[95] L. Dyrskjot, T. Thykjaer, M. Kruhoffer, J. L. Jensen, N. Marcussen, S. Hamilton-Dutoit, H. Wolf, and T. F. Orntoft, "Identifying distinct classes of bladder carcinoma using microarrays," *Nat Genet*, vol. 33, no. 1, pp. 90–96, 2003. 10.1038/ng1061.

[96] L. True, I. Coleman, others, and P. S. Nelson, "A molecular correlate to the gleason grading system for prostate adenocarcinoma," *Proceedings of the National Academy of Sciences*, vol. 103, no. 29, pp. 10991–10996, 2006.

[97] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue classification with gene expression profiles," *J Comput Biol*, vol. 7, no. 3-4, pp. 559–83, 2000.

[98] B. Qin, Y. Xia, and F. Li, *DTU: A Decision Tree for Uncertain Data*, vol. 5476 of *Lecture Notes in Computer Science*, book section 4, pp. 4–15. Springer Berlin Heidelberg, 2009.

[99] B. Chromy, I. Fodor, N. Montgomery, P. Luciw, and S. McCutchen-Maloney, "Cluster analysis of host cytokine responses to biodefense pathogens in a whole blood ex vivo exposure model (weem)," *BMC Microbiology*, vol. 12, no. 1, p. 79, 2012.

[100] V. Lvovschi, L. Arnaud, C. Parizot, Y. Freund, G. Juillien, P. Ghillani-Dalbin, M. Bouberima, M. Larsen, B. Riou, G. Gorochov, and P. Hausfater, "Cytokine profiles in sepsis have limited relevance for stratifying patients in the emergency department: A prospective observational study," *PLoS ONE*, vol. 6, p. e28870, 12 2011.

[101] A. Helmy, C. A. Antoniades, M. R. Guilfoyle, K. L. H. Carpenter, and P. J. Hutchinson, "Principal component analysis of the cytokine and chemokine response to human traumatic brain injury," *PLoS ONE*, vol. 7, p. e39677, 06 2012.

[102] H.-L. Wong, R. M. Pfeiffer, T. R. Fears, R. Vermeulen, S. Ji, and C. S. Rabkin, "Reproducibility and correlations of multiplex cytokine levels in asymptomatic persons," *Cancer Epidemiology Biomarkers & Prevention*, vol. 17, no. 12, pp. 3450–3456, 2008.

[103] D. Dhananjay, D. M. Richard, H. Pranab, S. Sachil, G. Sumit, M. Bafadhel, W. Jo, H. Athula, K. A. Ian, G. Ruth, W. Andrew, B. Peter, D. P. Ian, and C. E. Brightling, *Cytokine Profiling In Severe Asthma Subphenotypes Using*

Factor And Cluster Analysis, pp. A3719–A3719. American Thoracic Society International Conference Abstracts, American Thoracic Society, 2011.

[104] Y. Kumar, C. Liang, Z. Bo, J. C. Rajapakse, E. E. Ooi, and S. R. Tannenbaum, "Serum proteome and cytokine analysis in a longitudinal cohort of adults with primary dengue infection reveals predictive markers of dhf," *PLoS Negl Trop Dis*, vol. 6, no. 11, p. e1887, 2012.

[105] B. A. McKinney, D. M. Reif, M. T. Rock, K. M. Edwards, S. F. Kingsmore, J. H. Moore, and J. Crowe, J. E., "Cytokine expression patterns associated with systemic adverse events following smallpox immunization," *J Infect Dis*, vol. 194, no. 4, pp. 444–53, 2006.

[106] G. Suntharalingam, M. Perry, S. Ward, S. Brett, A. Castello-Cortes, M. Brunner, and N. Panoskaltsis, "Cytokine storm in a phase 1 trial of the anti-cd28 monoclonal antibody tgn1412," *New England Journal of Medicine*, vol. 355, no. 10, pp. 1018–1028, 2006.

[107] H. H. Yiu, A. L. Graham, and R. F. Stengel, "Dynamics of a cytokine storm," *PLoS One*, vol. 7, no. 10, p. e45027, 2012.

[108] H. Park, Z. Li, X. O. Yang, S. H. Chang, R. Nurieva, Y. H. Wang, Y. Wang, L. Hood, Z. Zhu, Q. Tian, and C. Dong, "A distinct lineage of cd4 t cells regulates tissue inflammation by producing interleukin 17," *Nat Immunol*, vol. 6, no. 11, pp. 1133–41, 2005.

[109] F. Shen, M. J. Ruddy, P. Plamondon, and S. L. Gaffen, "Cytokines link osteoblasts and inflammation: microarray analysis of interleukin-17- and tnf-alpha-induced genes in bone cells," *J Leukoc Biol*, vol. 77, no. 3, pp. 388–99, 2005.

[110] S. L. Gaffen, "Structure and signalling in the il-17 receptor family," *Nat Rev Immunol*, vol. 9, no. 8, pp. 556–67, 2009.

[111] F. Fossiez, O. Djossou, P. Chomarat, L. Flores-Romo, S. Ait-Yahia, C. Maat, J. J. Pin, P. Garrone, E. Garcia, S. Saeland, D. Blanchard, C. Gaillard, B. Das Mahapatra, E. Rouvier, P. Golstein, J. Banchereau, and S. Lebecque, "T cell interleukin-17 induces stromal cells to produce proinflammatory and hematopoietic cytokines," *J Exp Med*, vol. 183, no. 6, pp. 2593–603, 1996.

[112] M. Chabaud, F. , J. L. Taupin, and P. Miossec, "Enhancing effect of il-17 on il-1-induced il-6 and leukemia inhibitory factor production by rheumatoid arthritis synoviocytes and its regulation by th2 cytokines," *J Immunol*, vol. 161, no. 1, pp. 409–14, 1998.

[113] S. L. Gaffen, "Recent advances in the il-17 cytokine family," *Curr Opin Immunol*, vol. 23, no. 5, pp. 613–9, 2011.

[114] P. Miossec and J. K. Kolls, "Targeting il-17 and th17 cells in chronic inflammation," *Nat Rev Drug Discov*, vol. 11, no. 10, pp. 763–76, 2012.

[115] A. Y. Tilahun, M. Holz, T. T. Wu, C. S. David, and G. Rajagopalan, "Interferon gamma-dependent intestinal pathology contributes to the lethality in bacterial superantigen-induced toxic shock syndrome," *PLoS One*, vol. 6, no. 2, p. e16764, 2011.

[116] M. Bachmann, K. Horn, I. Rudloff, I. Goren, M. Holdener, U. Christen, N. Darsow, K. P. Hunfeld, U. Koehl, P. Kind, J. Pfeilschifter, P. Kraiczy, and H. Muhl, "Early production of il-22 but not il-17 by peripheral blood mononuclear cells exposed to live borrelia burgdorferi: the role of monocytes and interleukin-1," *PLoS Pathog*, vol. 6, no. 10, p. e1001144, 2010.

[117] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, pp. 21–27, January 1967.

[118] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," 1995.

[119] S. Mukherjee, P. Tamayo, S. Rogers, R. Rifkin, A. Engle, C. Campbell, T. R. Golub, and J. P. Mesirov, "Estimating dataset size requirements for classifying dna microarray data," *J Comput Biol*, vol. 10, no. 2, pp. 119–42, 2003. Mukherjee, Sayan Tamayo, Pablo Rogers, Simon Rifkin, Ryan Engle, Anna Campbell, Colin Golub, Todd R Mesirov, Jill P Comparative Study Journal Article Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, Non-P.H.S. United States J Comput Biol. 2003;10(2):119-42.

[120] R. Figueroa, Q. Zeng-Treitler, S. Kandula, and L. Ngo, "Predicting sample size required for classification performance," *BMC Medical Informatics and Decision Making*, vol. 12, no. 1, p. 8, 2012.

[121] V. Indira, R. Vasanthakumari, and V. Sugumaran, "Minimum sample size determination of vibration signals in machine learning approach to fault diagnosis using power analysis," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8650 – 8658, 2010.

Appendix A. List of Abbreviations

Abbreviations	Definition
CM	Confusion Matrix
CRS	Cytokine Release Syndrome
CTG	Cardiotocography
DLBCLs	Diffuse Large B-Cell Lymphomas
DTC	Decision Tree Classification
ELISA	Enzyme-linked Immunosorbent Assay
FHR	Fetal Heart Rate
HCA	Hierarchical Clustering Analysis
ITML	Information-Theoretic Metric Learning
KITML	Kernelized Information-Theoretic Metric Learning
KNN	k-Nearest Neighbor
LMNN	Large Margin Nearest Neighbors
mAbs	Monoclonal Antibodies
PCA	Principal Component Analysis
QSAR	Quantitative Structure Activity Relationship
RBF	Radial Basis Function
SE-DML	Severity Estimation using Distance Metric Learning
SVM	Support Vector Machine
TCGA	The Cancer Genome Atlas
UC	Uterine Contraction
UCI	University of California-Irvine

Appendix B. List of Symbols

Symbol	Definition
A	learned positive semi-definite m by m parameter matrix in a Mahalanobis distance metric
A_0	prior positive semi-definite m by m parameter matrix in a Mahalanobis distance metric
β	projection parameter (Lagrange multiplier) corresponding to the current constraint
C	number of classes in a data set
D	inequivalent constraint set
$d(\mathbf{x}, \mathbf{y})$	distance between $\mathbf{x}$ and $\mathbf{y}$
$d(A_0 \| \| A)$	distance between $Pr(x\|A)$ and $Pr(x\|A_0)$
$\mathbf{e}_i$	unit basis vectors in which only the entry i is 1 and the rest are 0
$\mathbf{E}^+$	positive control
$\mathbf{E}^-$	negative control
$\mathbf{E}_i^?$	group i with unknown severity level
γ	kernel parameter in radial basis function kernel
FN	number of false negatives
FP	number of false positives
K	learned positive semi-definite n by n kernel matrix in a distance metric
K_0	prior positive semi-definite n by n kernel matrix in a distance metric
l	lower bound of the constraints
m	number of features in a sample
n	number of samples in a data set
P_{macro}	precision in macro-averaged F1
R_{macro}	recall in macro-averaged F1
$Pr(x\|A)$	multivariate Gaussian distribution where A^{-1} is the covariance matrix
Σ	Covariance Matrix
s_i	silhouette coefficient for i_{th} sample
S	equivalent constraint set
TN	number of true negatives
TP	number of true positives
u	upper bound of the constraints
W	test statistic in Wilcoxon signed-ranks test
$\mathbf{X}$	a set of data points
y_i	severity level for group i

Appendix C. 10-Fold Cross Validation for Decision Tree Classification

A 10-fold cross validation [118] was used here to estimate the DTC models classification accuracy on new data with unknown class labels. This method consists of splitting known data into 10 equal parts, about 90% of the data were used to train the algorithm and the classification accuracy was tested with the remaining 10% of the data (cross validation). This process was repeated 10 times using different parts of the data for training and cross validation. The classification accuracy estimate was defined as the average classification accuracy for the 10 iterations. The classification accuracy estimate of our DTC model built using training data set 1 was 98.1%; the corresponding Confusion Matrix is shown in Table C.1. For training data set 2, the average classification accuracy for the DTC model was 96.2%. The Confusion Matrix is shown in Table C.2.

Table C.1: Confusion Matrix of DTC model in Figure 5.6(a) for 10-fold cross validation

		Classifier Outcome		Test accuracy (98.1%)
		Safe	CD28	
Known	Safe	132	4	False alarm/False positive 2.9%
	CD28	0	80	Misdetection/False negative 0%

Table C.2: Confusion Matrix of DTC model in Figure 5.7(a) for 10-fold cross validation

		Classifier Outcome		Test accuracy (96.2%)
		Safe	CD28	
Known	Safe	179	5	False alarm/False positive 2.7%
	CD28	5	75	Misdetection/False negative 6.25%

Appendix D. Sample Size Requirement Assessment for Binary Severity Estimation

In order to perform clustering (by PCA followed by K-means clustering) or classification (by DTC), we need to estimate sample size. There exist several approaches that could be used to perform this estimation, depending on the data set and focus of study [75, 119, 120, 121]. We present in this appendix several approaches that proved useful for our datasets.

D.1 PCA Followed by K-means Clustering

First we estimate sample size for PCA followed by K-means clustering. We select 80 samples at random from each of the negative control sets (PBS and AutoPlasma) and the Anti-CD28 SA set for a total of 240 samples. This subset of data was used to estimate sample size.

Before we estimate sample size for unlabeled data, it is instructive to use labeled sets for reference. In our case we had a labeled set of PBS, AutoPlasma and Anti-CD28 SA. We performed PCA followed by K-means clustering, specifying 3 clusters. First we used only 6 samples from the dataset, and applied PCA followed by K-means clustering 10 times. We then computed the average error rate by comparing the clustering results to the labels of the data. Next, we included three more samples (selected at random) in the dataset and applied the algorithm again. We repeated this procedure until we reached all 240 samples available for analysis. The results are illustrated in Figure D.1(a). It shows that the error rates stabilized after about 100 samples, indicating that this is approximately the lower bound on sample size.

When labels are not available, we can generate artificial data with the same dis-

tribution as the data we need to process, and employ a similar process. In our case, we generated three groups of artificial data using the statistics from the samples of PBS, AutoPlasma and Anti-CD28 SA. We generated 240 samples of artificial data in total. The sample size estimation was performed through the same procedure as with assay data, and the results of this assessment are illustrated in Figure D.1(b). Here the error rates seem to stabilize after about 80 samples, a result sufficiently close to the values found with the real data (which was 100). The use of artificial data would probably require the use of a guard band, adopting a somewhat higher sample size in practice than the one that was estimated in simulations.

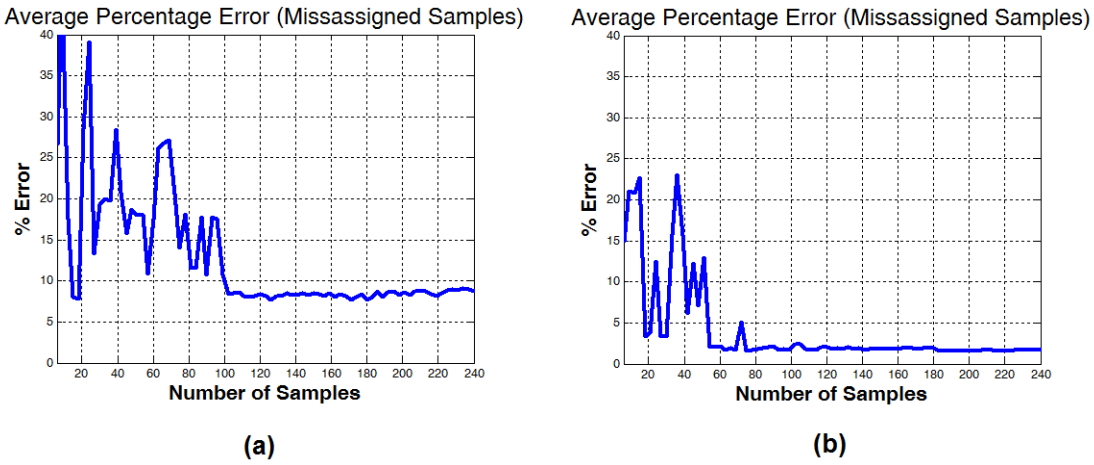


Figure D.1: Percentage error as function of the number of samples used: (a) Measured data; (b) Artificially generated data

When the labels for the handled data are unknown, there are several other valid approaches. As an example, we used the silhouette metric [75] to estimate the adequate number of samples necessary to get consistent results with PCA followed by K-means clustering. The silhouette metric provides an assessment of how good the

clustering results are, irrespective of the clustering method. It measures how close data samples are placed together within defined clusters, and how far away the clusters are from each other. This metric is commonly used to assess the optimal number of clusters k in K-means clustering [75]. The value provided by the silhouette metric is calculated for different values of k (i.e. $k = 2, 3, 4, \ldots, 9$), and the value of k that maximizes the metric is chosen.

We used the silhouette metric to obtain optimal values of k for different sample sizes out of our 240-sample set. We started by choosing 12 samples at random to develop the estimate. We repeated the estimation by increasing the sample set by 12 samples at a time, until all 240 samples were included. The process was repeated 10 times. The average optimal values for k for each sample size are plotted in Figure D.2. The number of clusters stabilized at $k = 3$ after 120 samples. The approach could be used for other data sets in a similar manner.

D.2 Decision Tree Classification

We applied DTC on a training data set with samples from the "CD28" class and samples from the AutoPlasma and PBS classes. We started with five (5) samples in each class, and increased the sample set of each class by 1 sample each time, till there were 80 samples in each class. We built the DTC model for each sample set. The accuracy of the models is shown in Figure D.3(a). We observed that the accuracy levels were inconsistent for small sample sizes. The accuracy level stabilized for sample sizes that exceed 20. Moreover, when the sample size of each class was below 20, the root nodes of the DTC models appeared to be picked randomly from all 11 cytokines. When the sample size of each class was above 20, almost all the root nodes of the DTC models were IFN-γ.

Next we performed the process of estimating sample size by using synthetic data.

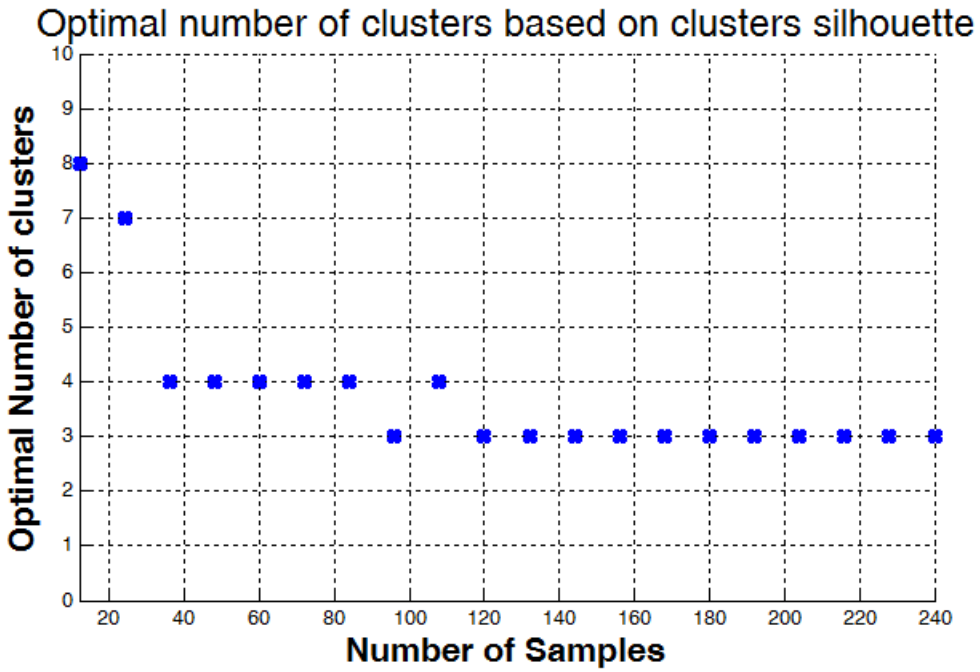


Figure D.2: Optimal number of clusters for different sample sizes, based on the silhouette metric

Synthetic data were generated based on the statistics of two classes: CD28 class (class 1) and samples from PBS and AutoPlasma (class 2). DTC was applied on these two classes. The accuracies for different sample sizes are shown in Figure D.3(b). The accuracy stabilizes after 50-60 samples. Since we have two classes, the total number of required samples is 100 -120.

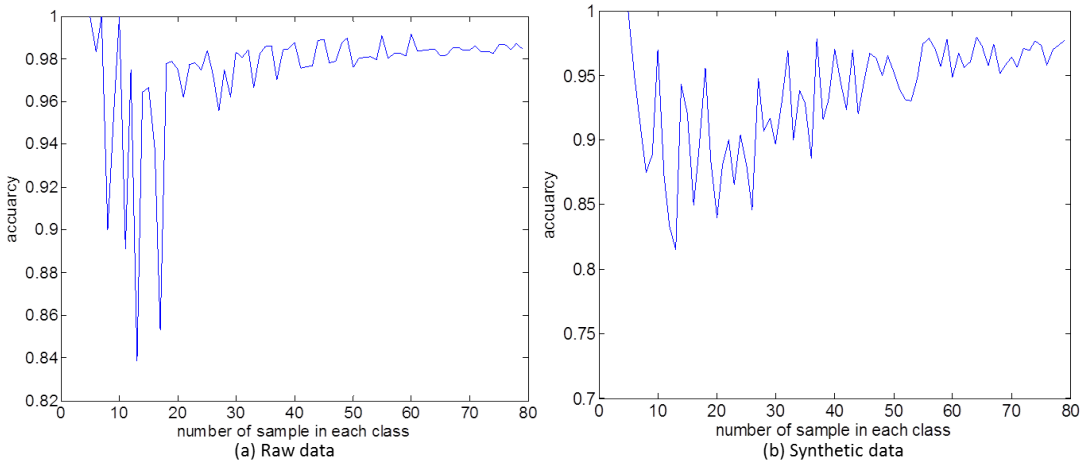


Figure D.3: Optimal number of clusters for different sample sizes, based on the silhouette metric

Appendix E. List of Publications

- F. Xiong, M. Janko, M. Walker, D. Makropoulos, D. Weinstock, M. Kam, L. Hrebien, "Analysis of cytokine release assay data using machine learning approaches", *International Immunopharmacology*, Volume 22, Issue 2, October 2014, Pages 465-479, ISSN 1567-5769

- F. Xiong, M. Kam, L. Hrebien, and Y. Qi, "Ranking with Distance Metric Learning for Biomedical Severity Detection", in *3rd Workshop on Data Mining for Medicine and Healthcare*, Apr 2014, Philadelphia

- F. Xiong, B. R. Hipszer, J. Joseph, and M. Kam, "Improved blood glucose estimation through multi-sensor fusion," in *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, Aug 2011, pp.377-380.

- F. Xiong, L. Bai, "Interoperable wireless sensor network model using multi-agent-based middleware," *Intelligent Signal Processing and Communication Systems (ISPACS), 2010 International Symposium on* , vol., no., pp.1,4, 6-8 Dec. 2010

- F. Xiong, L. Bai, F. Ferrese, "Multi-agent-based interoperable wireless sensor network model," *Proceedings of IEEE Sensors*, 2009. p. 1427-1432

- L. Bai, F. Xiong; M. Korostelev, S. Biswas, "Optimal Updating Time Using Theory of Reliability," *Parallel and Distributed Systems, 2008. ICPADS '08. 14th IEEE International Conference on* , vol., no., pp.439,446, 8-10 Dec. 2008

- F. Xiong, M. Kam, L. Hrebien, and Y. Qi, "Kernelized Metric Learning for Cancer Diagnosis using High-Dimensional Molecular Profiling Data", in *ACM TKDD Special Issue: Connected Health at Big Data Era*, under review