Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Allison W Irvine

Entitled
Computational Analysis of Flow Cytometry Data

For the degree of Master of Science

Is approved by the final examining committee:

Dr. Murat Dundar
Chair

Dr. Mihran Tuceryan

Dr. Snehasis Mukhopadhyay

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Dr. Murat Dundar

Approved by: Dr. Shiaofen Fang 04/25/2012
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

Computational Analysis of Flow Cytometry Data

For the degree of Master of Science

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Allison W Irvine

Printed Name and Signature of Candidate

04/25/2012

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

COMPUTATIONAL ANALYSIS OF FLOW CYTOMETRY DATA

A Thesis

Submitted to the Faculty

of

Purdue University

by

Allison W. Irvine

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2012

Purdue University

Indianapolis, Indiana

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. Murat Dundar, for guiding me and supporting me during my graduate studies here at IUPUI. I would also like to thank my committee members, Dr. Mihran Tuceryan and Dr. Snehasis Mukhopadhyay. I would like to thank Dr. Tuceryan and Dr. Dundar for teaching the machine learning courses from which I learned many important techniques.

I would also like to express appreciation towards Dr. Bartek Rajwa from the Bindley Bioscience Center at Purdue University for providing me with data and teaching me about flow cytometry, as well as encouraging me to be enthusiastic on the subject.

I would like to thank Mr. Andy Harris for providing guidance in developing my teaching skills as a T.A. and encouraging me to be the instructor for a course this semester.

Finally, I would like to thank Joshua Morrison for helping me get my paperwork and formalities together throughout this time.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Irvine, Allison W. M.S., Purdue University, August 2012. Computational Analysis of Flow Cytometry Data. Major Professor: Murat Dundar.

The objective of this thesis is to compare automated methods for performing analysis of flow cytometry data. Flow cytometry is an important and efficient tool for analyzing the characteristics of cells. It is used in several fields, including immunology, pathology, marine biology, and molecular biology. Flow cytometry measures light scatter from cells and fluorescent emission from dyes which are attached to cells. There are two main tasks that must be performed. The first is the adjustment of measured fluorescence from the cells to correct for the overlap of the spectra of the fluorescent markers used to characterize a cell’s chemical characteristics. The second is to use the amount of markers present in each cell to identify its phenotype. Several methods are compared to perform these tasks. The Unconstrained Least Squares, Orthogonal Subspace Projection, Fully Constrained Least Squares and Fully Constrained One Norm methods are used to perform compensation and compared. The fully constrained least squares method of compensation gives the overall best results in terms of accuracy and running time. Spectral Clustering, Gaussian Mixture Modeling, Naive Bayes classification, Support Vector Machine and Expectation Maximization using a gaussian mixture model are used to classify cells based on the amounts of dyes present in each cell. The generative models created by the Naive Bayes and Gaussian mixture modeling methods performed classification of cells most accurately. These supervised methods may be the most useful when online classification is necessary, such as in cell sorting applications of flow cytometers. Unsupervised methods may be used to completely replace manual analysis when no training data is given. Expectation Maximization combined with a cluster merging post-processing step gives the best results of the unsupervised methods considered.

CHAPTER 1 INTRODUCTION

1.1 Flow Cytometry

Flow cytometry is a technique for rapidly measuring cell characteristics of large numbers of cells. Cells are tagged and/or stained to highlight components (proteins and genes for example) present in the cell. Then the cells are passed one by one through a tube using hydrodynamic forces [5,26]. Lasers are aimed at the tube, and as the cells pass through the tube, they scatter light, showing cell shape, size and the amount of a tagged/stained component in the cell. Particles labeled with fluorochromes are attached to cell surface receptors [26]. Fluorochromes are a type of dye that emits fluorescent light when excited with a laser. The fluorescence emission is detected by a series of bandpass photodetectors, where the number of photodetectors varies depending on the flow cytometer. In 2010, flow cytometers had been developed that were capable of measuring up to 18 different cell surface markers at once, and that number is continually increasing, allowing the identification of more cell types [21]. Forward scatter (FSC) and side scatter (SSC) are measurements of light reflection from the cell at different angles and are independent of fluorescence spectra. Thousands of cells per second can be analyzed [21,36]. The resulting data, commonly referred to as FCM data (Flow CytoMeter data), is an n x d matrix, where n is the number of cells analyzed and each of the d dimensions is the amount of a component present in each cell.

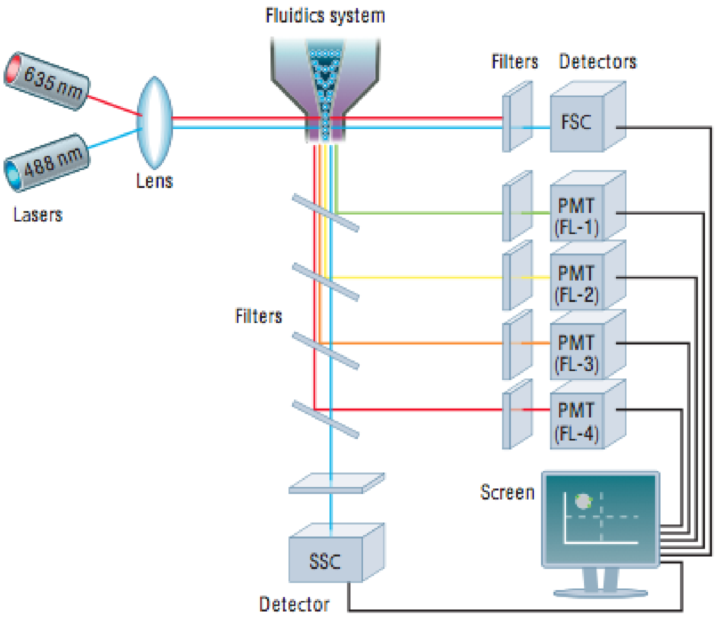

Figure 1.1 A Typical Flow Cytometer Setup [26]

1.2 Motivation

Flow cytometry is a versatile tool for the analysis of cells, revealing information about cell cycle stages, DNA and protein content. It is used in a variety of biology-related fields as a large-scale quantitative technique.

It is widely used in immunology and pathology, and has become an effective way to diagnose cancer. Quantitative analysis of tumor cell heterogeneity is made possible and efficient by means of flow cytometry analysis [4]. Flow cytometry can be used to quantify abnormal DNA content, which is an indicator of the malignancy of a tumor [4]. Leukemia, lymphoma, and myeloma are some diseases which may be diagnosed by this quantification [26]. Because the information is quantitative as opposed to qualitative histological diagnosis from images, flow cytometry can also be used to more accurately measure changes over time from the development of a disease or the use of a therapeutic treatment.

Flow cytometry is also commonly used in marine biology. One popular use is in the cell cycle analysis of prokaryotes. This is used to measure the growth rate of phytoplankton in bodies of water [23]. Phytoplankton live near the surface of bodies of water and create organic compounds from carbon dioxide and sunlight, making them an essential part of the aquatic food chain and thereby indicating the quality of an aquatic environment.

Currently the analysis of FCM data is performed manually, and since high-throughput analysis results in feature vectors for thousands of cells for a single biological sample, automated methods are needed to avoid many hours of unnecessary human labor [1,5,9,13,19,21,25,35,36].

1.3 Objective

Given a large multidimensional dataset of numerical cell attribute values, the goal is to develop automated methods to identify the phenotype of each cell. Cell types which are expected to appear as well as rare types and new types should be detected. From a clinical standpoint, the amount of different cell types in a human’s blood can diagnose certain life-threatening diseases and quantify the effects of treatment. From a more research-oriented view, flow cytometry provides an efficient means of analyzing variations in biological systems and organisms.

1.4 Application Considerations

1.4.1 Biological Variation

Often the number of cells in a cell type group varies between data files, introducing bias in a trained classifier that would not be appropriate for unseen data [13]. Training is also difficult to generalize to unseen data because of offset due to experimental factors such as instrument settings and difference of manufacturers of antibodies [13]. FCM data often includes outliers due to the cell preparation and variations in equipment [5].

1.4.2 Current Data Analysis Methods

The identification of cell types of interest is typically performed by manual analysis by trained experts. Preprocessing includes compensation, which is the process of calculating the proportion of each dye present in a cell from the detected light in each spectral band. The next step is gating. Classification of cell phenotypes is accomplished by this process [5,21,26,34]. The purpose of gating is to identify populations of different cell types within a biological sample, for example blood or cell cultures, that has been run through a flow cytometer. Gating is a process where a user views a plot of the FCM data, selects a region of interest, and then views the data in 2 or 3 dimensions at a time, manually drawing boundaries around potential clusters and refining the boundary using a histogram threshold [5,26]. This is the only ground truth available and is highly variant depending on the lab in which the data was produced and the scientist who performed the labeling [5,13].

CHAPTER 2 METHODS

This paper compares methods for the automated performance of the three main tasks involved in analyzing FCM data. These three main steps are gating on forward and side scatter, compensation, and cell type classification.

2.1 Overview of Methods

2.1.1 Gating on FSC and SSC

Before phenotype analysis is performed, the measured forward scatter and side scatter is used to distinguish between lymphocytes, monocytes, granulocytes and cellular debris [16,20]. Lymphocytes, monocytes, and granulocytes are three different types of white blood cells. In cancer diagnosis, the informative cell phenotypes are lymphocyte subtypes, so lymphocytes are identified first and then the fluorescence values are used to classify them into subclasses.

2.1.2 Compensation

Compensation is a process in which the actual amount of each dye present in a cell is inferred from the measured fluorescence in each spectral band. Usually the fluorochromes which are used to stain a biological sample are chosen such that their emission spectra occur mainly within separate bandwidths. However, the emissions often have some amount of overlap [27]. The estimated spectral emissions from each pure dye are estimated by analyzing samples stained with only a single dye at a time. Then methods using unconstrained and constrained optimization are used to minimize the error between the observed signal and the proportion of each dye present multiplied by the spectral emissions of the pure dyes.

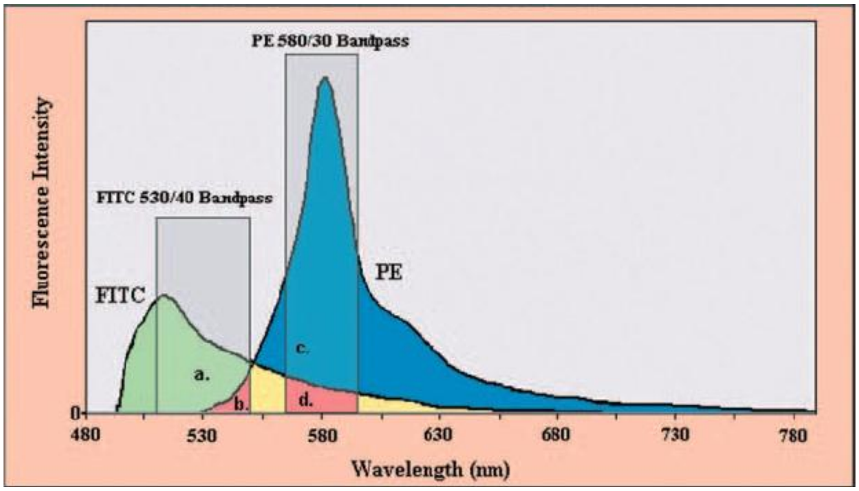

Figure 2.1 Fluorescence Overlap. The fluorescence emission from the fluorochromes FITC and PE. Each dye primarily emits light within separate spectral bands, but there is overlap between the entire emission spectra of the two dyes. [34]

2.1.3 Cell Type Classification

The identification of informative cell types is done by using the fluorescence emission from each cell to determine the amount of certain labeled components attached to a cell. After compensation, each cell is represented by the abundance of each dye attached to it. These dyes, or fluorochromes, indicate the amount of dyed particles attached to cell surface receptors. The amount of different types of proteins attached to a cell indicate the cell type. In its most basic form, the problem is to classify a large number of *d*-dimensional data points, where each data point represents a cell. Several supervised and unsupervised methods are compared. The supervised methods used are Gaussian mixture models, Support Vector Machine, and Naive Bayes as a baseline. Unsupervised methods used are Spectral Clustering and Expectation Maximization using a Gaussian mixture model.

2.2 Data Description

2.2.1 FlowCap-I Dataset

The FCM data used to test methods for phenotype classification described in this paper was collected for the FlowCap-I competition in 2010 [10]. It consists of 5 datasets. Human experts label all of the data numerically. Since the labels given are numeric, the actual cell type names are not known. This data is already compensated and gated on forward and side scatter by the researchers who analyzed and provided the data. That is why this dataset cannot be used to test compensation methods.

The first dataset is the Diffuse Large B-cell Lymphoma set (referred to as “Lymph” in the file set). Cells from 30 lymph node biopsies were stained for three markers and analyzed by a flow cytometer. All of the patients were treated at the British Columbia Cancer Agency (BCCRC) and were confirmed to have diffuse large B-cell lymphoma. The dataset was provided by the BCCRC.

The second dataset is the Symptomatic West Nile Virus set (referred to as “CFSE” in the file set). Peripheral blood mononuclear cells stimulated in-vitro with peptide pools to approximate the West Nile Virus polyprotein were analyzed. The samples are from patients infected with symptomatic West Nile Virus. The dataset was provided by McMaster University.

The third dataset is the Normal Donors set (referred to as “NDD” in the file set). The purpose of the analysis was to observe the differences in response of several cell types to various stimuli. The samples were taken from healthy human donors. The dataset was provided by Amgen Inc.

The fourth dataset is the Hematopoietic Stem Cell Transplant set (referred to as “StemCell” in the file set). There are 30 samples derived from hematopoietic stem cell transplants performed at the Terry Fox Laboratory. The dataset was provided by the BCCRC.

The fifth dataset is the Graft versus Host Disease (GvHD) set (referred to as “GvHD” in the file set). The purpose of the study was to find a way to detect GvHD early in patients. The dataset was provided by the BCCRC and Treestar Inc.

2.2.2 Bindley Bioscience Center Dataset

FCM data acquired from blood samples from healthy donors was provided by Dr. Bartek Rajwa at the Bindley Bioscience Center. 5 dyes were used to label the dataset, and indicate the presence of lymphocyte subtypes B-cell, T-cell, and cytokine. 15 blood samples stained with these dyes are included, as well as one sample for each one of the 5 dyes, stained only with that dye. These files are called the “controls” and are used to estimate the fluorescence spectrum of each pure dye. This data was used to compare compensation methods.

CHAPTER 3 LYMPHOCYTE GATING

Flow cytometry is often used for the identification and quantification of lymphocyte subsets [20]. Most of the datasets used for this project all involve analysis of lymphocyte populations. A sample that is analyzed by a flow cytometer often contains several types of cells, cells that are stuck together, uninformative debris and dead cells. Forward scatter (FS) and side scatter (SS) are used to identify populations of lymphocytes from the total analyzed sample. FS and SS are collected by every flow cytometer and do not analyze spectral content, only the overall amount of light. FS is collected by a lens that is facing the lasers and SS is measured at a 90-degree angle to the lasers [26]. FS gives information about the object size and can be used to distinguish between cellular debris and living cells [26]. SS gives information about the granularity of the object.

The major populations of cells usually present in a sample are lymphocytes, monocytes, and granulocytes [16,20]. These populations are expected to have the same relative positions to each other in a sample [16,20]. Before analyzing fluorescence, gating on FS and SS is performed. In order for the control matrix to properly represent the spectral signatures of the dyes with respect to lymphocytes, it must be calculated from lymphocytes only. Some of the other cell types may acquire fluorescent antibodies during staining, and will change the calculated spectral signatures in the control matrix.

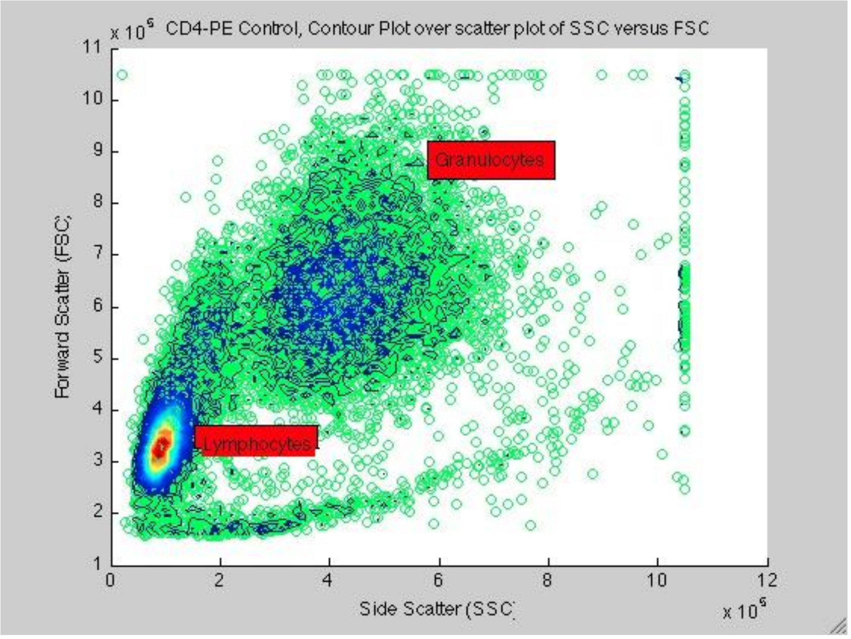

Figure 3.1 Side Scatter versus Forward Scatter. A contour plot of the histogram is superimposed on top of the untransformed data.

The Bindley Bioscience Center dataset was gated for lymphocytes before compensation, as mentioned in the previous section. Note from Figure 3.1 that many noisy cells have FSC and SSC values that take on a maximum value, due to the limits of the photodetectors in the flow cytometer. There are three distinct clusters, but a lot of noise is present as well. In order to identify the lymphocyte population, the knowledge of the relative population location in this feature space was used along with the observations just mentioned in a heuristic method.

First the cells with maximum FSC and SSC values were removed. Then a 2-dimensional histogram was created of the data using the FSC and SSC features. The bin counts of the histogram were clustered into two groups using Expectation Maximization to separate noise from distinct clusters. Then the bins in the cluster with the higher-valued mean were retained, and clustered again based on the FSC and SSC values into 3 clusters. The cluster with the lowest mean SSC value was identified as the lymphocyte population. The mean and covariance of this cluster were used to calculate the probability of every cell belonging to this Gaussian class, and those cells with a probability of at least .001 were marked as lymphocytes.

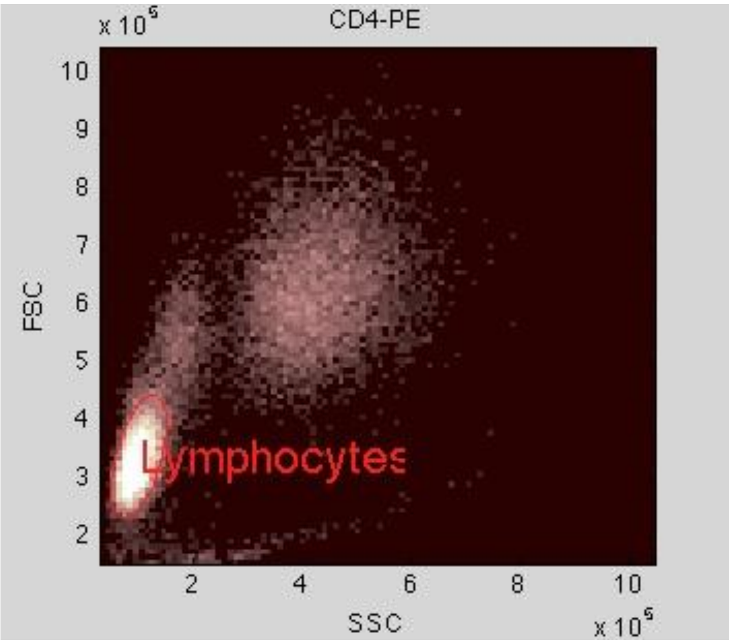

Figure 3.2 Results of Automated Lymphocyte Gating. An ellipse is drawn around the mean of the class identified as lymphocytes. Code for drawing the ellipse was written by A. Maida, obtained from the course website http://www.cacs.louisiana.edu/~maida/Classes/cmps523/cmps523.htm.

CHAPTER 4 COMPENSATION

Compensation is a preprocessing step performed before any classification can occur. The raw data acquired by the flow cytometer is the amount of light detected in each spectral band for which there is a photodetector. Separate photodetectors are filtered to measure mostly light from one fluorochrome [15]. When multiple dyes are used in a experiment, fluorescent dyes with limited overlap are used to ensure that each photodetector will measure the color from primarily one dye [3]. However, the spectra of different dyes may overlap. The spectra emitted by a dye may fall within multiple detector bands, and compensation is almost always necessary to correct this [3,27].

In order to determine what is known as the spectral signature of a dye, a control is run through the flow cytometer. A control is a biological sample stained only with the dye for which we would like a spectral signature [15]. The average output from all of the cells in the control sample is used to form a single vector of spectral band outputs, which is the spectral signature. Often a spectral signature is acquired for autofluorescence, a small amount of natural fluorescence emitted by subcellular structures in cells, by running an unstained biological sample through the flow cytometer [27].

Once spectral signatures are acquired for every dye used in the experiment, they are used to determine how much of each dye is present in the analyzed biological sample. The most common practice currently is to subtract the amounts of spectral overlap from the total observed signal [15]. In that case it is assumed that each detector is primarily detecting one dye, and the compensated output from a detector is interpreted as the amount of the dye it is primarily measuring. The amount of each dye present in a cell is positively correlated with the amount of some component, such as a protein, present in the cell. The amounts of these components are used to determine the phenotype of the cell.

4.1 Mathematical Representation of Compensation

Let d be the number of observed spectral bands. Let $\mathbf{r}$ be the observed d x 1 vector of measurements for a single cell. Let $\mathbf{M}$ be the d x c control matrix, where c is the number of dyes being used in the experiment. By analyzing controls stained with a single dye each, we can calculate the average signal detected within each spectral band for a single dye [3]. This is the dye's d x 1 spectral signature vector, and the matrix $\mathbf{M}$ contains the spectral signatures of all dyes used in the experiment. By analyzing an unstained control, we can also obtain a spectral signature, $\mathbf{n}$, for autofluorescence noise [3].

Autofluorescence, as described earlier, is a small amount of fluorescence naturally emitted by certain subcellular structures. The spectral signature of a single observation can be represented by the linear regression model [14]:

Equation 4.1

$$\mathbf{r} = \mathbf{Ma} + \mathbf{n}$$

where $\mathbf{a} = \{a_1, a_2, ..., a_c\}$ is a c x 1 vector representing the abundance of each dye present in the observation. In other words, the spectral signature of a single observation is the product of the amount of each dye present and the spectral signatures of the individual dyes, plus noise from autofluorescence. If an autofluorescence control is provided, we calculate the average spectral output of this control sample as we would a control for a dye. Then, before performing any compensation, the estimated autofluorescence spectral signature is subtracted from the data.

The representation of the observed data as a linear model is used to formulate optimization algorithms to calculate the amount of each dye present in each observed cell. Optimization is appropriate for this problem because constraints should be imposed on the results for them to be physically meaningful. In addition, there may be experimental variables, dye-dye interactions or dye-cell interactions that can affect the spectrum of the dye, and these interactions are biologically variant and cannot be modeled exactly [27]. Therefore, approximating the dye abundances by minimizing the error between the observation and the model is the approach used.

4.1.1 Constraints

Two constraints may be placed on this linear model: nonnegativity and sum-to-one. In reality, we are measuring fluorescent output from a fluorochrome. The minimum this value can be is 0, which is not output. Also, the minimum amount of dye present in a cell is 0. Therefore, a nonnegativity constraint should be applied to the abundance values as well as the spectral signatures of controls and observations. Sometimes the flow cytometer will compensate for background fluorescence and automatically subtract this from the measurements being taken, resulting in negative spectral intensity values. Nonnegativity is easily imposed on the measured spectral signatures by adding an appropriate amount to all observed values. The nonnegativity constraint on dye abundances is enforced in the optimization process. The second constraint, which is referred to as the sum-to-one constraint, states that the abundances of each dye present in a cell should sum to one. In other words, the total spectral output of a cell is the summation of the fluorescent emissions produced by all of the dyes present in a cell (and autofluorescence).

4.2 Literature Review

One of the first to consider the problem of compensation in flow cytometry as a mathematical model was Bagwell [3]. Compensation was traditionally performed in the hardware during data acquisition based on manual potentiometer settings, but in [3], matrix operations were performed to compensate the data post-acquisition.

4.2.1 Relationship to Hyperspectral Imaging

Although there has been little theoretical work on compensation within the field of flow cytometry, there has been much research on a similar problem in the field of hyperspectral imagery. In hyperspectral imaging, a single pixel in a scene often contains several different objects such as water, soil or vegetation. Each of these objects has its own spectral signature. “Spectral unmixing” is performed to determine how much of each object is present in a pixel, and therefore determine what objects are present in an area whose picture was taken by a remote sensing platform from a high altitude where there is low spatial resolution [18]. This problem is also called “linear decomposition” and various solutions are applied in spectral karyotyping, bright-field microscopy, and live-imaging analysis [11]. Due to the additive properties of fluorescence emission spectra, the same techniques used for spectral unmixing can be applied to the problem of compensation [11]. Instead of a pixel, we are considering the total fluorescence emission from a cell where the spectral output of all dyes present in a cell produce a combined output. Each dye can be considered as an object present in a cell, and each dye has a spectral signature. Although we are taking measurements from discrete spectral bands in flow cytometry which determine the dimensionality of the output, the algorithms for spectral unmixing do not make assumptions about the dimensionality of the data. In Bagwell’s method it is assumed that there is one spectral band for each dye being measured, creating a square matrix of spectral signatures for all dyes [3]. This allows one to explicitly solve for the abundances. However, by using spectral unmixing methods, we are not restricted to having the number of photodetectors equal the number of dyes being used in an experiment. We can measure more channels to possibly improve results, or we may use more dyes than available photodetectors. The following section will describe various methods for spectral unmixing that may be applied to flow cytometry data.

4.3 Unconstrained Compensation

Recall that controls are samples stained with a single dye, and there is one for each dye being used in an experiment. The common practice in flow cytometry data compensation is to solve explicitly for the abundance vector **a** in the linear model of Equation 4.1 [3,29,32]. This approach assumes that each dye is principally detected in a single photodetector in the flow cytometer and each dye has a primary spectral band. Therefore the number of dyes, *c*, is equal to the number of spectral bands being measured, *d*, and the matrix **M** is square. If this is the case, the solution for the abundances is given by [3,32].

Equation 4.2 $$\mathbf{a} = \mathbf{M}^{-1}(\mathbf{r} - \mathbf{n})$$

This result is unconstrained, meaning fluorescence values may be negative and the abundances do not have to sum to one.

4.4 Unconstrained Least Squares

If assumptions are not made about the matrix **M** being square, that is, if it is not assumed that there is the same number of photodetectors as dyes being used in an experiment, we can obtain a closed-form solution for the abundances using the pseudo-inverse of the control matrix **M** [14,18].

Equation 4.3 $$\mathbf{a} = (\mathbf{M}^{T}\mathbf{M})^{-1}\mathbf{M}^{T}(\mathbf{r} - \mathbf{n})$$

In order for this result to be valid, the dimensionality of the data must be greater than the number of dyes being used.

4.5 Orthogonal Subspace Projection

The method of Orthogonal Subspace Projection (OSP) considers one target at a time [14]. In the case of flow cytometry, a “target” refers to the dye whose abundance will be calculated in the current application of OSP. The goal of OSP is to project the observed data onto a vector that is orthogonal (independent) to the spectral signatures of non-target dyes. After this projection, the abundance of the target dye is estimated in the least squares sense [6,14]. To create the OSP projector, first the spectral signature of the target dye is removed from the matrix **M**, giving us a vector, **m***, the spectral signature of the target dye, and **M***, the *d* x (*c*-1) matrix of the spectral signatures of the non-target dyes. Also, let a_t be the abundance of the target dye, and let the vector $\mathbf{a}_{t-}$ be the abundances of the non-target dyes. Then the linear model can be rewritten as [6,14]

Equation 4.4 $$\mathbf{r} = (\mathbf{m}^*)a_t + (\mathbf{M}^*)\mathbf{a}_{t-} + \mathbf{n}$$

The orthogonal subspace projector *P* is

Equation 4.5 $$P = \mathbf{I} - \mathbf{M}^* f(\mathbf{M}^*)$$

and *f*(M*) is the pseudo-inverse of **M***.

Equation 4.6 $$f(\mathbf{M}^*) = (\mathbf{M}^{*T}\,\mathbf{M}^*)^{-1}\mathbf{M}^{*T}$$

The OSP linear model is obtained by multiplying the orthogonal subspace projector to both sides of the original linear model from Equation 4.1, removing the non-target spectral signatures **M***.

Equation 4.7 $$P\mathbf{r} = P\mathbf{m}^* a_t + P\mathbf{n}$$

In this case the noise is also suppressed, so we do not need to estimate and subtract noise from autofluorescence from the data.

If we assume that $P\mathbf{m}^* a_t$ is drawn from a normal distribution and the projected noise $P\mathbf{n}$ is drawn from another normal distribution, maximizing the Fisher's criterion function with respect to a weight vector **w** for the two classes gives the optimal value of **w**, **m***, as the orthogonal subspace projector. Multiplying this value by the projected value of the observed signal r results in the orthogonal subspace projection classifier Ψ for the target dye *t*.

Equation 4.8 $$\Psi = (\mathbf{m}^{*T})P\mathbf{r}$$

The projected value Ψ of an observed cell is positively correlated with the amount of the target dye present in the cell. This process is done for each of the dyes, each time using a different dye as the target. The resulting values are normalized from 0 to 1. This result is unconstrained.

4.6 Fully Constrained Least Squares

Although the approaches described in the previous sections address the principal problem of spectral overlap between dyes, they are not fully optimal solutions. As stated in section 4.1, there are two constraints which make the results of compensation physically meaningful. The first constraint is that all fluorescence intensities must be non-negative, because the minimum amount of fluorescent emission from a cell is 0. Therefore, we formulate the nonnegativity constraint [14,17,18]

Equation 4.9 $$a_j \geq 0 \qquad \text{for all } 1 \leq j \leq c$$

Now we turn to the actual meaning of the abundance variables a_j . A cell's fluorescent emission is the summation of the emissions of all dyes present in the cell and autofluorescence. Since the information we are unmixing is this fluorescent emission, the abundances are interpreted as the proportion of each dye that contributes to the total emission. With this definition given to **a**, we must state that the values of a sum to one, that is, each value a_j is the proportion of dye *j* present in a cell out of all dyes (and autofluorescence) that contribute towards the total fluorescent emission.

Equation 4.10 $$\sum_{j=1}^{c} a_j = 1$$

When these two constraints are considered in spectral unmixing, it is called *fully constrained* [14,17].

A fully constrained approach no longer allows us to acquire a closed-form solution to the least square error problem. The problem is now framed as a constrained least squares problem.

Equation 4.11 $$\min_a \left[\frac{1}{2} \left\| \mathbf{Ma} - \mathbf{r}_i \right\|^2 \right], \qquad 1 \le i \le N$$

subject to:

Equation 4.12 $$-\mathbf{I}_c \mathbf{a} \le 0$$

Equation 4.13 $$\mathbf{1}_c \mathbf{a} = 1$$

$\mathbf{I}_c$ is a *c*-dimensional identity matrix and $\mathbf{1}_c$ is a vector of *c* ones. The parameters of this problem can be passed to a constrained least squares optimization function, such as *lsqlin* in Matlab or *lsei* from the *limsolve* package in R. In this implementation the Matlab routine *lsqlin* was used.

4.7 Fully Constrained One Norm

A variation of FCLS is Fully Constrained One Norm. It is a variation of fully constrained least squares where we are minimizing the magnitude of the bounds on the error between the observed data **r** and the true signal, **Ma**.

4.7.1 Formulation of the Method

First we define **w** to be the error between **r** and **Ma**.

Equation 4.14 $$\mathbf{w} = \mathbf{r} - \mathbf{Ma}$$

Let **v** be the bounds on the error term **w**.

Equation 4.15 $$-\mathbf{v} \leq \mathbf{w} \leq \mathbf{v}$$

Now we augment the solution, **a**, with the slack variables **v**.

Equation 4.16 $$\mathbf{x}^T = [\mathbf{a}, \mathbf{v}]$$

x is a (*c*+*d*) length vector, since it is the concatenation of the *c*x1 vector **a** and the *d*x1 vector **v**.

Using **x** as the solution, we can formulate the linear programming problem

Equation 4.17 $$\min_{\mathbf{x}} \mathbf{fx}$$

where

Equation 4.18 $$\mathbf{f} = [0_1, \ldots, 0_c, 1_1, \ldots, 1_d]$$

f is the concatenation of a vector of *c* zeros and a vector of *d* ones. Therefore the objective function is minimizing the **v** variables within **x**. This is because we only wish to minimize **v**, not **a**. The constraints on the abundances a described in the section above must now be considered. The sum-to-one constraint can be described as an equality constraint on the problem

Equation 4.19 $$\mathbf{g}^T\mathbf{x} = 1$$

where

Equation 4.20 $$\mathbf{g} = [1_1,...,1_c,0_1,...,0_d]$$

Similar to **f**, **g** is a vector whose product with **x** yields the sum of the **a** components of **x**. The nonnegativity constraint is defined by a lower bound of a vector of zeros on the solution **x**. Both **a** and **x** must be greater than or equal to zero. To define the role of **v** in bounding the model error the following is derived:

Equation 4.21 $$-\mathbf{v} \le \mathbf{w} \le \mathbf{v}$$

Equation 4.22 $$-\mathbf{v} \le \mathbf{r} - \mathbf{Ma} \le \mathbf{v}$$

Equation 4.23 $$\mathbf{Ma} - \mathbf{v} \le \mathbf{r} \le \mathbf{Ma} + \mathbf{v}$$

Equation 4.24 $$\mathbf{A}_1 = [\mathbf{M}^T, -\mathbf{I}_d],\ \mathbf{A}_2 = [\mathbf{M}^T, \mathbf{I}_d]$$

$\mathbf{I}_d$ is a *dxd* identity matrix. $\mathbf{A}_2$ is formulated such that the product $\mathbf{A}_2\mathbf{x}$ represents the sum of **Ma** and the error bounds **v** (or –**v** for $\mathbf{A}_1\mathbf{x}$). In order to implement the upper and lower bounds simultaneously as a single input parameter to a linear problem solver, we concatenate $\mathbf{A}_1$ and $\mathbf{A}_2$ into the matrix **A**

Equation 4.25 $$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$$

and form the constraint

Equation 4.26 $$\mathbf{Ax} \le [\mathbf{r}, -\mathbf{r}]^T$$

This method was implemented using the Matlab routine *linprog*. The sum-to-one constraint is imposed by using a vector of *c* ones and *d* zeros.

Equation 4.27 $$[\mathbf{1}_c, \mathbf{0}_d]\mathbf{x} = 1$$

The nonnegativity constraint is enforced for both the abundances and the slack variables.

Equation 4.28 $$\mathbf{0}_{d+c} \le \mathbf{x}$$

4.8 Experimental Results

The Bindley Bioscience Center dataset was used to compare the results of the different methods of compensation. Unconstrained least squares (ULS), orthogonal subspace projection (OSP), fully constrained least squares (FCLS), and fully constrained one norm (FCON) were applied to the dataset. In the case of ULS and OSP, some of the resulting abundance values were negative. Therefore an adjustment was made for abundances to be positive so that the summation of abundances in the estimation of accuracy would not cancel out. For each cell, the minimum value of its abundances was added to all cell abundances.

4.8.1 Calculation of the Control Matrix

The spectral signatures of the individual dyes are based on the average spectral output from all cells in the corresponding control sample [3,27]. In this experiment, the dataset from the Bindley Bioscience center was used. For each of the 5 dyes, a control sample was measured. Recall that the control sample is a set of cells stained with only one dye.

The control matrix had to be calculated from lymphocytes only, so the files each had to have the lymphocyte population identified and extracted first. The spectral signatures would not be meaningful unless they were calculated from the correct cell type. Then the cells whose fluorescent emission contributed to the average spectral signatures were those lymphocytes whose total fluorescence was in the upper 0.75th quantile. This is to account for cells in the control sample which may not have acquired the dye being measured, for example, cells only emitting autofluorescence. This procedure is carried out for every control. Concatenating the (*1xd*) spectral signatures of each dye results in the (*cxd*) control matrix **M**.

4.8.2 Evaluating the Accuracy of Compensation

In order to empirically view the success of compensation and compare the different methods discussed, the control samples used to calculate **M** were compensated. If the compensation is accurate, all of the cells in a control sample, when compensated, should have low abundance values for every other dye, and most should have a high abundance for the target dye.

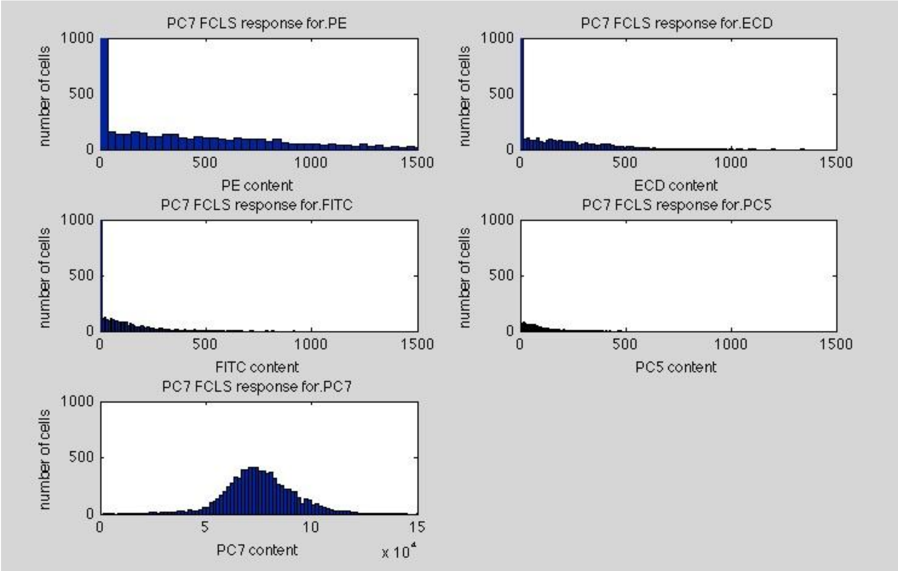

Figure 4.1 Histogram of Abundances of Each Marker. The control sample for the marker PC7 was compensated using FCLS in this case.

Although most cells in a control sample should have a high abundance of the dye they were labeled with, some cells may be of a type which does not carry that particular marker. Therefore it is more accurate to say that no cells should carry markers with which the sample was not labeled. The accuracy may be considered the 'purity' of the cells, or the proportion of the dye with which the sample was labeled to the false perception of other dyes present in the control sample. If a sum-to-one constraint is placed on abundance values, we may simply use the abundances of the target label, then take the average over all cells in the control sample. If there was no sum-to-one constraint on the abundances in the compensation method being evaluated, this measure must be normalized by the sum of all abundance values for the cell. The accuracy for the j^{th} dye may be estimated as

Equation 4.29
$$Accuracy(j) = \frac{1}{N}\sum_{i=1}^{N}\frac{a_{ij}}{\sum_{h=1}^{d} a_{ih}}$$

4.8.3 Comparison of Methods

A comparison of the accuracy of each method is presented in Table 4.1.

Table 4.1 Accuracy of Compensation Methods. Accuracy as given by Equation 4.29 for four compensation methods, tested on the Bindley Bioscience Center dataset. ULS is Unconstrained Least Squares, OSP is Orthogonal Subspace Projection, FCON is Fully Constrained One Norm, and FCLS is Fully Constrained Least Squares. Each column corresponds to a control sample for a target dye.

Method	CD4-PE	CD45-ECD	CD45-FITC	CD45-PC5	CD45-PC7	Average	Average Running Time (s)
ULS	0.7159	0.9652	0.9805	0.9757	0.9714	0.92174	0.0033
OSP	0.7156	0.9484	0.9815	0.977	0.9785	0.9202	0.0043
FCON	0.7382	0.9827	0.9954	0.9941	0.9947	0.94102	104.6012
FCLS	0.7444	0.9849	0.9955	0.9949	0.9945	0.94284	22.9282

OSP and ULS had the shortest running times, but some gain in accuracy was made by using the optimization methods. Overall the best performance was by the fully constrained least squares method. The fully constrained methods have an advantage in that they return proportional abundance values which may be multiplied by the overall fluorescence output from a cell, preserving the relative intensities of the cells and making outlier detection easier. In a real-time situation, however, unconstrained least squares may be useful if one is willing to sacrifice accuracy for a shorter run time. It may be observed from the results that certain markers such as CD4-PE have a wider spectral response, meaning their colors overlap with other dyes more, and are therefore more difficult to quantify.

CHAPTER 5 CELL TYPE CLASSIFICATION

The way a cell is classified as a particular cell type is based on the amount of particular fluorescent antibodies which have attached to the cell. Often these amounts are simply defined as positive (+) or negative (-) amounts of a dye. For example, a T-cell would be described as CD45+, CD3+, CD4+, where CD45, CD3, and CD4 are fluorescent antibodies with which the blood sample was stained.

5.1 Current Standards

The standard practice in analyzing FCM data is to manually place decision boundaries on 2-dimensional or 1-dimensional plots of the data [5,9,19,21,34]. This method is used for the identification of lymphocytes based on forward and side scatter as well as the identification of lymphocyte subtypes based on fluorescent emission from antibodies.

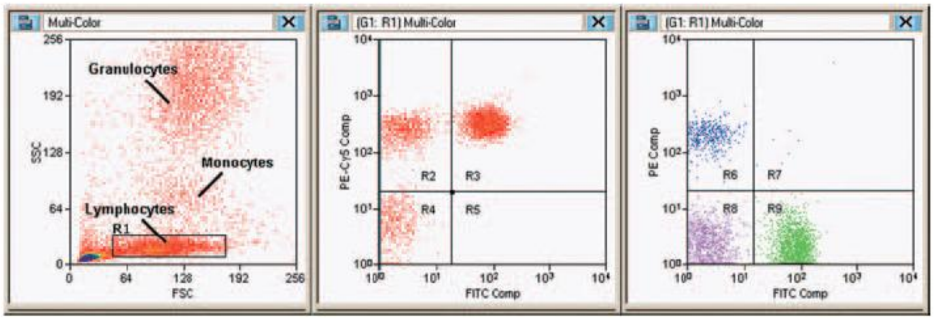

Figure 5.1 Example of the Gating Process [34]. The first plot shows FCM data plotted on forward scatter on the x-axis and side scatter on the y-axis. The three populations identified are lymphocytes, monocytes and granulocytes. The second and third plots show the FCM data plotted on two fluorescent dye expressions at a time. The horizontal and vertical lines are the "gates" manually placed by an analyst. These types of plots are used in most FCM data analysis software.

The basic principle is to find a separating boundary to distinguish two populations for each dye, those cells with a low expression of the dye and those with a high expression of the dye. The cells with low expression are considered as negative for that dye, and cells with a high expression are considered as positive for that dye. Histograms of the expression of each individual dye may be used to refine the decision boundaries placed in two dimensions. This process is commonly referred to as “gating”. Once these decision boundaries are set, cells are labeled as a subtype based on their membership in positive and negative groups for expressions of each dye. For example, if a cell is ‘positive’ for CD45, CD3, and CD4, then it may be labeled as a T-cell.

This method is highly subjective and time-consuming since it is performed manually [5,9,13,19,21,35]. The boundary placement is based on visual interpretation of density of data points as well as experience from analyzing many FCM datasets. In addition, the data is often transformed before it is gated to make the clusters appear more circular. This is because cell populations are often not symmetrically distributed [24,25]. Also, after unconstrained compensation by the method discussed in section 2.B, some cell groups have low values and some cells usually end up having negative abundance values. Several different transformations are used. Logarithmic and “Logicle” transformations are the most common. “Logicle” transformation is linear up to some value and then logarithmic for all greater values [15,24]. These transformations often result in a mistreatment of negative values and a large amount of cells appearing to have abundance values close to zero. The entire process is extremely heuristic and based on the preferences of the human analyst.

5.2 Objectives of Automated Classification

An automated method is needed for phenotype classification for two major reasons. The first is that manual analysis of FCM data takes an extremely long amount of time. The amount of time dedicated to manual FCM data analysis has been increasing significantly as hardware advances in flow cytometers have enabled cells to be analyzed faster and with increasing numbers of photodetectors [5,12,21]. For a single blood sample, millions of cells may be analyzed with 20 to 30 photodetectors. This indicates an increased number of fluorescent antibodies with which the cells are stained, also increasing the number of phenotypes which may be identified with these antibodies.

The second major motivation for the development of automated classification methods for FCM data is the large amount of subjectivity involved in manual gating. As mentioned previously, the transformations applied to make the data more visually appealing may result in data loss, as negative and very small values may appear as zero values. Then visual inspection by each human analyst may be different, based on his or her experience and preferences. An automated method will use the same criteria objectively for every dataset it is applied to. It will also not require any misleading transformations because visualization will not be the basis of classification.

The following methods seek to classify cells into phenotypes based on the amount of fluorescent antibodies present in the cells. The methods discussed analyze each blood sample individually. Some methods look for specific cell types expected to appear in the blood sample based on training data or by using the definition of specific cell types with respect to “positive” or “negative” expressions of antibodies. Other methods only seek to identify all of the cell type populations present within the sample. An advantage of these unsupervised methods is that they may more easily identify abnormal populations which arise from disease. Also, they are not affected by biological variation and variation due to hardware calibration as a supervised learning method may be.

5.3 Literature Review

Supervised approaches such as Support Vector Machines and Neural Networks have been applied to FCM data, but there have been nearly twice the amount of studies done using unsupervised methods [5]. K-means was first applied to FCM data 20 years ago, and clustering has since been popularly used [21]. FlowMeans [1] uses k-means to cluster the data and follows it with iterative merging based on minimum Euclidean distance. [2] uses Expectation Maximization to model the data as a mixture of Gaussians. Subsequent investigators noted that cell populations tend to be non-symmetric, highly skewed, noisy and contain outliers [25]. FlowClust [19] uses a Box-Cox transform (modified to handle negative data) to reduce skew and then uses EM to fit t-mixture models to the data. FlowMerge [9] uses the same technique, but extends it by merging clusters after applying EM. FLAME [25,31] uses EM to fit skew t-distribution models and attempts to create metaclusters across all datasets to relate FCM datasets containing offset. SamSpectral [35] clusters the data into small groups using a distance threshold and then treats each group as a single point. The adjacency matrix, which is used for spectral clustering, contains the summed distances between all points for each pair of groups. This method still requires the calculation of the distance matrix for the entire dataset. Probability binning is used by [28], involving a k-nearest-neighbors analysis followed by computation of a statistic comparing a control dataset and test datasets.

5.4 Clustering

Given that the data is variant between experiments because of variation in fluorescent antibody suppliers, staining procedures and calibration of the flow cytometer [25], clustering has been the most popular approach to solving the FCM data classification problem because of its independence from offset [21]. Clustering methods are also useful for detecting unexpected cell types which may have biological significance. Spectral Clustering and Expectation Maximization using a gaussian mixture model are discussed and compared.

5.4.1 Spectral Clustering

In spectral clustering, clustering is viewed as an optimal graph cut of a similarity graph which describes the relationships between samples [22]. Spectral clustering classifies data based on the density of data points, and is therefore able to distinguish clusters of arbitrary shape as long as they are compact. The basic process of the algorithm is given in Table 5.1.

Table 5.1. Spectral Clustering Algorithm

1. Compute W, a similarity matrix where W$_{ij}$ is the similarity between point *i* and point *j*.
2. Compute D, a diagonal matrix where the values on the diagonal are the sum of the weights of all edges incident to each point.

$$\mathbf{D}_{ii} = \sum_{j=1}^{N} \mathbf{W}_{ij}$$

3. Compute L, the normalized symmetric graph laplacian, defined as:

$$\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$$

4. Compute the first *k* eigenvectors and eigenvalues of L, where *k* is the number of clusters.
5. Let U be the matrix of the eigenvectors of L, where each column of U is an eigenvector. Normalize U. Cluster the rows of U as if they were data points using k-means.
6. The class assigned to each *ith* row corresponds to the class assigned to the *ith* sample.

This algorithm is outlined in [22]. The data is represented by a completely connected similarity graph W wherein the vertices are the sample points and the edges are the similarity between the two points that are connected by the edge. Kernel functions are used to compute the similarity between pair of samples. The diagonal of W is set to 0.

5.4.1.1 Estimating Parameters

In this implementation of spectral clustering, four different kernels were tested to calculate the similarity matrix in independent experiments. The kernels implemented are gaussian, cauchy, log power, and generalized *t* [30]. The log power and generalized *t* are too complicated for use with the large sized FCM datasets. The cauchy kernel resulted in a lower accuracy than the gaussian kernel. Therefore the gaussian kernel was used in the comparison of spectral clustering to the other unsupervised methods.

Equation 5.1 Gaussian kernel:

$$k(x_i, x_j) = \exp\left(-\frac{\left\| x_i - x_j \right\|^2}{2\sigma^2} \right)$$

The Gaussian kernel requires the use of a tuning parameter, sigma, which controls the spread of the kernel. This value was found experimentally by iteratively running spectral clustering on a single file from a dataset with different values of sigma. The sigma value resulting in a similarity matrix whose average similarity value was closest to 0.5 was chosen. The reasoning is that the value of the Gaussian kernel ranges from 0 to 1, so the average similarity should be close to the mean value of 0 and 1.

5.4.1.2 Estimating the number of clusters

The number *k* of clusters also had to be estimated. In step 4 of the spectral clustering algorithm described above, a large number of eigenvalues and eigenvectors were calculated, and the value *k* was chosen from these.

A technique mentioned in [22] was used, wherein the eigenvalues of the normalized Laplacian matrix were searched to find the largest difference between two adjacent eigenvalues when they were in sorted descending order. The location of this largest gap was used as the number of clusters. Then only the first *k* eigenvectors were clustered. A similar approach is used in SamSpectral, a spectral clustering method developed for FCM data analysis [35].

5.4.1.3 Implementation

Spectral Clustering was implemented in Matlab. The algorithm for Spectral Clustering requires the calculation of a distance matrix, which is N by N, requiring a matrix with N^2 elements. The algorithm used in SamSpectral requires the distance matrix to be calculated for the entire dataset before it is clustered into small groups.

In order to deal with memory limitations, the data was randomly permuted and then partitioned into subsets. Spectral clustering was run on each subset, and then the resulting labels were numerically matched. Labels were matched by calculating the mean of each cluster and matching the labels with the closest means. Specifically, each subset's class means were matched to an overall mean of the means of the classes that had already been matched.

To confirm that this was a valid approach, a range of different sampling intervals was tested, and the accuracy for each subset was calculated in order to observe any significant effects by subsampling the data. The size of the subset was increased until a memory error occurred. The standard deviation of the accuracy was only about 0.005, while there was a significant increase in running time when the sample size was increased from 14,210 to 17,051.

Table 5.2 Spectral Clustering on Random Subsets. Accuracy and running time of spectral clustering using a gaussian kernel applied to random subsets of a data file. Total number of samples: 85,255

Subset size	Accuracy	Running time (seconds)
5684	0.8619	0.011
6090	0.8576	0.012
6559	0.8672	0.0135
7105	0.8549	0.0162
7751	0.8543	0.0193
8526	0.8488	0.0234
9473	0.8591	0.0288
10657	0.8512	0.0433
12180	0.8596	0.0487
14210	0.8613	0.0695
17051	0.8546	2.2397

5.4.2 Expectation Maximization (Gaussian)

Expectation Maximization (EM) is an iterative process used to calculate maximum likelihood estimates of distribution parameters for a mixture of distributions [8]. It seeks to maximize the observed complete-data likelihood function by iteratively updating class memberships and maximum likelihood estimators for each distribution involved in the mixture model. The parameter estimates are derived from the complete-data log-likelihood by differentiating it with respect to each parameter and setting it to 0. In this case, cells are assumed to come from a Gaussian mixture model with *c* components. The process involves the 3 repeating steps described in Table 5.3. The algorithm was implemented in Matlab.

Table 5.3 Expectation Maximization Algorithm. EM for a gaussian mixture model.

1. Expectation: Calculate the posterior probabilities for each sample for each class:

$$z_{ik} = \frac{p(C_k \mid x_i, \mu_k, \Sigma_k)}{\sum_{j=1}^{K} p(C_j \mid x_i, \mu_j, \Sigma_j)} = \frac{\pi_k \times f(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} [\pi_j \times f(x_i \mid \mu_j, \Sigma_j)]}$$

   - $f(x_i \mid \mu_k, \Sigma_k)$ is the pdf of a Normal Distribution. μ_k is the estimated mean of cluster *k* and Σ_k is the estimated covariance of cluster *k*. π_k is the prior probability of cluster *k*. *K* is the total number of clusters in the mixture model.

2. Maximization: Calculate the weighted maximum likelihood estimates for the parameters for each class *k* using the posterior probabilities calculated in the previous step.

   - prior probability: $$\pi_k = \frac{1}{N} \sum_{i=1}^{N} z_{ik}$$
   - mean: $$\mu_k = \frac{\sum_{i=1}^{N} z_{ik} x_i}{\sum_{i=1}^{N} z_{ik}}$$
   - covariance: $$\Sigma_k = \frac{\sum_{i=1}^{N} z_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{N} z_{ik}}$$
   - where *N* is the total number of data points.

Table 5.3 Continued. Expectation Maximization Algorithm. EM for a gaussian mixture model.

3. Calculate the complete-data log likelihood.

$$L(X \mid \theta) = \sum_{i=1}^{N} \ln \sum_{j=1}^{K} z_{ik}$$

- where θ is the set of all cluster parameters and X is the entire set of observed data.

The condition for termination is when the change in *L*, the complete-data log likelihood is less than a user-defined threshold from one iteration to the next. In other words, each iteration should increase *L*, since that is the function we would like to maximize by the entire process. The threshold parameter for the change in *L* for each iteration was set at 0.0001.

The parameter values are initialized by performing k-means on the data and calculating the sample means and covariances of the resulting clusters. These parameters are used as the initial distribution parameters to perform the first expectation step.

5.4.2.1 Estimating the Number of Clusters

K, the number of clusters, was estimated experimentally. A range of reasonable values for *K* was chosen, and EM was applied to the data using each value of *K*. Bayesian Information Criterion (BIC) was used to evaluate the performance of the classifier in each case.

Equation 5.2 Bayesian Information Criterion: $BIC = -2\ln L + |\theta| \ln(N)$

$|\theta|$ is the number of parameters being estimated by the algorithm. The value of *K* that minimized the BIC was chosen.

5.4.2.2 Cluster Merging

The value of *K* used to model the FCM data is a much higher number than the number of actual cell phenotypes present in the data because it takes several gaussians to model the asymmetric shape of the class distributions. Some researchers have performed transformations on the data before applying EM or K-means [9,20]. Performing a transformation the data before clustering would make clusters appear more symmetric and therefore more easily modeled by symmetric distributions, but could potentially result in misinterpretation of the data or loss of data with low abundance values.

In this experiment the resulting clusters from EM were iteratively merged based on minimum Bhattacharya distance. A similar approach is taken by FlowMerge [9], FlowMeans [1], and SamSpectral [35]. Each of these methods iteratively merge clusters based on some criterion until it is not possible to merge anymore. Then the number of clusters at each iteration is plotted against the corresponding values of the criterion function. At each point in the plot, a line is fitted to all of the points on each side of the current point. The optimal number of clusters is chosen at the point where there is least square error between the two fitted lines and the points they are fitted to.

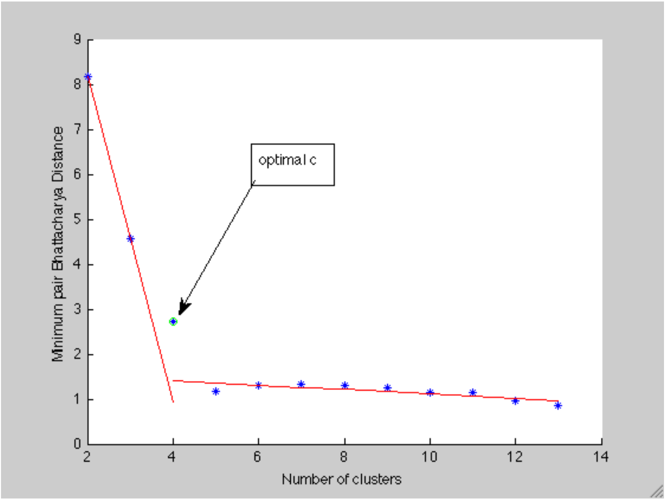

Figure 5.2 Estimating the Number of Clusters to Merge. At each iteration the number of remaining clusters is plotted against the minimum pairwise Bhattacharya distance. Two lines are fitted to the points on either side of a testing point to find which partition results in the least squared error between the lines and the points they are fitted to.

In this experiment, the Bhattacharya distance was calculated between all pairs of classes. The Bhattacharya distance between two classes C_i and C_j is defined below.

Equation 5.3 $$Bhat(C_i,C_j) = \frac{1}{8}(\mu_i - \mu_j)^T \Sigma_P^{-1}(\mu_i - \mu_j) + \frac{1}{2}\ln\left(\frac{\det \Sigma_P}{\sqrt{\det \Sigma_i \det \Sigma_j}}\right)$$

where μ_i is the mean and Σ_i is the covariance of cluster C_i and

Equation 5.4 $$\Sigma_P = \frac{\Sigma_i + \Sigma_j}{2}$$

The pair of clusters with the minimum Bhattacharya distance was merged into one class, and the class parameters were updated to represent all samples now in that merged class. This process was repeated until one cluster remained. The number of remaining clusters was plotted against the distance between the pair that was merged in each iteration. Then least squares regression was used to fit two disjoint lines separated at each point along the curve in order to find an optimal break point.

5.5 Supervised Methods

Supervised methods have the advantage of making use of expert domain knowledge. Given a set of labeled blood samples, a supervised classifier should be able to identify meaningful populations of cell types. However, when using a supervised classifier, some obstacles must be addressed. If a blood sample contains abnormal cell types because of a disease for example, the classifier should be able to identify this as a new class. In addition, the classifier should be flexible enough to tolerate offset in measured values due to biological and experimental variation and hardware calibration.

In this report, a support vector machine was used to train a classifier using the labeled data. This method was compared to another in which each phenotype was modeled with a gaussian mixture model using maximum likelihood estimates of the labeled data. Over multiple labeled blood samples, the same phenotype population found in each of these files was represented as a single component in the gaussian mixture model for that phenotype.

5.5.1 Support Vector Machine (SVM)

Support Vector Machine is an algorithm that maximizes the margin between two classes, creating a decision boundary based on those points which lie closest to the opposite class. Libsvm [7] was used to train and classify the data. Libsvm uses a one-against-one approach to perform multiclass classification, as was the case with the Flowcap-I dataset. The C-SVM method was used in which the primal optimization problem is solved using the dual formulation. The primal problem is

Equation 5.5 $$\min_{w,b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

subject to

Equation 5.6 $$y_i(\mathbf{w}^T\phi(x_i)+b) \geq 1-\xi_i$$

Equation 5.7 $$\xi_i \geq 0, i = 1,\ldots,l$$

where $\phi(.)$ is a basis function.

This problem maximizes the boundary while allowing some amount of error for data points which overlap between classes. The dual problem is solved in this case is

Equation 5.8 $$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$

subject to

Equation 5.9 $$\mathbf{y}^T\alpha = 0$$

Equation 5.10 $$0 \leq \alpha_i \leq C, \quad i = 1,\ldots,l$$

where *e* is a vector of ones and *Q* is a matrix with elements

Equation 5.11 $$Q_{ij} = y_i y_j K(x_i, x_j)$$

K is the kernel function. Polynomial kernels of various degrees were tested on a subset of the data, and in the end a linear kernel was used because it gave the highest accuracy. Two kernel-independent user-defined parameters are used in this, *C* and *e*. *C* is the trade-off between training error and the margin and *e* is the error allowed for termination [7].

Due to the long training time of SVM for the large FCM datasets, subsampling was used to train the model. For each file in the training fold, a percentage of samples from each class were selected and used in the training set. Testing was performed on the entire test fold.

Table 5.4 Accuracy of SVM on 5% Subsets of CFSE Dataset. 5% of samples from each class were chosen from each file in the training set for the CFSE dataset. 5-fold cross-validation was used.

Fold	Misclassification Accuracy	Running Time (min)
1	0.7875	28.4967
2	0.9106	13.6417
3	0.9645	20.2067
4	0.9914	24.7900
5	0.9686	30.8917
Average Result	0.9245	23.6053

5.5.2 Gaussian Mixture Model

In this approach, each phenotype C_p was modeled by a Gaussian mixture model with *K* components, where *K* is equal to the number of files used for training.

Equation 5.12

$$p(x \mid C_p) = \sum_{k=1}^{K} \pi_k N(x \mid \mu_k, \Sigma_k)$$

For a single cell type *p*, the population of that type within each separate file was modeled as a single Gaussian, and the mixture of all of these Gaussians was the overall model for that cell type. For a new cell x_i, its posterior probability z_{ikp} was calculated for each component *k* over all mixture models.

Equation 5.13
$$p(z_{ikp} \mid x_i) = \frac{p(z_{ikp})p(x_i \mid z_{ikp}, \mu_{kp}, \Sigma_{kp})}{p(x_i \mid C_p)}$$

Then the component for which the cell had a maximum posterior probability was used to predict the phenotype of the cell.

Equation 5.14
$$x_i \in C_p, \max_{k,p} z_{ikp}$$

C_p is the p^{th} mixture model and z_{ikp} is the posterior probability of x_i belonging to the k^{th} component of the p^{th} mixture model. This approach takes into consideration the offset in measurement between different data files by treating a cell type in each file as a separate component. By considering the entire phenotype model as a mixture of these components, the observed variation is explicitly modeled. This algorithm was implemented in Matlab.

5.6 Results

Several implementation issues were encountered because of the extremely large size of some of the datasets. Certain methods such as spectral clustering and SVM had to be implemented using a subsampling scheme, as described in previous sections, to make it possible to run these algorithms within a reasonable time frame. Table 5.5 shows the average number of samples and dimensionality within each file in each of the Flowcap-I datasets.

Table 5.5 Size of FlowCap-I Datasets. Average number of samples and dimensionality in each file in the FlowCap-I datasets.

Dataset	Average Number of Samples	Dimensionality
CFSE	92767	8
GvHD	14389	6
Lymph	8976	5
NDD	59490	12
StemCell	9766	6

For each dataset, k-fold cross validation was performed to compare the supervised methods. In practical applications, one file is analyzed at a time, so each fold was a group of files. The order of the files was randomly permuted once and then separated into k groups. In an iterative process, one fold was chosen to be the test data and the rest were used as training data. The average accuracy was compared between all methods based on the ground truth labels provided by trained experts. Accuracy is simply defined as one minus the proportion of labels that were incorrect. A Naive Bayes classifier was compared to the supervised methods as a baseline.

Table 5.6 shows the average results of the supervised methods over all Flowcap-I datasets. The results for each individual dataset vary and are displayed in the Appendix.

Table 5.6 Results of Supervised Methods. Average over all datasets.

Method	Average Accuracy
Naive Bayes	0.8870
Gaussian Mixture Model	0.8843
SVM	0.8589

To compare the unsupervised methods described in this paper, each individual data file was clustered and the average classification accuracies over all files in the dataset were compared. Table 5.7 shows the average results over all datasets, and the results for individual datasets are shown in the Appendix.

Table 5.7 Results of Unsupervised Methods. Average over all datasets.

Method	Average Accuracy
EM Gaussian with merging	0.8714
Spectral Clustering	0.8058

Among the supervised methods, the Gaussian mixture model and the Naive Bayes classifiers gave comparable results in terms of accuracy and running time. For certain datasets the gaussian mixture model performed better, while in others the Naive Bayes classifier performed better. The two methods are very similar. Calculating the estimated distribution parameters for labeled data is a trivial procedure. Naive Bayes treats the all samples from a class as a single distribution while the gaussian mixture model keeps the populations from each data file separate. The same phenotype present in each blood sample may be offset by some amount, but the gaussian mixture model approach models each offset population and assumes that the variation between data files is relatively consistent. Given enough training files, the distribution of a phenotype is likely to be close to the distribution of that phenotype in another file.

Among the unsupervised methods, Expectation Maximization using a single Gaussian mixture model with cluster merging was the faster algorithm and resulted in a better misclassification accuracy than spectral clustering. Spectral Clustering had a long running time because of the calculation of the eigenspace for the very large distance matrix. The requirement of a distance matrix also causes a lot of memory problems, and is impractical for use on FCM data unless subsampling or partitioning is used since the datasets tend to be very large.

Two other unsupervised methods, EM using a skew *t* mixture model [25,31,33], and a Dirichlet Process Mixture Model, were considered. EM using a skew *t* mixture model was considered because it would potentially be better at fitting each of the asymmetric populations with a single distribution [25,33]. It was run on a single file from the CFSE dataset containing about 92,000 data points and did not complete after three days. The Dirichlet Process (DP) Mixture Model was considered because it would be able to automatically infer the number of clusters needed to model the data, as opposed to EM where the number of clusters had to be estimated experimentally.

The DP mixture model was run on the GvHD dataset, where each file contains on average 14,389 data points. The accuracy was comparable to EM using a gaussian mixture model, but the running time for each file was on average 46 minutes. Therefore this method was deemed impractical because a human expert would probably be able to analyze this data in less time. In addition, the DP mixture model generated about 10 clusters, which is about the same as EM, so the same post-processing method of cluster merging had to be performed.

Table 5.8 Comparison of EM and DP. Accuracy and running time of EM and DP mixture models on the GvHD dataset. Note that the running time for EM includes the time to experimentally estimate the number of clusters.

Method	Average Accuracy	Average Running Time
EM Gaussian	0.9182	3.6162
DP Mixture Model	0.9291	2764.1000

CHAPTER 6 CONCLUSIONS

Automated methods for FCM data analysis are still being developed, and although several methods exist, it is difficult to generalize them to different experiments. The Equipment and markers used, the type of biological samples being analyzed, and variations in procedures between different labs create datasets that are diverse. The procedure of gating on FSC and SSC can be difficult because of the lack of separation between certain groups as well as the presence of different objects, depending on the sample being analyzed. Much of the process of staining cells with fluorescent markers can result in variation in the fluorescence features of FCM data. The dyes may come from different suppliers or the lab may have slightly different procedures for staining samples.

Generative models seem to work better than discriminative models for FCM data. This may be because probabilistic classifiers are able to incorporate the uncertainty resulting from this variation between data files. This is illustrated by the results of supervised classifiers in the Appendix. SVM worked well on the NDD dataset because the samples all came from normal human donors, where the cell type populations remain relatively stable between donors. However, SVM performed relatively poorly on the CFSE dataset and the StemCell dataset because these blood samples were from patients with a virus or cancer, and therefore the blood samples were abnormal and cell populations varied significantly.

Sometimes compensation is conducted automatically in the hardware, and sometimes it is done in software. Compensation resulting in negative values is often followed by the removal of cells with negative fluorescence values, resulting in data loss. By using a fully constrained compensation method such as the fully constrained least squares discussed earlier, compensation can result in informative marker abundances with no data loss.

Since datasets tend to be diverse and often experiments are designed to identify and quantify particular cell groups, supervised methods seem to be the most useful. Even though an unsupervised method may identify a rare or abnormal population, the same would be possible using a supervised method with outlier detection, in the simplest case using a probability threshold. Also, methods that require a long running time or a large amount of memory are impractical for FCM data, due to the often very large size of an FCM data file.

An important use of flow cytometers is in cell sorting, wherein each cell is sent to a different container after it is measured. Emerging areas such as stem cell research make use of cell sorting. Currently cell sorters depend on user-defined thresholds to classify cells and sort them. For automated methods to be useful in this area, online learning is necessary. Given a proper training set including control samples and a few labeled samples, online classification may be successful using some of the techniques described in this paper. Fully constrained least squares compensation followed by the calculation of the posterior probability of trained cell types would allow each cell to be identified immediately after passing through the photodetectors in a flow cytometer. The creation of a user-defined threshold for cell sorting would require manual analysis of some data beforehand as well, so there would not be a significant change in the amount of manual labor to be performed, but the results would be more consistent and objective. Also, calculating probabilities is better for handling offset between biological samples than a hard threshold. Unsupervised methods will never be useful for cell sorting because they require an entire dataset or some subpopulation to perform classification. However, they may be used to replace manual analysis of an entire dataset, which could then be used for training a classifier for online learning. If training data were not provided, a human operator would still have to identify the phenotype of each cluster resulting from the application of an unsupervised method.

REFERENCES

REFERENCES

1. Aghaeepour, N., R. Nikolic, H. H. Hoos, and R. Brinkman. 2011. Rapid cell population identification in flow cytometry data. Cytometry Part A. 79A:6-13.

2. Boedigheimer, M. J., and J. Ferbas. 2008. Mixture modeling approach to flow cytometry data. Cytometry A. 73(5):421-429.

3. Bagwell, C. B., and E. G. Adams.1993. Fluorescence Spectral Overlap Compensation for Any Number of Flow Cytometry Parameters. Ann N Y Acad Sci. 677:167-184.

4. Barlogie, B., M. Raber, J. Schumann, T. Johnson, B. Drewinko, D. Swartzendruber, W. Gohde, M. Andreeff, and E. Freireich. 1983. Flow Cytometry in Clinical Cancer Research. Cancer Res. 43:3982.

5. Bashashati, A., and R. Brinkman. 2009. A Survey of Flow Cytometry Data Analysis Methods. Adv Bioinformatics. 584603.

6. Chang, C. 2005. Orthogonal subspace projection (OSP) revisited: a comprehensive study and analysis. IEEE Transactions on Geoscience and Remote Sensing. 43(3):502-518.

7. Chang, C. C., and C. J. Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1-27:27, 2011. Retrieved April 1, 2012 from http://www.csie.ntu.edu.tw/~cjlin/libsvm

8. Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the *EM* Algorithm. Journal of the Royal Statistical Society, Series B. 39(1):1-38

9. Finak, G., A. Bashashati, R. Brinkman, and R. Gottardo. 2009. Merging mixture components for cell population identification in flow cytometry. Adv Bioinformatics. 247646.

10. FlowCap-I Summit. FlowCAP - Flow Cytometry: Critical Assessment of Population Identification Methods. Retrieved April 12, 2012 from http://flowcap.flowsite.org/

11. Garini, Y., I. Young, and G. McNamara. 2006. Spectral Imaging: Principles and Applications. Cytometry Part A. 69(8):735-747.

12. Gosink, J., G. D. Means, W. A. Rees, C. Su, and H. A. Rand. 2009. Bridging the Divide between Manual Gating and Bioinformatics with the Bioconductor Package flowFlowJo. Adv Bioinformatics. 809469.

13. Gottardo, R., R. Brinkman, G. Luta, and M. P. Wand. 2009. Recent bioinformatics advances in the analysis of high throughput flow cytometry data. Adv Bioinformatics. 461763

14. Heinz, D. C., and C. Chang. 2001. Fully Constrained Least Squares Linear Spectral Mixture Analysis Method for Material Quantification in Hyperspectral Imagery. IEEE Transactions on Geosciences and Remote Sensing. 39(3):529-545.

15. Herzenberg, L. A., J. Tung, W. A. Moore, and D. R. Parks. 2006. Interpreting flow cytometry data: a guide for the perplexed. Nat Immunol. 7(7):681-685.

16. Hoffman, R. A., P. C. Kung, W. P. Hansen, and G. Goldstein. 1980. Simple and rapid measurement of human T lymphocytes and their subclasses in peripheral blood. PNAS. 77(8):4914-4917.

17. Keenan, M., J. A. Timlin, M. Van Benthem, and D. Haaland. 2002. Algorithms for constrained linear unmixing. Imaging Spectrometry VIII. 4816:193-202.

18. Keshava, N., and J. Mustard. 2002. Spectral Unmixing. IEEE Signal Processing Magazine. 19(1):44-57.

19. Lo, K., R. Brinkman, and R. Gottardo. 2008. Automated gating of flow cytometry data via robust model-based clustering. Cytometry Part A. 73A:321-332.

20. Loken, M., J. Brosnan, B. Bach, and K. Ault. 2005. Establishing Optimal Lymphocyte Gates for Immunophenotyping by Flow Cytometry. Cytometry Part A. 11(4):453-459.

21. Lugli, E., M. Roederer, and A. Cossarizza. 2010. Data analysis in flow cytometry: The future just started. Cytometry Part A. 77A:705-713.

22. Luxburg, U. 2007. A tutorial on spectral clustering. Statistics and Computing. 17(4): 395-416.

23. Marie, D., F. Partensky, S. Jacquet, and D. Vaulot. 1997. Enumeration and Cell Cycle Analysis of Natural Populations of Marine Picoplankton by Flow Cytometry Using the Nucleic Acid Stain SYBR Green I. Appl Environ Microbiol. 63(1):186-193.

24. Parks, D. R., M. Roederer, and W. A. Moore. 2006. A New “Logicle” Display Method Avoids Deceptive Effects of Logarithmic Scaling For Low Signals and Compensated Data. Cytometry Part A. 69(6):541-551.

25. Pyne, S., X. Hu, K. Wang, E. Rossin, T. Lin, L. Maier, C. Baecher-Allan, G. McLachlan, P. Tamayo, and D. Hafler. 2009. Automated high-dimensional flow cytometric data analysis. Proceedings of the National Academy of Sciences. 106:8519.

26. Rahman, M. 2006. Introduction to Flow Cytometry. Oxford, UK: Serotec Ltd.

27. Roederer, M. 2001. Spectral Compensation for Flow Cytometry: Visualization Artifacts, Limitations, and Caveats. Cytometry. 45:194-205.

28. Roederer, M., W. Moore, A. Treister, R. R. Hardy, and L. A. Herzenberg. 2001. Probability binning comparison: a metric for quantifying multivariate distribution differences. Cytometry. 45(1):47-55.

29. Rogers, W., and H. Holyst. 2009. FlowFP: A Bioconductor Package for Fingerprinting Flow Cytometric Data. Adv Bioinformatics. 193947.

30. Souza, C. 2010. Kernel Functions for Machine Learning Applications. Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. Retrieved August 1, 2011 from http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html

31. Sujit, K. S., D. K. Dey, and M. D. Branco. 2003. A New Class of Multivariate Skew Distributions with Applications to Bayesian Regression Models. The Canadian Journal of Statistics / La Revue Canadienne de Statistique. 31(2):129-150.

32. Tung, J. W., D. R. Parks, W. A. Moore, and L. A. Herzenberg. 2004. New approaches to fluorescence compensation and visualization of FACS data. Clin Immunol. 110(3):277-283.

33. Wang, K., S. H. Ng, and G. McLachlan. 2009. Multivariate skew t mixture models: Applications to fluorescence-activated cell sorting data. In: Hao Shi, Yanchun Zhang, Murk J. Bottema, Anthony J. Maeder and Brian C. Lovell, Proceedings of Digital Image Computing: Techniques and Applications. DICTA 2009. 2009 Conference of Digital Image Computing: Techniques and Applications, Melbourne, Australia, (526-531). 1-3 December 2009.

34. Wulff, S. (ed). Flow Cytometry Educational Guide, 2nd Ed. 2006. Dako: Carpinteria, CA.

35. Zare, H., P. Shooshtari, A. Gupta, and R. Brinkman. 2010. Data reduction for spectral clustering to analyze high throughput flow cytometry data. BMC Bioinformatics. 11:403.

36. Zeng, Q. T., J. Pratt, J. Pak, D. Ravnic, H. Huss, and S. Mentzer. 2007. Feature-guided clustering of multi-dimensional flow cytometry datasets. Journal of Biomedical Informatics. 40(3):325-331.

APPENDIX

APPENDIX

Table A.1 Number of Samples in K-Fold Cross-Validation. Number of samples in each training set and testing set when using k-fold cross-validation.

Dataset	Training Set Size	Testing Set Size
CFSE	881066	263551
GvHD	152567	46029
Lymph	239372	56624
NDD	1086860	216711
StemCell	229720	43021

Table A.2 Results of Supervised Methods on CFSE Dataset. 5-fold cross-validation.

Method	Average Accuracy	Average Running Time (s)
Naive Bayes	0.9529	0.7192
Gaussian Mixture Model	0.9601	1.6848
SVM (5% subsets)	0.9245	141600

Table A.3 Results of Supervised Methods on GvHD Dataset. 6-fold cross-validation.

Method	Average Accuracy	Average Running Time (s)
Naive Bayes	0.8956	0.1135
Gaussian Mixture Model	0.8726	0.1352
SVM (25% subsets)	0.8761	7134.5667

Table A.4 Results of Supervised Methods on Lymph Dataset. 6-fold cross-validation.

Method	Average Accuracy	Average Running Time (s)
Naive Bayes	0.8175	0.1404
Gaussian Mixture Model	0.8404	0.3204
SVM (20% subsets)	0.8370	5519.4000

Table A.5 Results of Supervised Methods on NDD Dataset. 6-fold cross-validation.

Method	Average Accuracy	Average Running Time (s)
Naive Bayes	0.9148	1.2699
Gaussian Mixture Model	0.8845	9.9855
SVM (20% subsets)	0.9535	553.8685

Table A.6 Results of Supervised Methods on StemCell Dataset. 6-fold cross-validation.

Method	Average Accuracy	Average Running Time (s)
Naive Bayes	0.8542	0.1316
Gaussian Mixture Model	0.8639	0.3875
SVM (20% subsets)	0.7035	907.0833

Table A.7 Results of Unsupervised Methods on CFSE Dataset.

Method	Average Accuracy	Average Running Time (s)
EM Gaussian with merging	0.9293	6.8528
Spectral Clustering	0.8723	52.1888

Table A.8 Results of Unsupervised Methods on GvHD Dataset.

Method	Average Accuracy	Average Running Time (s)
EM Gaussian with merging	0.9182	3.6162
Spectral Clustering	0.8693	63.5248

Table A.9 Results of Unsupervised Methods on Lymph Dataset.

Method	Average Accuracy	Average Running Time (s)
EM Gaussian with merging	0.8935	5.6855
Spectral Clustering	0.8303	69.2528

Table A.10 Results of Unsupervised Methods on NDD Dataset.

Method	Average Accuracy	Average Running Time (s)
EM Gaussian with merging	0.8926	64.3761
Spectral Clustering	0.8216	1244.6139

Table A.11 Results of Unsupervised Methods on StemCell Dataset.

Method	Average Accuracy	Average Running Time (s)
EM Gaussian with merging	0.7234	8.0864
Spectral Clustering	0.6357	64.6912