DEVELOPMENT OF A UNIFIED ANALYSIS FRAMEWORK FOR MULTICOLOR FLOW CYTOMETRY DATA BASED ON QUASI-SUPERVISED LEARNING

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Electronics and Communication Engineering

by
Başak Esin KÖKTÜRK GÜZEL

July 2017
İZMİR

We approve the thesis of **Başak Esin KÖKTÜRK GÜZEL**

Examining Committee Members:

Prof. Dr. Bilge KARAÇALI
Department of Electrical and Electronics Engineering, İzmir Institute of Technology

Assist. Prof. Dr. Mustafa ÖZUYSAL
Department of Computer Engineering, İzmir Institute of Technology

Assist. Prof. Dr. Mehmet Zübeyir ÜNLÜ
Department of Electrical and Electronics Engineering, İzmir Institute of Technology

Assoc. Prof. Dr. Devrim ÜNAY
Department of Biomedical Engineering, İzmir University of Economics

Assoc. Prof. Dr. Mustafa Alper SELVER
Department of Electrical and Electronics Engineering, Dokuz Eylül University

13 July 2017

Prof. Dr. Bilge KARAÇALI
Supervisor, Department of Electrical and Electronics Engineering
İzmir Institute of Technology

Prof. Dr. Enver TATLICIOĞLU
Head of the Department of
Electrical and Electronics Engineering

Prof. Dr. Aysun SOFUOĞLU
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

PhD has been a truly life-changing experience for me and I could not achieve to success without help of many people.

I would like to express my special appreciation and thanks to my advisor Professor Dr. Bilge Karaçalı. I am grateful him because his contribution on both my academic research and my life have been priceless. I believe that, I will need his guidance during my life.

Besides my advisor, I would like to thank the rest of my thesis progress committee: Assist. Prof. Dr. Mustafa Özuysal and Assist. Prof. Dr. Mehmet Zübeyir Ünlü and also my jury members: Assoc. Prof. Dr. Devrim Ünay, Assoc. Prof. Dr. Alper Selver, Assoc. Prof. Dr. Türker İnce and Assoc. Prof. Dr. Mehmet Engin.

I would also thank to Prof. Dr. Enver Tatlıcıoğlu for his endless supports and advice on my life and academic carrier.

I would especially like to thank Oktay Karakuş and Özgecan Özdoğan who never leave me alone during my PhD life.

A special thanks to my family. I could not express my feelings with words how grateful I am to my mother and father, Hilal and Hüseyin Köktürk. They have been endless support to me.

And finally to my beloved Burç, who is my best friend, soul-mate, and husband. These past several years have not been easy for me both academically and personally. I am grateful him for his endless support, even when I was irritable and depressed.

ABSTRACT

DEVELOPMENT OF A UNIFIED ANALYSIS FRAMEWORK FOR MULTICOLOR FLOW CYTOMETRY DATA BASED ON QUASI-SUPERVISED LEARNING

In this dissertation, automatic compensation and gating strategies are investigated for multi-color flow cytometry data analysis. We propose two clustering algorithms that combine the quasi-supervised learning algorithm with an expectation-maximization routine for automatic gating. The quasi-supervised learning algorithm estimates the posterior probabilities of the different cell populations at each sample in a dataset in a manner that does not involve fitting a parametric model to the data.

We have developed two different binary divisive clustering algorithms based on expectation maximization with responsibility values calculated using the quasi-supervised learning algorithm instead of the probabilistic models used in conventional expectation maximization applications. Our clustering algorithms determine the number of clusters in run-time by measuring the overlap between the estimated clusters in each provisional division and comparing it with the previous one to determine whether the division is warranted or not. Since this type of clustering is indifferent to the underlying distribution of dataset, it is well suited to automatic flow cytometry gating.

The second clustering algorithm improves upon the first one using a simulated annealing approach. Its iterative structure allows finding the global minimum of a cost functional that achieves the best separation point by gradually smoothing the decision regions in each iteration.

Finally, we have developed a joint diagonalization and clustering method for automatic compensation of flow data based on the methods above. The proposed method identifies cell sub groups using the annealing-based model-free expectation-maximization algorithm and finds a data transformation matrix that achieves orthogonality of the covariance structure of each identified cell cluster using fast Frobenius diagonalization.

We have tested all proposed algortihms on both synthetically created datasets and real multi-color flow cytometry datasets. The results show that our automated gating algorithms are very successful in identifying the distinct cell groups so long as there is enough statistical evidence for their presence. In addition, the automated compensation procedure was also successfully applied on the synthetically created dataset and real multi-color flow cytometry data of lymphocytes that are a low autofluorescence cell group. However, the automated compensation algorithm needs further study to be generalized to high autofluorescence cell types where proper compensation does not necessarily coincide with an orthogonal covariance structure.

ÖZET

ÇOK RENKLİ AKIŞ SİTOMETRİSİ VERİLERİ İÇİN YARIGÜDÜMLÜ ÖĞRENME TEMELLİ TÜMLEŞİK BİR BİR ANALİZ PLATFORMU GELİŞTİRİLMESİ

Bu tezde, çok renkli akış sitometri veri analizi için otomatik kompensasyon ve kapılama stratejileri incelenmiştir. Otomatik kapılama için yarı-güdümlü öğrenme algoritmasını ve beklenti en iyileme rutinini birleştirerek iki gruplama algoritması önerilmiştir. Yarı-güdümlü öğrenme algoritması veriye parametrik bir model uydurmadan, her bir örnekteki farklı hücre popülasyonlarının sonsal olasılıklarını tahmin eder.

Sorumluluk değerleri konvansiyonel beklenti en iyileme uygulamalarında kullanılan olasılık modelleri yerine, yarı-güdümlü öğrenme algoritması ile hesaplanarak beklenti en iyilemeye dayalı iki tane ikili kümeleme algoritması geliştirilmiştir. Kümeleme algoritmalarımız, her bir geçici bölünmede tahmini kümeler arasındaki örtüşmeyi ölçerek ve bu örtüşmeyi bir önceki ile karşılaştırarak, bölünmenin doğru olup olmadığı belirler ve böylelikle işleyiş sürecinde küme sayısını belirler. Bu tür kümeleme, veri kümesinin altında yatan dağılıma kayıtsız olduğundan, otomatik akış sitometri kapılaması için uygundur.

İkinci kümeleme algoritması benzetimli tavlama yaklaşımını kullanarak ilk kümeleme algoritmasını geliştirmiştir. Benzetimli tavlama yaklaşımının tekrarlayıcı yapısı bir maaliyet fonksiyonun global minimumunu bulmayı sağlar ve biz bu yaklaşımı karar bölgelerini her tekrarda kademeli olarak yumuşatarak en iyi ayrışma noktasını bulmak için kullandık.

Son olarak, yukarıdaki kapılama yöntemlerine dayalı olarak akış verisinin otomatik olarak kompensasyonu için bir ortak köşegenleştirme ve kümeleme yöntemi geliştirdik. Kompensasyon, farklı florokrom kanalları arasındaki spektral yayılımı gidermek için kullanılan bir prosedürdür. Önerilen yöntem, hücre alt gruplarını tavlama temelli modelden bağımsız beklenti en iyileme algoritması kullanarak tanımlamakta ve tanımlanan her bir hücre kümesinin kovaryans yapısının dikkenliğini, hızlı Frobenius köşegenleştirme yöntemini kullanarak elde eden bir veri dönüşüm matrisi bularak sağlamaktadır.

Önerilen algoritmaları sentetik olarak oluşturulan veri kümeleri ve gerçek çok renkli akış sitometrisi veri kümeleri üzerinde test edilmiştir. Sonuçlar, otomatik kapılama algoritmalarımızın yeterli istatistiksel kanıtı olduğu sürece farklı hücre gruplarını tanımada çok başarılı olduğunu göstermektedir. Buna ek olarak, otomatik kompensasyon prosedürü, başarılı bir şekilde sentetik olarak oluşturulmuş veri setine ve gerçek düşük otofloresanslı lenfosit hücre gruplarına başarıyla uygulanmıştır, ancak, dikgen kovaryans matrisinin geçerli olmadığı yüksek otofloresanslı hücre türlerine genellenebilmesi için daha fazla bir çalışmaya ihtiyaç duyulmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES viii

LIST OF TABLES xi

CHAPTER 1. INTRODUCTION 1

CHAPTER 2. BACKGROUND AND LITERATURE REVIEW 4

2.1. Flow Cytometry 4

2.1.1. Flow Cytometry Experiment........ 4

2.1.2. Data Structure and Analysis 8

2.1.2.1. Data Preprocessing 9

2.1.2.2. Gating 21

2.2. Quasi-supervised Learning Algorithm 24

2.2.1. Analytical Computation of Posterior Probabilities using Quasi-Supervised Learning 25

2.2.2. Class Overlap Measures and the Selection of the Optimal Reference Set Size 29

2.3. Expectation-Maximization Algorithm 34

2.4. Fast Algorithm for Joint Diagonalization with Non-orthogonal Transformations 35

CHAPTER 3. MODEL-FREE EXPECTATION MAXIMIZATION CLUSTERING ... 38

3.1. Methodology 39

3.2. Performance Evaluation........ 43

3.3. Dataset & Results 44

3.3.1. Synthetically Generated Gaussian Mixture Dataset 44

3.3.2. Diffuse Large B-cell lymphoma (DLBCL) Dataset 46

3.3.3. Hematopoietic Stem Cell Transplant (HSCT) Dataset 48

3.4. Discussion 51

CHAPTER 4. ANNEALING-BASED MODEL-FREE EXPECTATION-MAXIMIZATION CLUSTERING 52

4.1. Methodology 52

4.2. Dataset & Results 55
4.2.1. Synthetically Generated Gaussian Mixture Dataset 55
4.2.2. Diffuse Large B-cell lymphoma (DLBCL) Dataset 57
4.2.3. Hematopoietic Stem Cell Transplant (HSCT) Dataset 57
4.3. Discussion 59

CHAPTER 5. AUTOMATIC COMPENSATION OF MULTICOLOR FLOW CYTOMETRY DATA USING JOINT DIAGONALIZATION........ 60
5.1. Methodology 60
5.1.1. Gamma Normalization 61
5.1.2. Joint Clustering and Orthogonalization Algorithm 64
5.2. Results 64
5.2.1. Synthetically Created Dataset 65
5.2.2. Results on the Real Multi-Color Flow Cytometry Dataset 69
5.3. Discussion 85

CHAPTER 6. CONCLUSION 86

REFERENCES 89

LIST OF FIGURES

Figure **Page**

Figure 2.1. A typical flow cytometry experiment 5

Figure 2.2. Jablonski Energy Diagram 6

Figure 2.3. Excitation and Emission Spectrum of Common Dyes in Flow Cytometry Experiment 7

Figure 2.4. Flow Cytometry Parts 8

Figure 2.5. Raw Data Representation: (a)-(c) Histogram plots of first, second and third variate respectively, (d) 3D Scatter plot of raw data 13

Figure 2.6. Log Scale Data Representation: (a)-(c) Histogram plots of first, second and third variate of log-scale data respectively, (d) 3D Scatter plot of log scale data 14

Figure 2.7. Arsinh Scale Data Representation: (a)-(c) Histogram plots of first, second and third variate of inverse hiyperbolic sine data respectively, (d) 3D Scatter plot of inverse hiyperbolic sine scale data 15

Figure 2.8. Logicle Scale Data Representation: (a)-(c) Histogram plots of first, second and third variate of logicle scale data respectively, (d) 3D Scatter plot of logicle scale data 16

Figure 2.9. Box-Cox Data Representation: (a)-(c) Histogram plots of first, second and third variate of Box-Cox transformed data respectively, (d) 3D Scatter plot of Box-Cox transformed data 17

Figure 2.10. Example of FITC spillover into the PE channel 18

Figure 2.11. Diagrammatic representation of a two-signal crossover system 19

Figure 2.12. Diagrammatic representation of a three-signal crossover system 21

Figure 2.13. Gating Example 22

Figure 2.14. Gating strategy using two markers 23

Figure 2.15. Quasi-supervised learning algorithm results for a toy problem. 32

Figure 2.16. True and Estimated Results of Log-likelihood Ratio for Different Sample Size 33

Figure 3.1. Block Diagram of Binary Divisive Clustering using Model-free Expectation-Maximization Algorithm (a). The Model-free Expectation-Maximization Algorithm is detailed in (b). 41

Figure 3.2. Tree Model of the clustering results for the toy problem involving a Gaussian mixture with three components. 42

Figure 3.3. Synthetically Created Gaussian Mixture and Respective Class Labels 45

Figure 3.4. Estimated Clusters using MFEM on Synthetically Created Gaussian Mixture 45
Figure 3.5. Manual Gating Results for The Diffuse Large B-cell Lymphoma (DLBCL) Dataset 47
Figure 3.6. Automated Gating Results for The Diffuse Large B-cell Lymphoma (DLBCL) Dataset Using MFEM 47
Figure 3.7. Automated Gating Results for The Diffuse Large B-cell Lymphoma (DLBCL) Dataset Using Conventional Expectation Maximization 48
Figure 3.8. Manual Gating Results for The Hematopoietic Stem Cell Transplant (HSCT) Dataset 49
Figure 3.9. Automated Gating Results for Hematopoietic Stem Cell Transplant (HSCT) Dataset using MFEM 50
Figure 3.10. Automated Gating Results for Hematopoietic Stem Cell Transplant (HSCT)) Dataset Using Conventional Expectation Maximization 50
Figure 4.1. Block Diagram of the Annealing Based Model-Free Expectation Maximization Algorithm 54
Figure 4.2. The data division is illustrated on the first column and respective $E(n)$ functions are given in the second column. The data is first divided into two clusters (upper row) and division cost is determined. Then, the procedure is applied on the daughter clusters (second and third rows). 56
Figure 4.3. Estimated CLusters for The Diffuse Large B-cell Lymphoma (DLBCL) Dataset using ABMFEM 58
Figure 4.4. Estimated CLusters for The Hematopoietic Stem Cell Transplant (HSCT) Dataset using ABMFEM 58
Figure 5.1. Gamma Normalized Instensity Values for Toy Dataset that Created in Chapter 2 63
Figure 5.2. Block diagram of the proposed method 64
Figure 5.3. Original toy dataset after the exponential transformation. In calculations, this data was used to simulate raw intensity data for compensation and clustering. 66
Figure 5.4. Gamma Normalized Toy Data with The Estimated Cluster Labels 67
Figure 5.5. The Estimated Cluster Labels on Raw Data 67
Figure 5.6. Gamma Normalized Data After Clustering and Joint Diagonalization (first alternative) 68
Figure 5.7. Gamma Normalized Data After Clustering and Joint Diagonalization (second alternative) 68
Figure 5.8. Scatter plots of the most overlapping channels (uncompensated) 71

Figure 5.9. Scatter plots of the most overlapping channels (manually compensated) 71
Figure 5.10. Scatter plots of the most overlapping channels (compensated according to scheme 1) 72
Figure 5.11. Scatter plots of the most overlapping channels (compensated according to sheme 2) 72
Figure 5.12. Scatter plots of the least overlapping channels (uncompensated) 73
Figure 5.13. Scatter plots of the least overlapping channels (manual compensated) 73
Figure 5.14. Scatter plots of the least overlapping channels (compensated according to scheme 1) 74
Figure 5.15. Scatter plots of the least overlapping channels (compensated according to scheme 2) 74
Figure 5.16. Pairwise scatter plots of the uncompensated dataset after gamma normalization. All values are within the unit interval. 75
Figure 5.17. Pairwise scatter plots of the manually compensated dataset 76
Figure 5.18. Pairwise scatter plots of automatically compensated dataset with the identified cell sub-groups. In total, three cell subgroups were identified, indicated by the red, green and blue colors respectively. 77
Figure 5.19. Pairwise scatter plots of the uncompensated LNA31 dataset after gamma normalization. All values are within the unit interval. 79
Figure 5.20. Pairwise scatter plots of the manually compensated LNA31 dataset after gamma normalization. All values are within the unit interval. 80
Figure 5.21. Pairwise scatter plots of automatically compensated LNA31 dataset with the identified cell sub-groups. In total, four cell subgroups were identified, indicated by the red, green, blue and purple colors respectively. 81
Figure 5.22. Pairwise scatter plots of the uncompensated LNA77 dataset after gamma normalization. All values are within the unit interval 82
Figure 5.23. Pairwise scatter plots of the manually compensated LNA77 dataset after gamma normalization. All values are within the unit interval. 83
Figure 5.24. Pairwise scatter plots of automatically compensated LNA77 dataset with the identified cell sub-groups. In total, three cell subgroups were identified, indicated by the red, green and blue colors respectively. 84

LIST OF TABLES

Table **Page**

Table 2.1. Matrix Representation of Flow Cytometry Data 9

Table 2.2. Most Popular Algorithms for Automated Gating of Flow Cytometry Data .. 24

Table 3.1. Contingency Table 43

Table 3.2. F-measures for the algorithm performance on synthetic dataset for different sample sizes in clusters 46

Table 4.1. F-measures for the algorithm performance on synthetic dataset for different sample sizes in clusters 55

CHAPTER 1

INTRODUCTION

Flow cytometry (FCM) provides rapid analysis of structural and chemical properties of individual cells in a heterogeneous population (Brown and Wittwer, 2000). It can be used to sort cells into groups according to their biological properties to which each of them belongs. Determination of cell diversity makes immunology the main area of FCM (Matton, 2014). It is widely used in academic research on the immune system and also used in community hospitals to detect and monitor diseases such as acute leukemia, malignant lymphoma, human immunodeficiency virus (HIV), etc.

The developed technology allows to analyse cells with more than 20 parameters and it can process thousands of cells per second. As a result, flow cytometry experiments produce high-dimensional datasets with a large number of events. Since FCM data is traditionally interpreted by flow cytometer experts manually on two or three-dimensional plots. In a flow cytometry experiment, there are two processes that can be automated: *compensation* and *gating*. Gating is the identification of cell subsets using their physical, chemical and biological characteristics as recorded by the flow cytometry experiment. Conventional gating methods rely on operator-drawn regions on 2D scatter plots and this process is laborious and time-consuming. Pathologists draw line segments on 2D scatter plots to identify cell populations that are of interest for specific diseases. This process is repeated until cell subgroups define homogeneous populations (Lee, 2011).

In the literature, there are several statistical methods to identify the cell populations automatically. For instance, Aghaeepour et. al. developed an automated method for cell subtype identification in high dimensional FCM data based on k-means clustering (Aghaeepour et al., 2013), while Pyne et. al. proposed a skew and heavy-tailed distribution fitting approach (Pyne et al., 2009). The FlowClust algorithm (Lo et al., 2008) aims to fit a t-mixture model to FCM data after the Box-Cox transformation. The FlowClust algorithm was later modified by Finak et.al. by introducing a merging step to avoid unwarranted cluster divisions (Finak et al., 2009). Most of the clustering methods in FCM data analysis applications use one of Bayesian information criteria (BIC), Akaike information criteria (AIC) or entropy to determine the unknown number of distinct clusters. This means that the clustering algorithm is to be run several times for varying number of clusters and the clustering result that achieves the optimal separation according to the criterion of choice is to be taken as the final output.

The second process that can be automated in a multi-color flow cytometry experiment, called compensation, arises from the physics of fluorescence emission. Increasing the number of fluorochromes used in an experiment causes increased spectral spillover, defined as the overlap between two or more fluorochromes' emission spectra. The detectors, in this situation, cannot identify their specific biomarker reliably since several fluorochromes with overlapping emission spectra jointly contribute to the detector readings. Compensation can be performed either during data collection on the flow cytometer following calibration or after data collection in software. The procedure is formalized as a linear algebra problem (Roederer, 2002; Bagwell and Adams, 1993a) since spillover parameters can be measured using control beads. The most important aspect in compensation is the ability to visualize all distinct subpopulations as distinctly as possible from each other. To this end, several research groups have been focusing on automatic compensation and automatic gating of multi-color flow cytometry data (Hahne et al., 2009; Sugár et al., 2011). However, all proposed methods rely on calculating the spillover coefficients using control beads (Roederer, 2001). On the other hand, using control beads for calibration purposes may not be possible or feasible before each experiment.

This dissertation offers an automated flow cytometry data analysis framework that contains two clustering algorithms for gating and one joint diagonalization procedure for compensating multicolor flow cytometry data. The principal contributions are as follows:

- We have developed model-free expectation-maximization clustering algorithm which is a binary hierarchical divisive clustering. This clustering method becomes the foundation of this thesis, because it provides fully automatizes cell subgroup identification without any model fitting. It also deviates significantly from the earlier automated gating algorithms that require model assumptions on the unknown flow data distributions.

- We have also developed another clustering algorithm, called as annealing-based model-free expectation-maximization clustering, that combines simulated annealing with model-free expecation-maximization clustering. It provides better clustering performance by optimizing the quasi-supervised learning reference set size. Both clustering algorithms provides automatic number of cluster determination in a heterogeneous dataset and also do not require any knowledge about data distribution or specific parameters in contrast to the earlier methods.

- We proposed, for the first time, to use gamma normalization for flow cytometry data visualization. Gamma normalization allows calculating the operational parameters automatically from raw intensities to obtain an optimal use of the dynamic range. Compared to the original linear scale, after gamma normalization, all three clusters are placed distinctly and can therefore be identified with relative case using a statistical clustering method of choice. Existing normalization schemes suffer from a lack of structured means of adjusting transfer parameters, while the parameters associated with gamma normalization are easily calculated from the raw intensity values.

- We have developed an automatic compensation procedure for low autofluorescence cell groups like lymphocytes based on orthogonalization of fluorochrome channels within each cluster. This procedure provides an automatic compensation without single-color control samples and better cluster placement that makes automatic quadrant identification enable. Automatic compensation is a poorly studied except in flow literature, this method represents the first effort to this end that does not rely on using control beads for calibration of compensation parameters.

In Chapter 2, we give the background information about flow cytometry and the mathematical techniques used in this thesis such as quasi-supervised learning (QSL), expectation maximization (EM), fast algorithm for joint diagonalization with non-orthogonal transformation (FFDIAG). First, the automated gating algorithm, model-free expectation maximization, is introduced in Chapter 3. The method is a binary hierarchical divisive clustering algorithm with linear decision regions. It automatically identifies the number of clusters and assigns samples to these clusters. We adapted this algorithm to detect cell sub groups in a multi-color flow cytometry data. Automated gating performance using the resulting model-free expectation maximization algorithm is shown on both synthetically created Gaussian mixtures and real multi-color flow cytometry data. After, model free expectation maximization algorithm we have developed annealing based model free expectation maximization that begins with a bigger reference set and decreases reference set size in each iteration. This produced more flexible decision regions and more accurate clustering results. We explain the methodology for annealing based expectation maximization algorithm and discuss on the results on the same datasets in Chapter 4. Chapter 5 explains the automated compensation procedure that simultaneously performs compensation and gating on multi-color flow cytometry data. The joint compensation and gating method is applied on real multi-color flow cytometry data and results are discussed in detail. At the end, in Chapter 6, we summarize all algorithms for flow cytometry data analysis automation and discuss their advantages and disadvantages before we offer concluding remarks.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

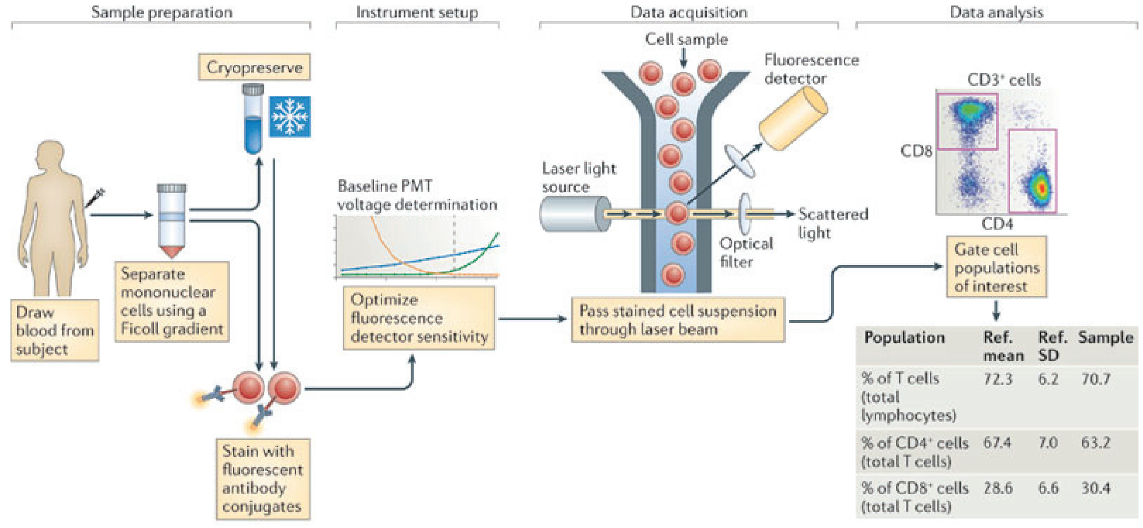
In this chapter, we provide background information about flow cytometry and the mathematical methods that are used in this study: quasi-supervised learning (QSL), expectation-maximization algorithm (EM), Fast Algorithm for Joint Diagonalization with Non-orthogonal Transformations (FFDIAG). The chapter begins with an overview of the literature on automated flow cytometry data analysis.

2.1. Flow Cytometry

Flow cytometry (FCM) is a powerful laser-based cell analysis technique that can characterize the physical, chemical and biological behavior of the individual cell in a heterogeneous population. Cells are labelled using specific fluorochromes and then excited by the laser to emit light at varying wavelengths. The intensity of light emitted by a given fluorochrome indicates the relative abundance of the corresponding biomarker. The main applications of flow cytometry are cell counting, sorting, biomarker detection and protein engineering. Flow cytometry is used in both research applications for distinguishing different cell types and clinical applications to detect the disease, especially blood cancer, and to monitor the disease progress following therapy (Aghaeepour et al., 2013). Flow cytometry technology is widely used in immune system applications since it allows grouping cells according to their biological properties. In flow cytometry experiments, cells are incubated with fluorochrome-conjugated antibodies to identify the biomarkers of interest regarding biological and biochemical behavior of cells. Nowadays, more than 20 biomarkers can be assessed simultaneously in multi-color flow cytometry experiments through the developed technology (O'Donnell et al., 2013). In a single session, a flow cytometer can analyze 10.000 to 1.000.000 cells. In this section, we explain the different stages of a flow cytometry experiment and subsequent data analysis. We describe the working mechanism of a flow cytometer, data pre-processing steps such as data transformations, and compensation and gating procedures of the flow cytometry data, along with a literature review on the automation of compensation and gating procedures.

2.1.1. Flow Cytometry Experiment

A basic flow cytometry experiment is illustrated in Figure 2.1 by Maecker et. al in four different steps. Sample preparation is the first step of the experiment. In this step, blood is obtained from the subject and mononuclear cells are separated; these mononuclear cells are either cryopreserved for later use or stained with aim specific fluorescents. In the second step, instrument setup, detector sensitivities are calibrated using non-stained and single-color-stained cells. In each experiment this step must be performed to obtain robust results. The third step is data acquisition; the stained mononuclear cells are passed through a tube and each cell collides with the laser beam at interrogation point. The light scatters according to physical, chemical and structural properties of the cell. The forward and side detectors capture scattered light and convert intensity values into voltage. After data acquisition, the last and most problematic step begins: data analysis. Voltage values are send through a computer and visualized using specific software. Cell populations of interest are marked on 2D scatter plots manually by experts. Identification of distinct cell populations is important to monitor or diagnose the disease.


Sample preparation
Instrument setup
Data acquisition
Data analysis
Cryopreserve
Draw blood from subject
Separate mononuclear cells using a Ficoll gradient
Stain with fluorescent antibody conjugates
Baseline PMT voltage determination
Optimize fluorescence detector sensitivity
Cell sample
Fluorescence detector
Laser light source
Optical filter
Scattered light
Pass stained cell suspension through laser beam
CD3⁺ cells
CD8
CD4
Gate cell populations of interest

Population	Ref. mean	Ref. SD	Sample
% of T cells (total lymphocytes)	72.3	6.2	70.7
% of CD4⁺ cells (total T cells)	67.4	7.0	63.2
% of CD8⁺ cells (total T cells)	28.6	6.6	30.4

Nature Reviews | Immunology

Figure 2.1. A typical flow cytometry experiment

(Source: Maecker et al., 2012)

A flow cytometer consists of three main parts; fluidics, optics and electronics. In fluidics part, the cells or particles are transported to interrogation point where each cell collides with the laser beam. The most important role of fluidics part is to provide that one cell or particle has to move through the laser beam at a time. To put this in effect, the sample is injected in the core flow chamber and the particles are accelerated and centered by the pressure of the sheath fluid which surrounds the core flow chamber (Sa et al., 2013). This process is called hydrodynamic focusing. In hydrodynamic focusing, sample pressure is always greater than the sheath fluid pressure and the proportion between the two pressures allows to determine the number of cells that pass through a laser beam in a minute.

The optics part consists of lasers and optical filters and this part is the most complicated part. Lasers produce a single wavelength light and in a flow cytometer, there can be more than one laser source. Forward scatter values give information about the cell size, larger cells produce larger forward scatter signal, while side scatter values are related with cell structure and granularity. The cell size and structure information are sufficient to distinguish numerous cell types but adding biomarkers to the cells gives opportunity to identify specific structures. Flow cytometer uses several side detectors with optical filters to distinguish the biomarker effect. The dichroic optical filter performs two functions; firstly it allows to pass the light with specified wavelength to the side detector, second it deflects the light with unspecified wavelength.

Cells using forward and side scatter values according to their physical characteristics. Also, cells can be separated by using fluorochromes as a biomarker to detect whether a cell express a target protein or not. Fluorochoromes have unique spectra for excitation and emission. A single fluorochrome is excited at a particular wavelength by the flow cytometer laser and it emits the light at a longer wavelength. When a fluorochrome is excited, it absorbs the light and its electrons move from a ground state ($S0$) to maximal energy level ($S2$). The duration of the excitation state depends on the fluorochrome but typically it is $1 - 10$ nanoseconds. After excitation, electrons release energy, through fluorescence, while they fall to lower and more stable energy level ($S1$). At the end, the electrons turn back to ground state energy level. This process is summarized using Jablonski Energy Diagram illustrated in Figure 2.2 (Ermolaev and Lubimtsev, 1987).

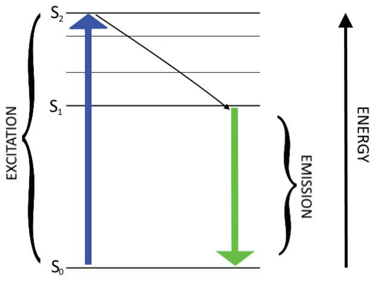


Figure 2.2. Jablonski Energy Diagram

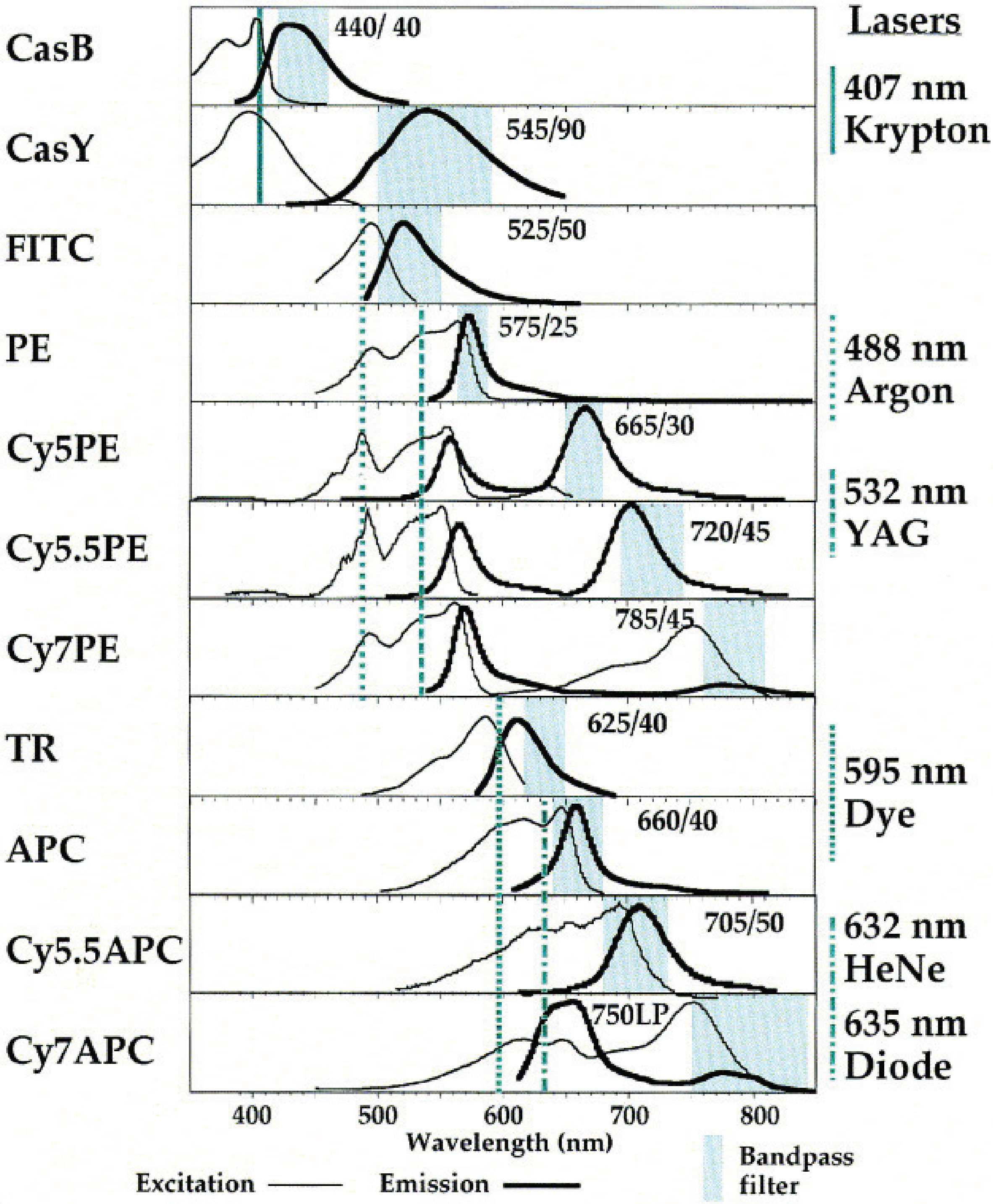


Figure 2.3. Excitation and Emission Spectrum of Common Dyes in Flow Cytometry Experiment

(Source: Baumgarth and Roederer, 2000)

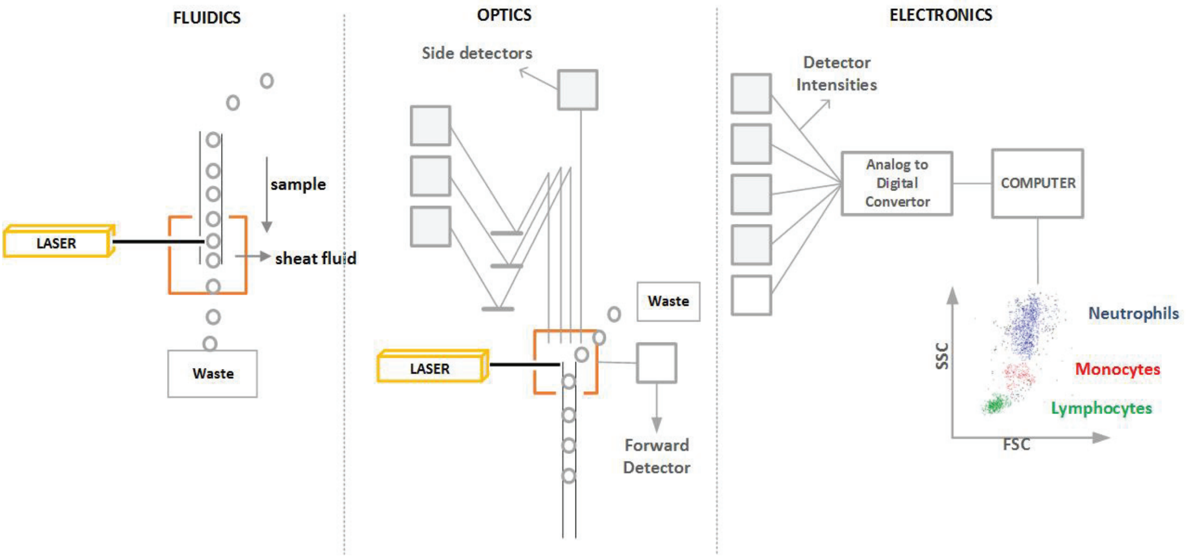


Figure 2.4. Flow Cytometry Parts

The wavelength or frequency difference between excitation and emission spectra is called as Stokes shift. The Stoke shift value is fluorochrome specific, so it changes according to the chosen fluorochrome. Flow cytometer uses several side detectors with optical filters to distinguish the biomarker effect. The laser, optical filters and detectors are shown in Figure 2.4.

The electronics part converts the light intensity that is captured by the forward and side detectors to analog voltage information in photomultiplier tubes (PMT). The voltage pulse generated by a single cell is directly related with the cell structure and fluorescence intensity. This analog information is then converted to digital information that can be processed by the computer. Some flow cytometers also have cell sorting functionality that can collect one cell type of choice using electrostatic charge, much like the operating principle of ink jet printers.

2.1.2. Data Structure and Analysis

In flow cytometry experiments, a single cell produces multivariate data that correspond to forward scattered light, side scattered light and the light captured with $FL_1 - FL_n$ detectors. When the scattered light information is converted to digital value it is stored in computer with a specific file format developed by the Society for Analytical Cytology, called as Flow Cytometry Standard (FCS) format (CYT, 1990). Table 2.1 shows a matrix representation of the flow cytometry data. Events represent individual cells and markers represent detectors.

Intensity vales are indicated by I_{ij} and i represents the event number and j represents the respective marker. Once a data file has been saved, flow cytometry data can be displayed in different forms in specific FCM data analysis softwares such as: FlowJo (Tree Star, Ashland, OR), FCS Express (De Novo Software, Glendale, CA) etc. Generally two or three dimensional plots are used with one parameter in each axis for data visualization.

Event	Marker 1	Marker 2	...	Marker N
1	I_{11}	I_{12}	...	I_{1N}
2	I_{21}	I_{22}	...	I_{2N}
$\vdots$				
10^6	$\vdots$	$\ddots$	$\ddots$	$\vdots$

Table 2.1. Matrix Representation of Flow Cytometry Data

Analysis of the flow cytometry data requires pre-processing steps such as data transformation and compensation. In the next section, we provide a description and literature review of the flow cytometry data analysis pre-processing steps in detail with illustration on a synthetically created toy dataset.

2.1.2.1. Data Preprocessing

The raw flow cytometry data contains intensity measurements captured from detectors and these raw data need multiple pre-processing steps to improve data quality and visualization. The first pre-processing step is data tranformation. Generally, cell populations are described using log-normal transformation, however there are some issues using this transformation. For example, log-normal transform does not work for normalized data if it contains negative values. In this case, using alternative transformations, such as logicle (Parks and Moore, 2005), Box-Cox (Box and Cox, 1964), generalized hyperbolic arcsin and son on, is a more effective solution.

Another pre-processing step is compensation. It is required to remove the spillover effects, caused by overlap, between the emission spectra of the fluorochromes. Control samples are stained using only one fluorochrome to create a baseline measurement. The baseline measurements for each channel are stored in a spillover matrix. The compensated data is generated by inverting the spillover matrix and multiplying it with the uncompensated raw data. Below, we elucidate different data transformation scales and the manual compensation procedure.

Data Transformation

Scale transformation is important for making quantitative comparisons of distance between intensity levels of samples in flow cytometry as well as for better visualization of cell sub groups. The basic data visualization scale in flow cytometry data analysis is the log scale (Finak et al., 2010). It can stabilize the variance of cell populations and it can be defined as;

$$S(x) = \log(x) \quad , \qquad x > 0 \tag{2.1}$$

The contradiction of the log transformation is that it cannot represent the negative data values of unstained cells, and it causes poor visualization for low intensity and unstained data.

Due to the problems of the log-transform with negative values, an alternative transformation is defined based on the hyperbolic sine function (Parks et al., 2006). It's called as generalized *arsinh* transformation.

Due to the problems of the log-transform with negative values, an alternative transformation is defined based on the hyperbolic sine function (Parks et al., 2006). It's called as generalized *arsinh* transformation.

$$S(x) = \frac{1}{2}(e^{x} - e^{-x}) \tag{2.2}$$

The log scale transformations can be further generalized as biexponential transformation which is described in Equation 2.3 with parameters a, b, c, d, f, w (Parks et al., 2006).

$$S(x; a, b, c, d, f) = ae^{(b(x-w))} - ce^{(-d(x-w))} + f \tag{2.3}$$

All improvements in log scale seek linear representation of negative and low intensity values and logarithmic representation of higher intensity values while exhibiting a smooth transition between the extreme values of the raw data (Finak et al., 2010). A subset of the biexponential functions that are linear near zero are called logicle function. The logicle method solves this problem by plotting data on axes that are linear around a data value of zero and logarithmic at higher (positive and negative) values (Parks et al., 2006) and it is the specialized biexponential tranformation for flow cytometry data visualization. It is defined as

$$S(x; T, m, w, p) = Te^{-(m-w)}(e^{x-w} - p^{2}e^{-(x-w)/p}) + p^{2} - 1 \quad \text{for} \quad x \geq x \tag{2.4}$$

where T defines the maximum data value to be displayed, m is the range of the display, w is the range of the linearization around 0 and p is introduced for compactness. However, a relation between p and w is defined trough Equation 2.5, and thus, p and w together represent a single adjustable parameter. The parameters m and w are in the units of natural logarithm; typically a range of 10^4 is specified as $m = 4ln(10) = 9.23$

$$w = \frac{2pln(p)}{p+1} \tag{2.5}$$

In statistical learning, if a distribution fits a dataset well, one can easily apply statistical methods. Using Box-Cox transformation on flow data is useful in some cases. The Box-Cox transformation yields a dataset that follows approximately a normal distribution (Box and Cox, 1964). The Box-Cox transformation is defined as

$$S(x) = \frac{x^{\lambda} - 1}{\lambda} \tag{2.6}$$

where λ is the transformation parameter. When $x \ll 1$ the expression above approaches the indeterminate form 0/0, and the Box-Cox formula is redefined for $\lambda = 0$ as

$$\begin{aligned} S(x) &= \frac{e^{\lambda log(x)} - 1}{\lambda} && (2.7) \\ &\approx \frac{(1 + \lambda log(x) + \frac{1}{2}\lambda^2 log(x)^2) - 1}{\lambda} && (2.8) \\ &\approx log(x). && (2.9) \end{aligned}$$

We have created a toy dataset with three distinct clusters, C_1, C_2 and C_3, to illustrate the effects of the various data transformation methods. The data consists of exponential of three normal distribution with different sample sizes and different mean and covariance matrices. The normal distributions are defined for clusters C_1, C_2 and C_3 respectively as;

$$\Sigma_i = \sigma_i^2 \times I_{3\times 3} \quad \text{for} \quad i = 1, 2, 3 \tag{2.10}$$

where $\sigma_1^2 = 0.6$, $\sigma_2^2 = 0.8$, $\sigma_3^2 = 0.5$ and mean vectors are

$$\mu_1 = \begin{bmatrix} 8 & 8 & 8 \end{bmatrix}^T \tag{2.11}$$

$$\mu_2 = \begin{bmatrix} 4 & 4 & 4 \end{bmatrix}^T \tag{2.12}$$

$$\mu_3 = \begin{bmatrix} 4 & 8 & 8 \end{bmatrix}^T. \tag{2.13}$$

with sample sizes $N_1 = 1000$, $N_2 = 800$ and $N_3 = 1000$. In Figure 2.5 we gave histogram of the intensity values of all three variates and three dimensional scatter plot of the synthetically generated toy dataset. Any statistical method or human eye can not detect without any transformation that the dataset has three different clusters. So, data transformation is needed and we applied log, arsinh, logicle and Box-Cox transformation.

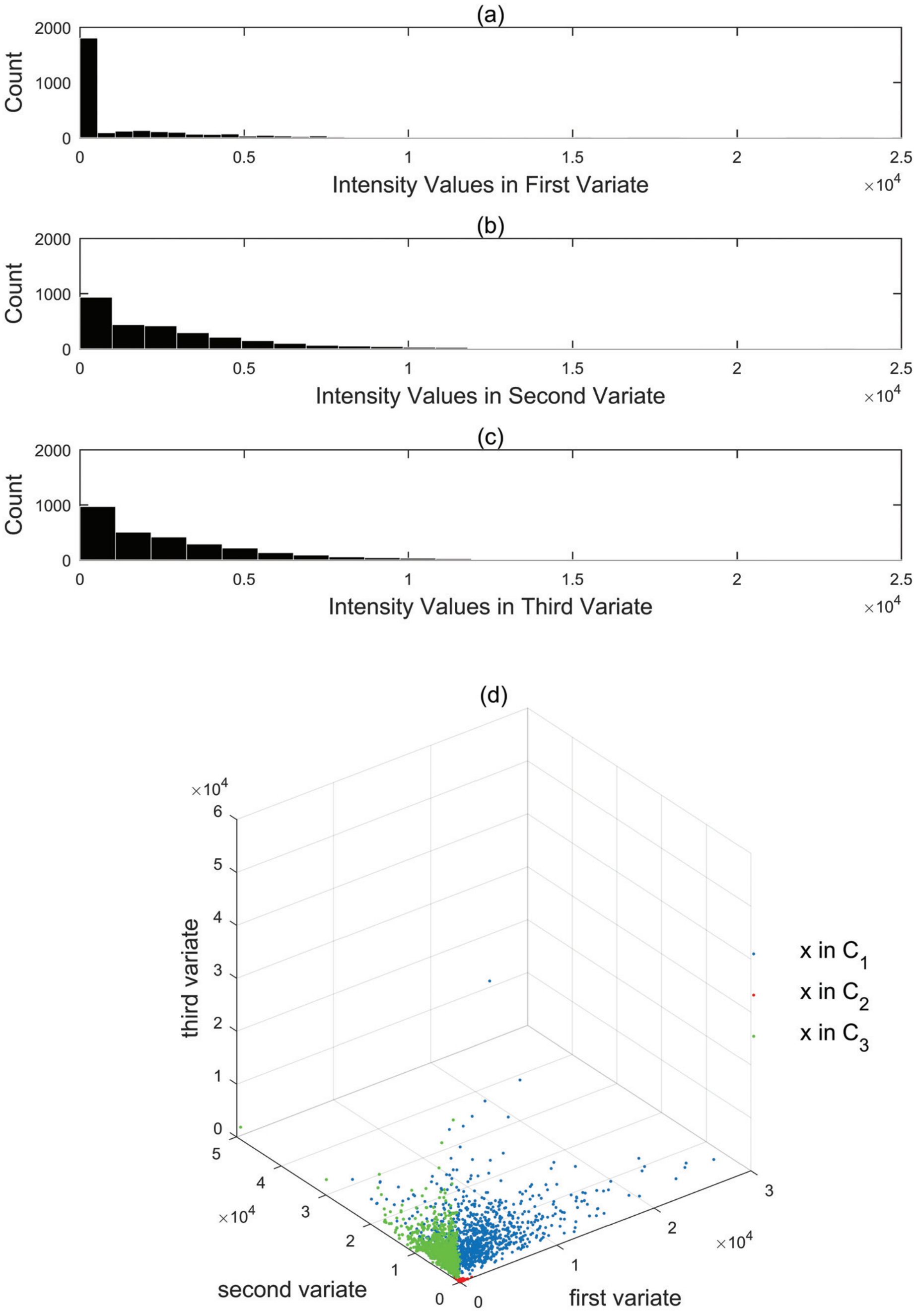


Figure 2.5. Raw Data Representation: (a)-(c) Histogram plots of first, second and third variate respectively, (d) 3D Scatter plot of raw data

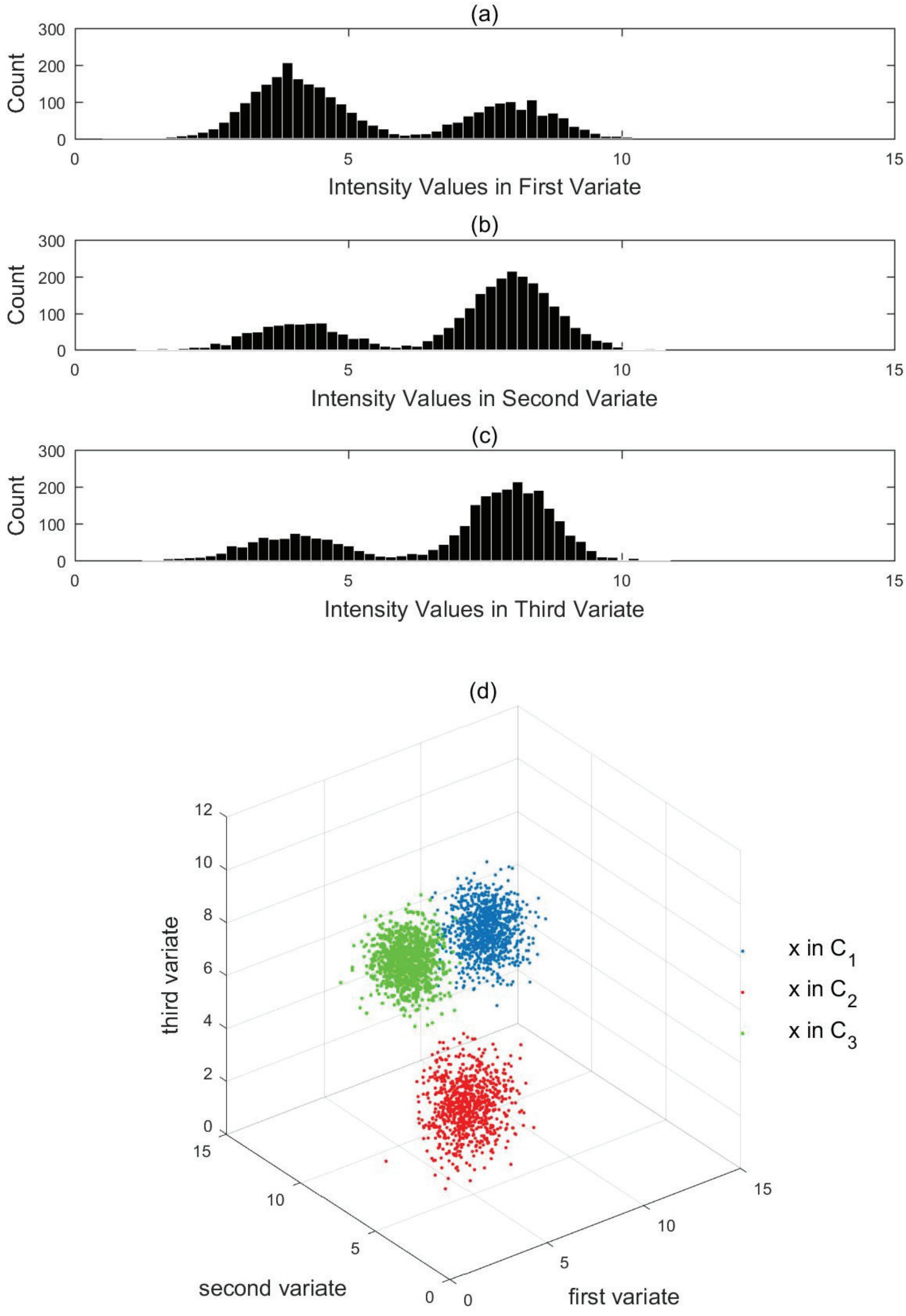


Figure 2.6. Log Scale Data Representation: (a)-(c) Histogram plots of first, second and third variate of log-scale data respectively, (d) 3D Scatter plot of log scale data

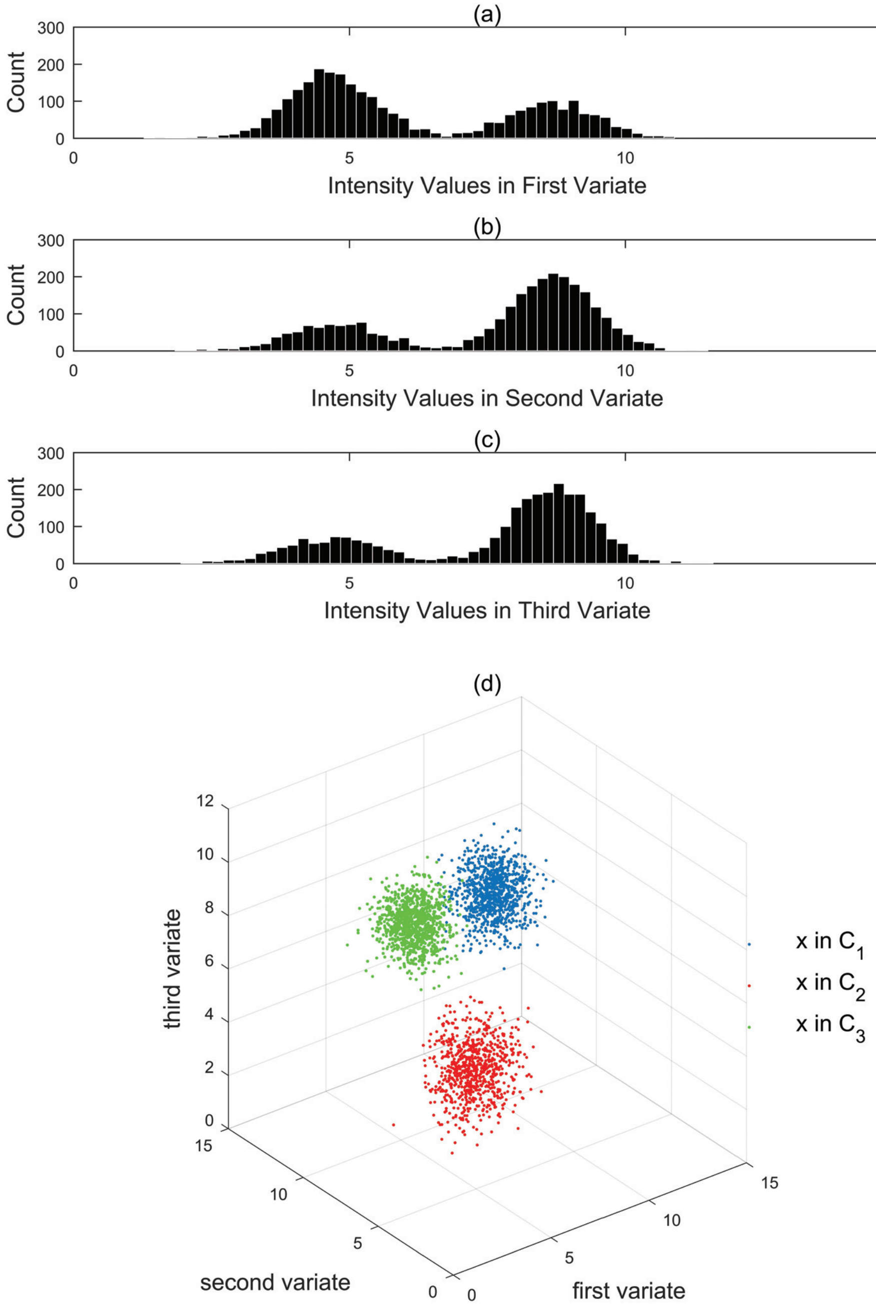


Figure 2.7. Arsinh Scale Data Representation: (a)-(c) Histogram plots of first, second and third variate of inverse hiyperbolic sine data respectively, (d) 3D Scatter plot of inverse hiyperbolic sine scale data

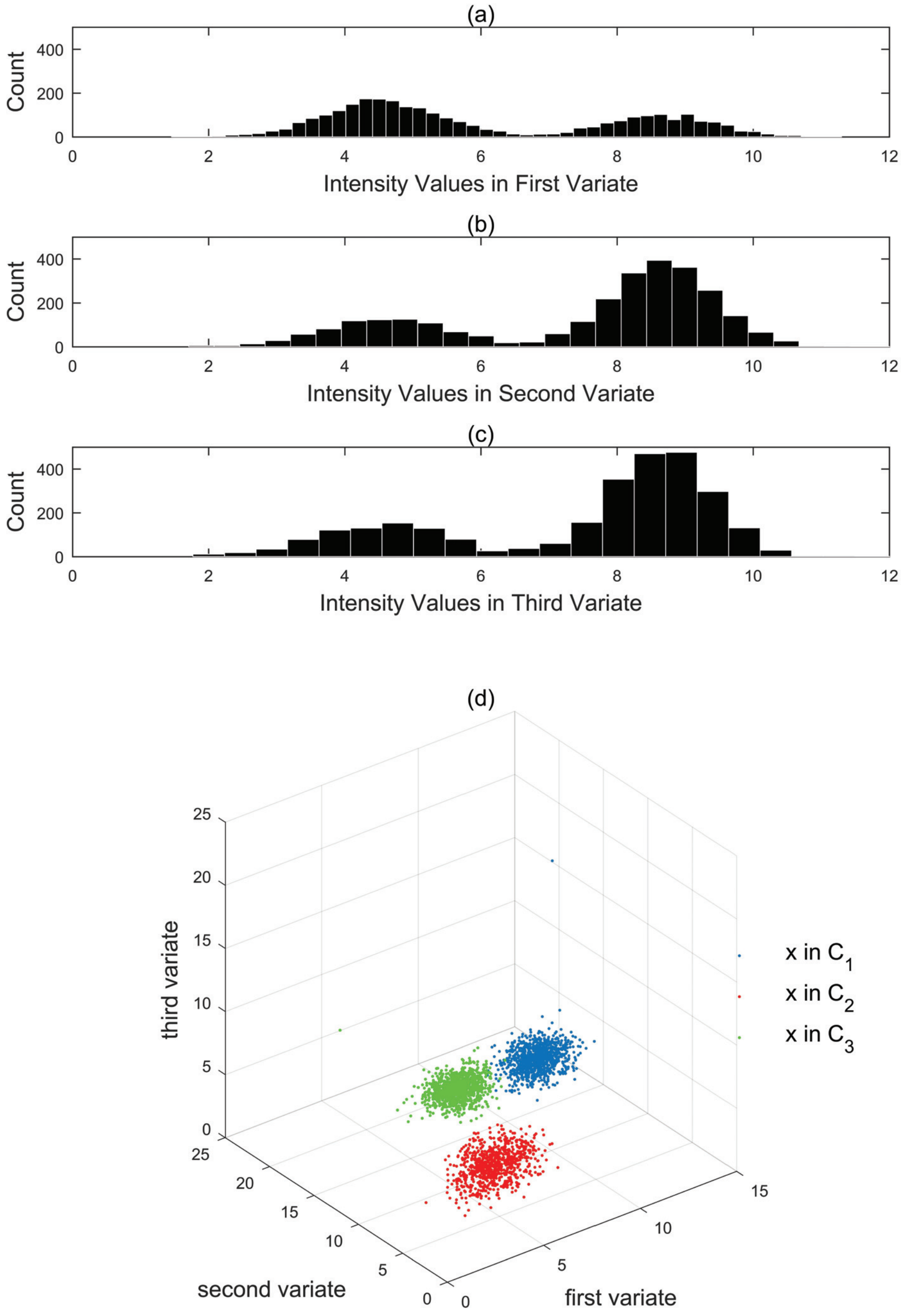


Figure 2.8. Logicle Scale Data Representation: (a)-(c) Histogram plots of first, second and third variate of logicle scale data respectively, (d) 3D Scatter plot of logicle scale data

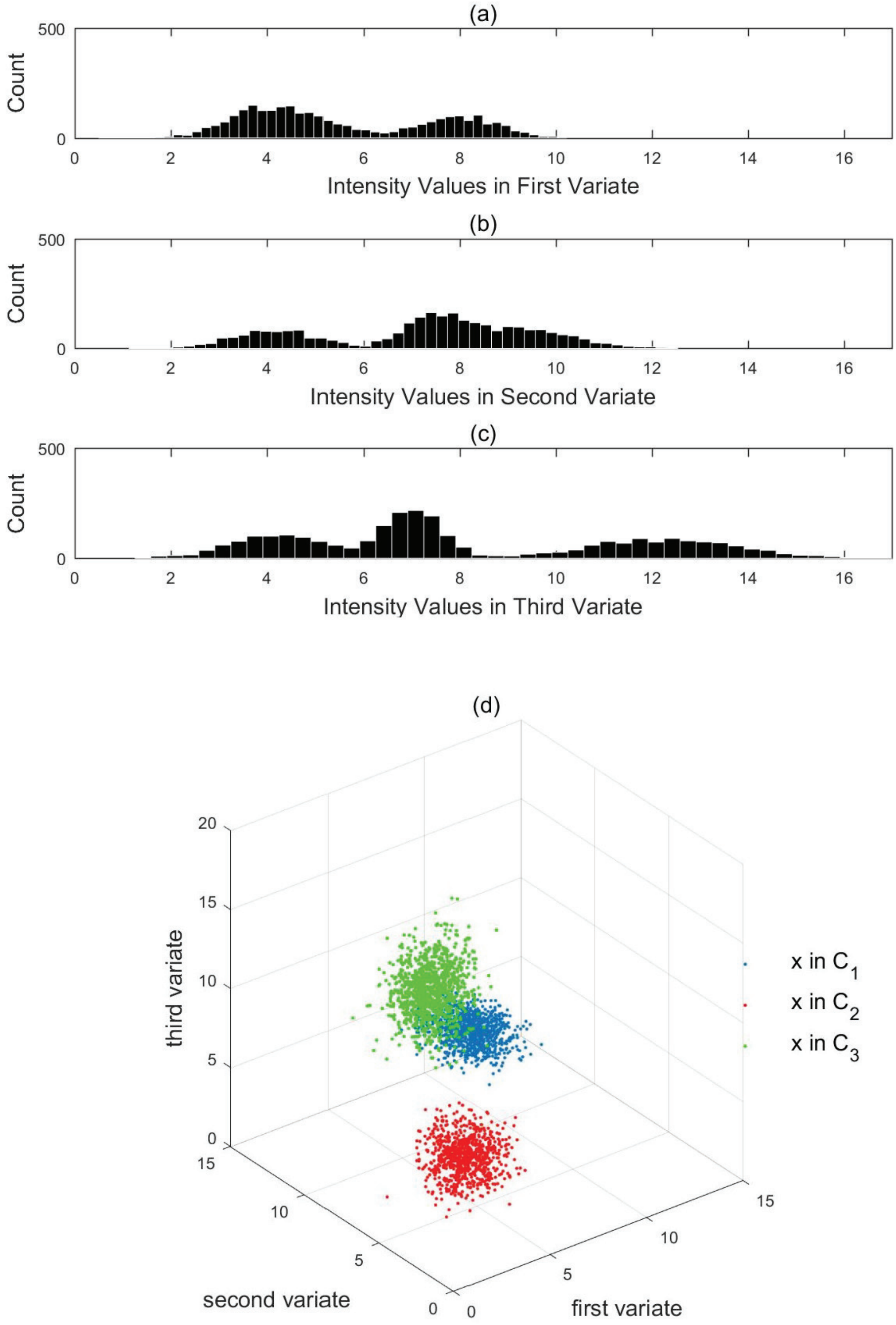


Figure 2.9. Box-Cox Data Representation: (a)-(c) Histogram plots of first, second and third variate of Box-Cox transformed data respectively, (d) 3D Scatter plot of Box-Cox transformed data

Compensation

All fluorochromes have specific excitation and emission spectra and flow cytometry measures cell properties using emitted light from these fluorochromes. Optical filters on the detectors capture the scattered light in a limited frequency range and when two or more fluorochromes' emission spectra overlap optical filters fail to detect which fluorochrome the increasing light is coming from (Sugár et al., 2011). Figure 2.10 illustrates the FITC spillover into PE channel. The amount of the spillover is a linear function; the data can thus be corrected using linear operations, through a procedure known as compensation.

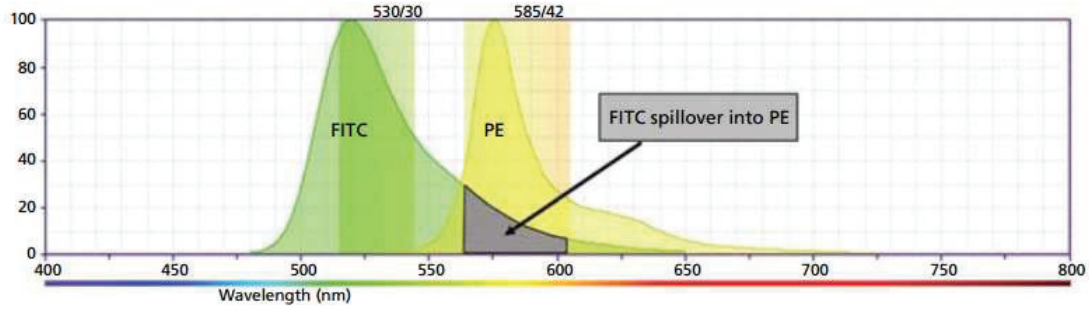


Figure 2.10. Example of FITC spillover into the PE channel

(Source: $https://www.bdbiosciences.com$)

Compensation can be performed either during data collection by flow cytometer or after data collection with specific software. The most important point in compensation is visualization of all distinct subpopulations as separate as possible from each other. Compensation procedure is performed by experts using control samples, in each experiment control samples and machine calibration is needed. Thus, this process is laborious work and it can be cause interpretation differences between experts and all experts can obtain different compensation parameters. A typical compensation procedure is summarized below;

- Firstly, unstained cell samples are passed through the flow cytometer and FCS and SSC detectors are adjusted to display the cell groups of interest on scale.
- Secondly, the spillover for all fluorochromes on all detectors are measured using single-color controls. The spillover values are then placed on a symmetric matrix.
- Finally, the compensation matrix is obtained by inverting the spillover matrix described in the second step.

To illustrate the procedure on two channels, Bagwell & Adams put forth a binary communication channel model (Bagwell and Adams, 1993b). Suppose s_1 and s_2 are original signals that represent fluorescence signals from fluorochrome 1 and 2 (FL_1 and FL_2), and

o_1 and o_2 are the observed fluorescence signals. Supppose also that k_{12} and k_{21} are the proportions of signal s_1 crossing over o_2 and s_2 crossing over o_1, respectively. A diagrammatic representation of this two-signal crossover system is illustrated in Figure 2.11. The observed signals o_1 and o_2 can then be obtained by

$$o_1 = (1 - k_{12})s_1 + k_{21}s_2 \tag{2.14}$$

$$o_2 = k_{12}s_1 + (1 - k_{21})s_2. \tag{2.15}$$

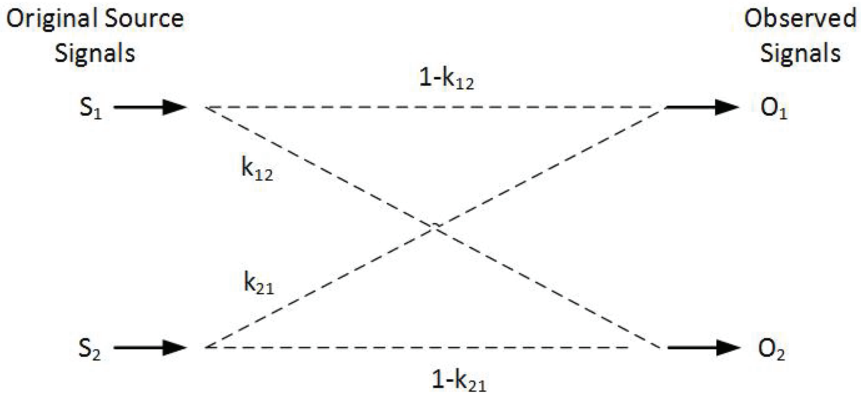


Figure 2.11. Diagrammatic representation of a two-signal crossover system

Algebraically solving Equation 2.14 and 2.15 for s_1 and s_2 in terms of o_1 and o_2, we get

$$s_1 = o_1\left(\frac{1 - k_{21}}{1 - k_{12} - k_{21}}\right) + o_2\left(\frac{-k_{21}}{1 - k_{12} - k_{21}}\right) \tag{2.16}$$

$$s_2 = o_1\left(\frac{-k_{12}}{1 - k_{12} - k_{21}}\right) + o_2\left(\frac{1 - k_{12}}{1 - k_{12} - k_{21}}\right) \tag{2.17}$$

The crossover constants, k_{12} and k_{21}, are estimated by appropriately analyzing two controls: fluorochrome 1 alone and fluorochrome 2 alone. In the notation of the formulation that follows, the subscript represents the signal and the superscript represents the control. Therefore, s_2^1, the signal s_2 when only control samples s_1 sent, and s_1^2, the signal s_1 when only control samples s_2 sent, are both 0 by definition. The crossover coefficient k_{12} can be calculated by

$$k_{12} = \frac{o_2^1}{o_1^1 + o_2^1} \tag{2.18}$$

using one event. If the fluorochrome 1 control contains n_1 events, the complete solution is

$$k_{12} = \frac{\sum^{n_1} o_2^1}{\sum_{n_1} o_1^1 + \sum_{n_1} o_2^1} \tag{2.19}$$

In the same manner k_{21} can be written with n_2 control events for fluorochrome 2 as

$$k_{21} = \frac{\sum^{n_2} o_2^2}{\sum_{n_2} o_1^2 + \sum_{n_2} o_2^2} \tag{2.20}$$

The source signals can be calculated by incorporating these crossover parameters into Equations 2.16 and 2.17. Moreover, we can write Equation 2.14 and 2.15 in matrix form as follows:

$$\begin{bmatrix} o_1 \\ o_2 \end{bmatrix} = \begin{bmatrix} 1-k_{12} & k_{21} \\ k_{12} & 1-k_{21} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \tag{2.21}$$

When we increase the number of fluorochromes, we need an expansion for two fluorochrome compensation and algebraically, calculation becomes more complex. We have illustrated a three signal crossover system in Figure 2.12. For an n-fluorochrome experiment, we can write the observed signals in matrix form as

$$\begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_n \end{bmatrix} = \begin{bmatrix} 1-k_{12} & k_{21} & \dots & k_{n1} \\ k_{12} & 1-k_{21} & \dots & k_{n2} \\ \vdots & \ddots & \ddots & \vdots \\ k_{1n} & k_{2n} & \dots & 1-k_{n1}-\dots-k_{nn-1} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \tag{2.22}$$

or equivalently

$$O = KS \tag{2.23}$$

where the O matrix denotes the observation signals that contains measured intensity values, the K matrix is the crossover coefficient matrix combining the spillover coefficients and S is the source signal matrix that we need to find for compensation. The source signals can then be found using matrix inversion as

$$S = K^{-1}O \tag{2.24}$$

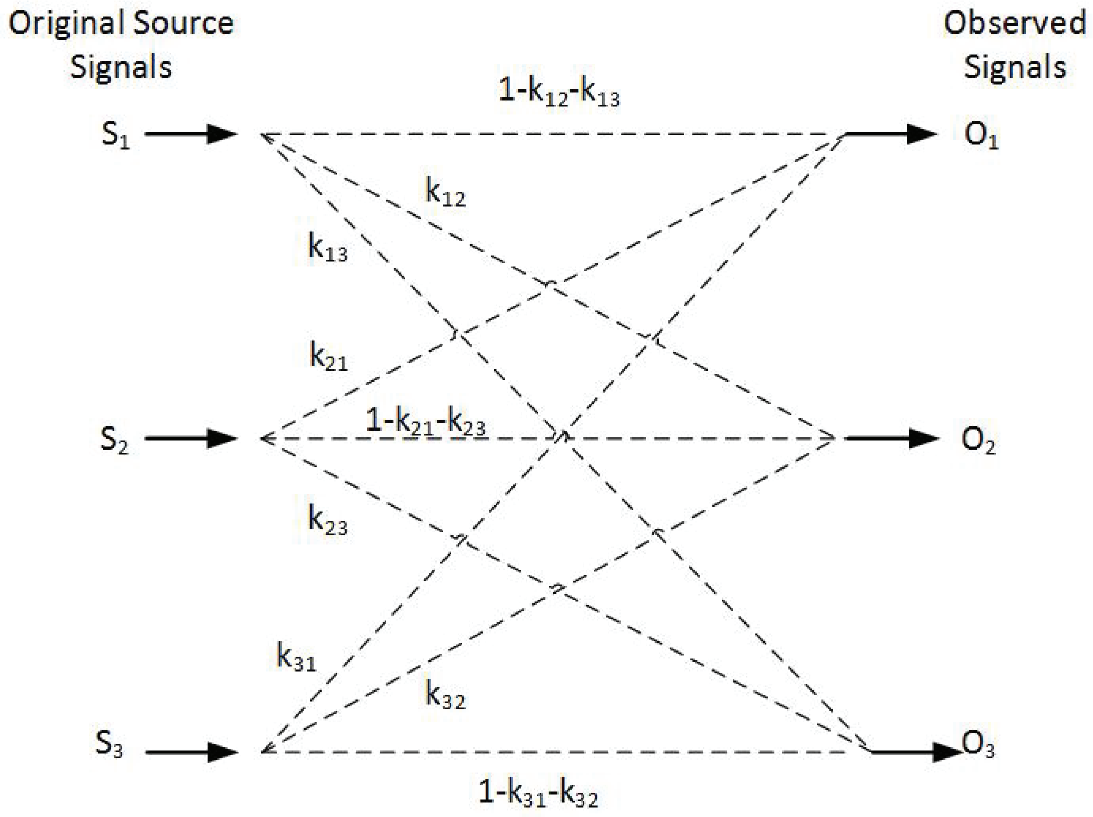


Figure 2.12. Diagrammatic representation of a three-signal crossover system

This formulation is valid when the compensation problem is constructed with the assumption of no autofluorescence. Otherwise, autofluorescence parameters must be added to the formulation. Furthermore, (Bagwell and Adams, 1993b) manual compensation is practically impossible for more than 3 fluorochromes, thus several research groups are focusing on automated compensation of multi-color flow cytometry data. However, all proposed methods to date rely on calculating the spillover matrix using control beads (Roederer, 2001; Sugár et al., 2011). Using control samples in an experiment causes many problems such as workload, non-repeatable experiment etc. Because flow cytometer calibration changes and this requires to create a new spillover matrix in each experiment. To do this, control samples are passed through the cytometer and manually compensation matrix is created before each experiment and since this process depends on the expert knowledge and there can be differences between manually created compensation matrices by the flow cytometry experts.

2.1.2.2. Gating

Identification of cell groups in an FCM dataset is called gating. In standard gating techniques, this is done manually by a pathologist. The data from the flow cytometer is plotted in a single dimension as a histogram plot or two-three dimensional scatter plots. The plots can then be divided into regions depending on the intensity of the fluorescence and gates are created as manually drawn region on the display. Gates or regions are marked on scatter plots to focus on specific population interest. Gating is a sequential process; after determining a gate, addictive gates are formed in that population possibly are remaining channels. Since the gates are determined by expert-knowledge, saying the how much information gets lost is difficult.

The first step in gating is distinguishing cell populations according to their light scatter properties such as size and granularity. For example, dead cells cause lower forward scatter values than live cells. The most popular example in flow cytometry gating is the lymphocyte gate shown in Figure 2.13. Since lymphocytes have smaller cell size and lower internal complexity compared to monocytes and neutrophils, their forward and side scatter intensity values are smaller. Likewise, neutrophils are of more granular structure than the others. As a result of this, their side scatter values are greater.

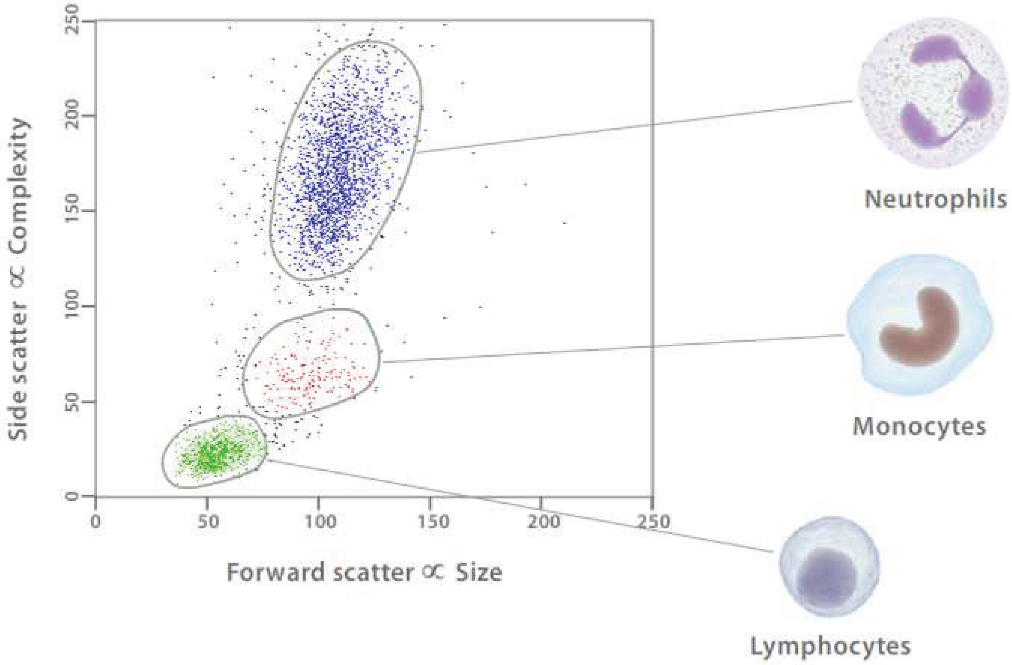


Figure 2.13. Gating Example

(Source: http://probes.invitriogen.com)

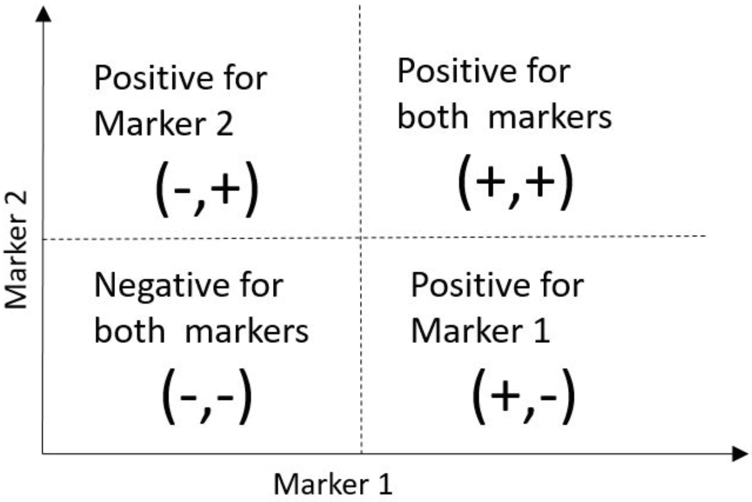


Figure 2.14. Gating strategy using two markers

In multi-color flow cytometry experiments, gating is also performed according to whether cells are positive or negative for respective biomarkes. In this procedure, the pathologist determines four quadrants on 2D scatter plots using two biomarkers in different data transformation scales such as log, arsinh, biexponential and so on to establish distinct cell sub groups. In this case, the dimensionality of the data creates problem: Human intuition is limited to two or three dimensional scatter plots. Typically, the subpopulations are identified by experts on 2 dimensional scatter plots, so this process is laborious and time-consuming. The developed technology currently allows analyzing cells up to more than 20 biomarkers, and this generates complex and high dimensional datasets. Furthermore, there are concerns over the reproducibility of the results, even by the same expert on the same FCM data (Lo et al., 2008). The main objective in computational analysis of flow cytometry data is automatic identification of cell populations in a heterogeneous population. In the literature, there are several methods that have been proposed to this end. Pyne et al. developed a mixture modelling approach that fits skew and heavy distributions to cell populations. They used expectation maximization algorithm to estimate likelihood when the algorithm optimally fits *k*-variate distributions to the available subpopulations (Pyne et al., 2009). The problem here is the determination of a specific distribution with optimal parameters and the determination of the number of subpopulations in the data. Aghaeepour et al. proposed automated cell subset identification method based on *k*-means clustering that can capture concave populations using multiple clusters (Aghaeepour et al., 2013). FlowClust algorithm proposed by Lo et al. fits a *t*-mixture model following a Box-Cox transformation (Lo et al., 2008). Finak et al. modi-

Table 2.2. Most Popular Algorithms for Automated Gating of Flow Cytometry Data

Algorithm Name	Supervised(S) / Unsupervised(U)	Automated # of Clusters	Ref
FLAME	U	Y	Pyne et al., 2009
FLOCK	U	Y	Qian et al., 2010
FlowClust/Merge	U	Y	Lo et al., 2008; Finak et al., 2009
FlowMeans	U	Y	Aghaeepour et al., 2011
RadialSVM	S	N	Quinn et al., 2007
GemStone	U	N	Miller et al., 2012

fied the FlowClust algortihm as FlowMerge that adds a merging step after all subpopulations are identified to eliminate superfluos cluster division (Finak et al., 2009). The most problematic part of these flow cytometry analysis methods is the determination of the actual number of clusters in the dataset. Most of these methods use Bayesian information criteria (BIC), Akaike information criteria (AIC) or entropy based cost functions to determine the number of clusters. This means that, in order to identify the number of clusters the algorithm should be run several times, and after that, optimal parameters such as the number of clusters are obtained according to one of these criteria. We have summarized most popular algorithms for automated gating of flow cytometry data in Table 2.2.

2.2. Quasi-supervised Learning Algorithm

In data clustering, both supervised and unsupervised methods have some challenges. Supervised methods need ground truth datasets for training, and unsupervised methods mostly propose fitting a distribution to dataset. Quasi-supervised learning (QSL) algorithm is basically constructed for identifying the two contrast clusters in a data by estimating the posterior probabilities of individual samples belonging in different groups (Karaçalı, 2010).

In previous works, QSL algorithm was proposed for recognition applications where labels are available for only one group of data (control samples). A second unlabeled data is also available which contains both control and target samples. QSL algorithm surpasses the alternative methods (support vector machines classification and minimum spanning trees) in synthetically created target identification problems under different scenarios. Önder et al. (Onder et al., 2013) and Köktürk et al. (Köktürk and Karaçalı, 2013) identified the tumor regions on colon histopathology images with different resolutions. Köktürk et al. also expand the QSL algorithm for multi-class problems and created M-ary QSL algorithm. They applied

M-ary QSL algorithm on electroencephalography data recorded under six different stimuli to identify the stimulus specific brain activity patterns (Köktürk and Karaçalı, 2012). Güven et al. has used QSL algorithm on aerial images to detect man-made enviroments in natural structures for military purposes (Güven, 2010).

In the next section, we have explained QSL algorithm which is also explained by Karaçalı in detail (Karaçalı, 2010). Since it does not need any knowledge about the dataset and it can automatically estimate posterior probability of individual sample in the dataset, we used the QSL algorithm for automatic data clustering instead of abnormality detection. We used the estimated posterior probability values as the responsibility value of the expectation-maximization algorithm and we created a clustering algorithm that reduces overlap between clusters by using expectation-maximization routine with quasi-supervised learning algorithm.

2.2.1. Analytical Computation of Posterior Probabilities using Quasi-Supervised Learning

We can define a nearest neighbour classifier $F_R(x)$ for a given dataset points $x_i \in X$ and their respective class labels $y_i \in \{0, 1\}$ where $i = 1, 2, \ldots, l$;

$$F_R(x_i) = y_{i_0} \quad \text{with} \quad i_0 = \underset{i=1,2,\ldots,l}{\operatorname{argmin}}\, d(x, x_i) \tag{2.25}$$

where $d(\cdot, \cdot)$ denotes the distance metric on X. The quasi-supervised learning algorithm envisions M identically distributed independent reference sets $R_j = \{x_i, y_i\}$ with $j = 1, 2, \ldots, M$ that consists of n points from two classes and the average fraction of times the nearest neighbor classifier with reference set R_j assigns a point x to the two classes: C_0 and C_1

$$f_0(x) = \frac{1}{M} \sum_{j=1}^{M} \mathbf{1}(F_{R_j}(x) = 0) \tag{2.26}$$

$$f_1(x) = \frac{1}{M} \sum_{j=1}^{M} \mathbf{1}(F_{R_j}(x) = 1) \tag{2.27}$$

For sufficiently large M, it can be shown that

$$f_0(x) \simeq \frac{p(x|x \in C_0)}{p(x|x \in C_0) + p(x|x \in C_1)} \tag{2.28}$$

$$f_1(x) \simeq \frac{p(x|x \in C_1)}{p(x|x \in C_0) + p(x|x \in C_1)} \tag{2.29}$$

While carrying out an exhaustive evaluation of all possible random nearest neighbor classifications is not feasible, it is still possible to compute the average number of times a given point would be assigned to either class at the end of such an evaluation. After an exhaustive nearest neighbor analysis, $f_1(x)$ represents the probability $Pr\{y = 1\}$ of assigning x to the class C_1 based on a reference set R with n points from both classes selected randomly from $\{x_i\}$:

$$f_1(x) = Pr\{y = 1\} \tag{2.30}$$

This probability can be decomposed over sorted samples $x_{(i)}$ conditionally on whether or not the point $x_{(1)}$ nearest to x is in R, providing

$$f_1(x) = Pr\{x_{(1)} \in R\}\mathbf{1}(y_{(1)} = 1) + Pr\{x_{(1)} \notin R\}Pr\{y = 1|x_{(1)} \notin R\} \tag{2.31}$$

since $Pr\{y = 1|x_{(1)} \in R\}$ is 1 if $y_{(1)} = 1$, and 0 otherwise. For notational simplicity, we can define E_k that describes the joint event where $x_{(1)}, x_{(2)}, \ldots, x_{(k)} \notin R$. So we can write Equation 2.31 as,

$$f_1(x) = Pr\{x_{(1)} \in R\}\mathbf{1}(y_{(1)} = 1) + Pr\{x_{(1)} \notin R\}Pr\{y = 1|E_1\} \tag{2.32}$$

$Pr\{y = 1|E_1\}$ can be further decomposed as follows:

$$Pr\{y = 1|E_1\} = Pr\{x_{(2)} \in R|E_1\}\mathbf{1}(y_{(2)} = 1) + Pr\{x_{(2)} \notin R|E_1\}Pr\{y = 1|E_2\} \tag{2.33}$$

We can thus generalize this decomposition for $Pr\{y = 1|E_{k-1}\}$ as

$$Pr\{y = 1|E_{k-1}\} = Pr\{x_{(k)} \in R|E_{k-1}\}\mathbf{1}(y_{(k)} = 1) + Pr\{x_{(k)} \notin R|E_{k-1}\}Pr\{y = 1|E_k\}. \tag{2.34}$$

Then, Equation 2.31 becomes

$$f_1(x) = Pr\{x_{(1)} \in R\}\mathbf{1}(y_{(1)} = 1) + Pr\{x_{(1)} \notin R\}(Pr\{x_{(2)}\} \in R|E_1)\mathbf{1}(y_{(2)} = 1) + \ldots + \\ Pr\{x_{l-1}| \notin R\}(Pr\{x_{(l)} \in R|E_{l-1}\}\mathbf{1}(y_{(l)} = 1) + Pr\{x_{(l)} \notin R|E_{l-1}\}Pry = 1|E_l)\ldots). \quad (2.35)$$

Note that the reference set R must contain $2n$ data points from each class. This means that we do not need to continue the decomposition beyond a point k^* which defined as;

$$k^* = \max\left\{k|\sum_{k'=k}^{l}\mathbf{1}(y_{k'} = 0) \geq n \quad \text{and} \quad \sum_{k'=k}^{l}\mathbf{1}(y_{k'} = 1) \geq n\right\} \quad (2.36)$$

since $Pr\{x_{(k^*)} \in R|E_{k^*-1}\} = 1$ and $Pr\{x_{(k^*)} \notin R|E_{k^*-1}\} = 0$. Note that $Pr\{x_{(k)} \in R|E_{k-1}\}$ defined in Equation 2.34 can be calculated by

$$\begin{aligned} Pr\{x_{(k)} \in R|E_{k-1}\} &= 1 - Pr\{x_{(k)} \notin R|E_{k-1}\} \quad (2.37) \\ &= 1 - \frac{\binom{l_0^{k+1}}{n}\binom{l_1^{k+1}}{n}}{\binom{l_0^{k}}{n}\binom{l_1^{k}}{n}} \quad (2.38) \end{aligned}$$

where l_0^k denotes the number of points that belong to C_0 in the set $\{x_{(k)}, x_{(k+1)}, \ldots, x_{(l)}\}$ and l_1^k represents the number of points that belong to C_1 on the same set:

$$l_0^k = \sum_{i=k}^{l}\mathbf{1}(y_{(i)} = 0) \quad \text{and} \quad l_1^k = \sum_{i=k}^{l}\mathbf{1}(y_{(i)} = 1) \quad (2.39)$$

When $y_{(k)} = 0, l_0^{k+1} = l_0^k - 1$ and $l_1^{k+1} = l_1^k$, when $y_{(k)} = 1, l_1^{k+1} = l_1^k - 1$ and $l_0^{k+1} = l_0^k$. This provides

$$Pr\{x_{(k)} \in R|E_{k-1}\}\begin{cases} \frac{n}{l_0^k}, & \text{if } y_{(k)} = 0 \\ \frac{n}{l_1^k}, & \text{if } y_{(k)} = 0 \end{cases}$$

Finally, posterior probabilities can be calculated for any sample x_i by carrying out the algorithm on the dataset that excludes the sample in question and its total.

We can simplify the posterior probability decomposition in Equation 2.35 by noting that the right-hand side of the equation corresponds to a weighted sum of labels. More specifically, distributing the multiplication terms $\Pr\{x_{(1)} \notin R\}$, $\Pr\{x_{(2)} \notin R|E_1\}$, $\Pr\{x_{(3)} \notin R|E_3\}$ and so on across the parantheses provides

$$\begin{aligned}Pr\{y=1\} &= Pr\{x_{(1)} \in R\}y_{(1)} + Pr\{x_{(1)} \notin R\}Pr\{x_{(2)} \in R|E_1\}y_{(2)} \\ &\quad +Pr\{x_{(1)} \notin R\}Pr\{x_{(2)} \notin R|E_1\}Pr\{x_{(3)} \in R|E_2\} + \ldots + \\ &\quad Pr\{x_{(1)} \notin R\}Pr\{x_{(2)} \notin R\}\ldots Pr\{x_{(\ell-1)} \notin R\}Pr\{x_{(\ell)} \in R|E\ell - 1\}y_{(\ell)} \qquad (2.40)\end{aligned}$$

or, in summation form,

$$\Pr\{y=1\} = \sum_{k=1}^{k^*} a_{(k)}y_{(k)}$$

where the weights $a_{(k)}$ are defined by

$$a_{(k)} = Pr\{x_{(k)} \in R\}\prod_{v=1}^{k-1}(1 - Pr\{x_{(v)} \in R\} \qquad (2.41)$$

for all $k = 1, 2, \ldots, k^*$. Note that this calculation can also be written as

$$\Pr\{y=1\} = \sum_{i=1}^{n} a_i y_i \qquad (2.42)$$

where a_i corresponds to the weight of y_i if it appears in the list of $y_{(k)}$ for $k = 1, 2, \ldots, k^*$, and is equal to 0 otherwise. Organizing $\{a_i\}$ and $\{y_i\}$ in column vectors a and y respectively, we can write

$$\Pr\{y=1\} = a^T y \qquad (2.43)$$

to calculate the probability with which a point x belongs to C_1 given the dataset $\{x_i, y_i\}$. The ability of quasi-supervised learning to estimate posterior probabilities for the points $\{x_i\}$ lies in carrying out the calculation above for a point $x = x_i$ while removing the pair (x_i, y_i) from the dataset used in the calculations. Using the notation above, this can be expressed by

$$\Pr\{x_i \in C_1\} = a_i^T y_i \tag{2.44}$$

where the vectors a_i and y_i corresponds to weights and point labels obtained by carrying out the posterior probability calculation over the reduced dataset following the removal of (x_i, y_i). Finally, letting

$$\pi_i = \Pr\{x_i \in C_1\} \tag{2.45}$$

and defining

$$A = [a_1 a_2 \ldots a_n] \tag{2.46}$$

we obtain the matrix form for the quasi-supervised learning algorithm as

$$\pi = Ay \tag{2.47}$$

where π denotes the column vector of posterior probabilities π_i, for $i = 1, 2, \ldots, n$.

2.2.2. Class Overlap Measures and the Selection of the Optimal Reference Set Size

Quasi-supervised learning algorithm basically aims to minimize overlap between two groups and good classification or clustering entail small class overlaps. We can define several class overlap measures using estimated posterior probabilities $f_0(x)$ and $f_1(x)$ (Karaçalı, 2010). The first one is log-likelihood ratio between two classes and represented with M_{LLR}. M_{LLR} class overlap measure can be defined as

$$M_{LLR}(x) = \log\left(\frac{f_0(x)}{f_1(x)}\right) \tag{2.48}$$

for all x with $f_0(x) \neq 0$ and $f_1(x) \neq 0$. The ratio of $f_0(x)/f_1(x)$ goes to 1 and M_{LLR} goes to zero when the sample x is located on the overlap region of two classes. The differences between posterior probabilities $f_0(x)$ and $f_1(x)$ provides an another overlap measure, M_{Diff}, defined as

$$M_{Diff} = f_0(x) - f_1(x), \tag{2.49}$$

so that, if the sample x is on located the overlap region posterior probabilities become closer and M_{Diff} goes to zero. Finally, a third overlap measure can be defined by Henze-Penroze affinity that computes the integral

$$\int_x \frac{2f_0(x)f_1(x)}{f_0(x) + f_1(x)} dx \tag{2.50}$$

for probability density function $f_0(x)$ and $f_1(x)$. This integral goes to 1 when two distribution are equal. The overlap measure $M_{HP-like}(x)$ can be formulated based on Equation 2.50 as follows:

$$M_{HP-like}(x) = f_0(x)f_1(x) \cong \frac{p(x|x \in C_0)p(x|x \in C_1)}{(p(x|x \in C_0) + p(x|x \in C_1))^2}. \tag{2.51}$$

The correct estimation of posterior probabilities also depends on the reference set size n. The reference set size must decrease the class-overlap while maintaining a small reference set size for better generalizability. A suitable cost function to determine the optimal reference set size is provided by the expression below to be minimized with respect to n, possibly via a line search:

$$E(n) = 4 \sum_{i=1}^{\ell} f_0(x)f_1(x) + 2n \tag{2.52}$$

The first term is related with class overlap and second term limits the reference set size n.

We have created a toy datasets to illustrate the QSL algorithm. In the first dataset, we created a two dimensional Gaussian mixture with two different classes C_0 and C_1. Then, we estimated the posterior probabilities of each sample belonging to C_0. The posterior probability values are represented using a heatmap on the scatter plot (Figure 2.15). The posterior probabilities on the far regions of the clusters are close to extreme probability values 1 and 0: If a sample is in C_0 and it is located far from the C_1 samples, its C_0 posterior probability close to 1. Also, the posterior probabilities on the samples located between the two clusters on the overlap region, are around 0.5 for both C_0 and C_1.

In the second case, we created two Gaussian distributed one dimensional groups with different number of samples. In Figure 2.16, we have demonstrated the log-likelihood ratio estimation using quasi-supervised algorithm. Results show that log-likelihood ratio estimation using quasi-supervised learning algorithm approaches to true values when the number of samples in the dataset increase. Since the number of sample is very low at the tails of the distributions, the estimated results saturate and begin to deviate from true values.

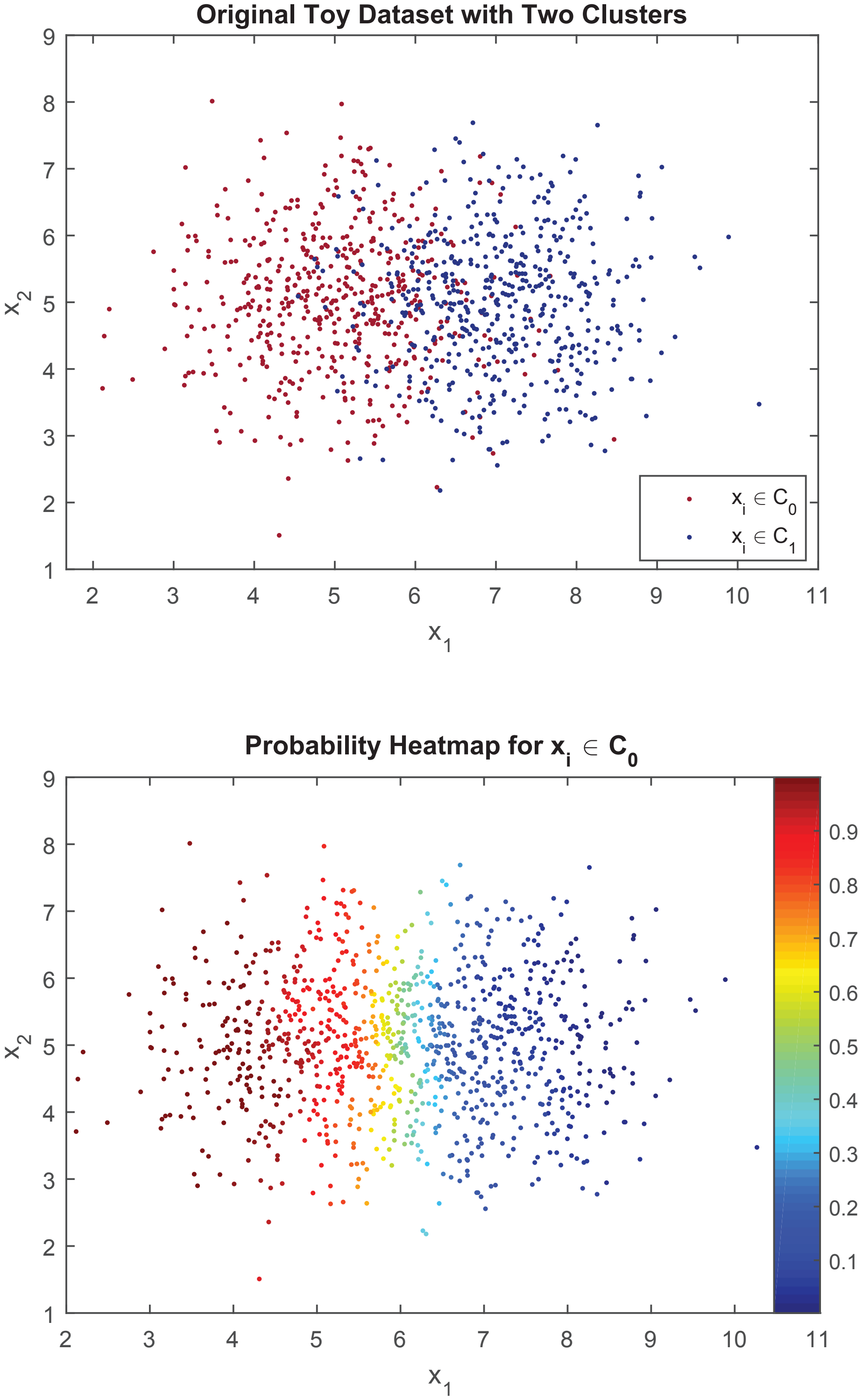


Figure 2.15. Quasi-supervised learning algorithm results for a toy problem.

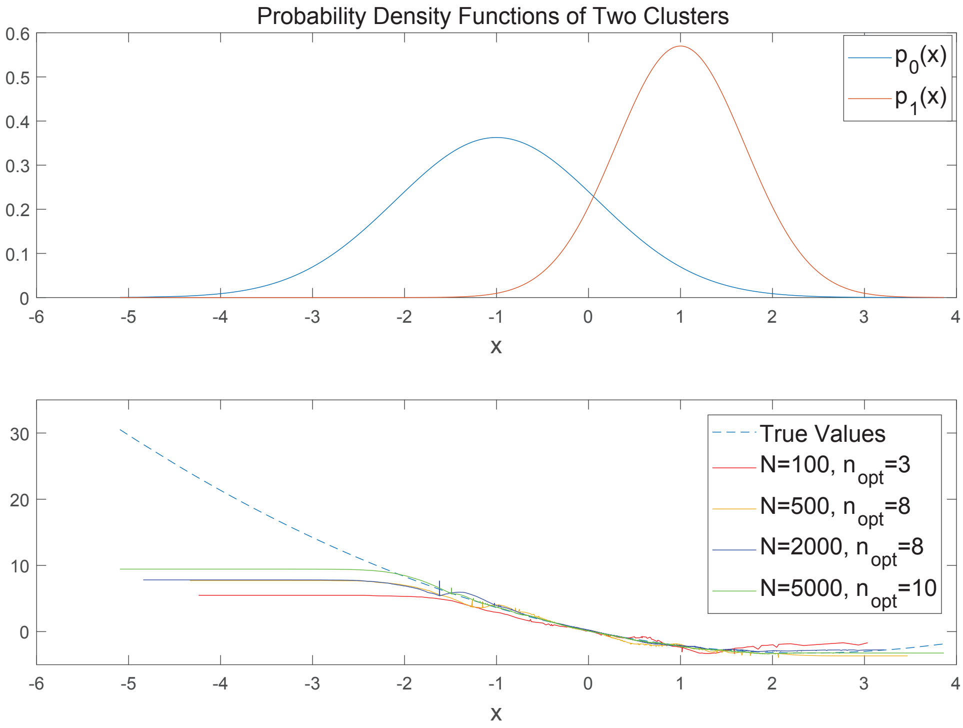


Figure 2.16. True and Estimated Results of Log-likelihood Ratio for Different Sample Size

2.3. Expectation-Maximization Algorithm

Expectation-maximization (EM) algorithm was first introduced by Dempster et. al in 1977 to find an approach for iteratively computation of maximum-likelihood estimate (Dempster et al., 1977). In EM framework, the group densities are unknown and the distribution parameters are estimated from cluster patterns (Jain et al., 1999). The expectation step estimates the likelihood function using observed data and the maximization step chooses the best parameters that maximize new likelihood function, and continuous to re-estimate the likelihood function until convergence.

The conventional EM algorithm aims to fit a specific distribution on mixture dataset (Shafer et al., 1976; Moon, 1996). Suppose we have observed data points x_i with $i = 1, 2, \ldots, \ell$ and we know that we have two or more groups with known distributional form with unknown pramaters. If we assume our data contains k different groups and θ_j is the data distribution parameter for j^{th} component in the mixture where $j = 1, 2, \ldots, k$, the expectation maximization algorithm carries out maximum likelihood estimation for each parameter. Under Gaussian form assumption, the parameter θ_j can be defined as a pairing of the mean μ_j and the covariance Σ_j of the Gaussian distribution for the corresponding component:

$$\theta_j = (\mu_j, \Sigma_j) \tag{2.53}$$

The likelihood function for each parameter θ_j can be expressed as

$$\begin{aligned} L_x(\theta_j; x_1, x_2, \ldots, x_\ell) &= f(x_1, x_2, \ldots, x_\ell | \theta_j) \\ &= \prod_{i=1}^{\ell} f(x_i | \theta_j) \end{aligned} \tag{2.54}$$

since the points are assumed to have been drawn independently. The maximum likelihood estimate for distribution parameter θ_j is given by:

$$\theta_j^{ML} = \arg\max_{\theta} \ell_x(\theta) \tag{2.55}$$

In many applications, using log-likelihood estimation is more practical. Log-likelihood function can be defined as

$$L_x(\theta) = \log \ell_x(\theta). \tag{2.56}$$

Since the logarithm is a monotonically increasing function, we can express the best choice for parameter θ_j via

$$\theta_j^{ML} = \arg\max_{\theta} L_x(\theta). \tag{2.57}$$

The EM approach calculates a responsibility value that describes the likelihood of each sample x_i to belong to the j^{th} component. The responsibility value $r_{i,j}$ can be described as (Guo et al., 2012)

$$r_{i,j} = \frac{p(x_i, \theta_j)}{\sum_{m=1}^{k} p(x_i, \theta_j)}. \tag{2.58}$$

The parameters θ_j are then revised in the subsequent maximization step using a maximum likelihood procedure that takes the responsibility values into account. A notable distinction between different expectation maximization procedures arises from the use of the responsibility values in the maximization step: In one alternative, the responsibility values can be used to associate each x_i with only one component by seeking the component achieving the maximum among $\{r(i, 1), r(i, 2), \ldots, r(i, k)\}$ for each i, and using only these points to estimate the corresponding model parameters. In the other alternative, the model parameters θ_j are estimated in a way that uses all points simultaneously, but in a way to be influenced more by the points x_i for which $r(i, j)$ are greater and less by the others.

In this thesis, we have combined expectation-maximization algorithm with QSL to create an automated clustering algorithm and applied it to automated gating of multicolor flow cytometry data. This method is described in next chapter.

2.4. Fast Algorithm for Joint Diagonalization with Non-orthogonal Transformations

Joint diagonalization of square matrices is an important problem and is needed in independent component analysis (ICA) and blind source separation (BSS) applications. There are some algorithms for joint diagonalization in the literature (Noble and Daniel, 1977; Golub and Van Loan, 2012; Bunse-Gerstner et al., 1993; Van Der Vorst and Golub, 2001). Ziehe et al. took these methods in hand detailed and developed a new joint diagonalization algorithm which is based on second order approximation of a cost function and called it as Fast Algorithm for Joint Diagonalization with Non-orthogonal Transformations (FFDIAG) (Ziehe et al., 2004).

Suppose that, we have a set $\{C_1, C_2, \ldots, C_K\}$ of real valued symmetric matrices, possibly covariance matrices, each of size $N \times N$. FFDIAG algorithm uses an iterative scheme to solve to follow the optimization problem

$$\min_{V \in R^{N\times N}} \sum_{k=1}^{K} \sum_{i \neq j} ((VC_k V^T)_{ij})^2 \tag{2.59}$$

where V is the transformation matrix that diagonalizes all matrices C_k jointly. The main constraint of this optimization problem is the invertibility of the matrix V to prevent convergence of the Equation 2.59 to the degenerate solution of zero. Invertibility can be enforced by carrying out the update of V in multiplicative form through another matrix W as

$$V^{(n+1)} \longleftarrow (I + W^{(n)})V^{(n)} \tag{2.60}$$

where I is the identity matrix, the update matrix $W_{(n)}$ has zeros on the main diagonal, and n denotes the iteration number. The update matrix $W^{(n)}$ is defined by

$$W_{i,j}^{(n)} = \frac{\sum_k E_{k_{i,i}}^{(n)}(D_{k_{i,i}}^{(n)} - D_{k_{j,j}}^{(n)})}{\sum_k (D_{k_{i,i}}^{(n)} - D_{k_{j,j}}^{(n)})^2} \tag{2.61}$$

where $D_k^{(n)}$ and $E_k^{(n)}$ contain the diagonal and off-diagonal elements of the matrices $C_k^{(n)}$ respectively (Ziehe et al., 2004), such that

$$C_k^{(n)} = D_k^{(n)} + E_k^{(n)} \tag{2.62}$$

Now, $I + W_{(n)}$ must be enforced to be invertible to ensure the invertibility of V. The Levi-Desplanques theorem states that *if an $n \times n$ matrix A is strictly-dominant, then it is invertible* (Horn and Johnson, 2012). This means that since the diagonal elements of $I + W_{(n)}$ are equal to 1, the matrix W must then satisfy the following constraint:

$$\max_i \sum_{i \neq j} |W_{ij}| = ||W^{(n)}||_\infty < 1 \tag{2.63}$$

This can be done by dividing $W_{(n)}$ by its infinity norm or its Frobenius norm when the last update matrix $W_{(n)}$ exceeds some fixed $\theta < 1$. The correction can be expresses via

$$W^{(n)} \longleftarrow \frac{\theta}{||W^{(n)}||_F} W^{(n)} \tag{2.64}$$

where $||W_{(n)}||_F$ is the Frobenius norm of $W_{(n)}$ and it's equal to trace of $W^{(n)}W^{(n)^H}$

$$||W_{(n)}||_F = \sqrt{tr(W^{(n)}W^{(n)^H})} \tag{2.65}$$

and $W^{(n)^H}$ denotes the conjugate transpose of $W^{(n)}$. Then FFDIAG algorithm iteratively diagonalizes the covariance matrices by updating them as follows:

$$C_k^{(n+1)} \longleftarrow (I + W^{(n)})C_k^{(n)}(I + W^{(n)})^T \tag{2.66}$$

The pseudo-code describing the FFDIAG method is outlined in Algorithm 1 below as presented by Ziehe et al (Ziehe et al., 2004):

```
INPUT: $C_k$ {Matrices to be diagonalized}
$W^{(1)} \longleftarrow 0, V^{(1)} \longleftarrow I, n \longleftarrow 1$
$C_k^{(1)} \longleftarrow V^{(1)}C_kV^{(1)^T}$
repeat
    compute $W^{(n)}$ from $C_k^{(n)}$
    if $||W^{(n)}||_F > \theta$ then
        $W^{(n)} \longleftarrow \frac{\theta}{||W^{(n)}||_F} W^{(n)}$
    end if
$V^{(n+1)} \longleftarrow (I + W^{(n)})V^{(n)}$
$C^{(n+1)_k} \longleftarrow (I + W^{(n)})C_k^{(n)}(I + W^{(n)})^T$
$n \longleftarrow n + 1$
until convergence
OUTPUT: $V, C_k$
```

CHAPTER 3

MODEL-FREE EXPECTATION MAXIMIZATION CLUSTERING

We have discussed flow cytometry gating in Chapter 2. To iterate briefly, identification of cell sub groups and analysis of flow cytometry data are performed by pathologists manually on two or three dimensional scatter plots. This process, however, is laborious and time-consuming. In addition this, there are concerns over reproducibility of the gating results even by the same expert on the same flow data (Lo et al., 2008). To overcome these problems, automated gating methods have been developed for multi-color flow cytometry data analysis. As described in Chapter 2, several methods have been proposed to model cell population characteristics. The proposed solutions have different challenges. Some of them assume all subpopulations have specific distributions such as Gaussian, t-distribution etc. and this is not realistic assumption. Fiting a model on an unknown data is not expected to give robust solutions. Also, the most of of them identify the number of clusters by running algorithm several times with different number of clusters and choose the best cluster number by optimizing the some information criteria (AIC, BIC or Entropy). We need to develop a fully automatic clustering method that can both identify the number of clusters and determine the distinct clusters without any knowledge about the dataset. To this aim, we combined quasi-supervised learning algorithm with expectation-maximization routine and we called it *”model-free expectation-maximization algorithm (MFEM)”*. MFEM clustering identifies cell sub groups in a multi-color flow cytometry dataset automatically.

The model-free expectation-maximization clustering algorithm is basically a binary divisive hierarchical clustering algorithm for all datasets type not only for flow cytometry data. It starts by dividing the whole dataset into two groups using an expectation maximization procedure that relies on a model-free calculation of the group posterior probabilities using the quasi-supervised learning algorithm (Köktürk and Karaçalı, 2014). The method continues to a binary division on the subgroups obtained by previous divisions until it achieves a stopping criterion. It controls further division in each step and automatically identify the number of clusters.

The main contribution of this method is to provide cell sub groups without making any model assumptions or number of cluster estimation. Technical details of the algorithm re discussed next. We applied this clustering algorithm to both synthetically created Gaussian mixtures and real multi-color flow cytometry datasets in Section 3.3. In Discussion Section, we summarize our obtained results and discuss possible improvement strategies for this clustering algorithm.

3.1. Methodology

Let assume that we have a dataset X with elements $\{x_i\}$ where $i = 1, \ldots, N$ and it has k clusters. The model-free expectation maximization clustering algorithm begins with randomly assigning sample points to the clusters C_0 and C_1. After randomly assigning labels, the algorithm estimates the posterior probabilities of the individual samples belonging to each cluster. These posterior probabilities are then used as the responsibility values of the expectation-maximization routine, and class labels are re-assigned according to the maximum likelihood rule.

Following the class label update, the algorithm re-calculates the posterior probabilities with the new and again update the class labels. This process is repeated until convergence or a maximum number of iteration. In our algorithm, convergence is defined as the observation of a label change in no more than one percent of the whole samples at last iteration or 100 iterations.

Our clustering algorithm can be summarized in an expectation-maximization perspective as follows:

First step : Expectation step

The posterior probabilities of C_0 and C_1 are computed for each sample x_i using the quasi-supervised learning algorithm:

$$f_0(x_i) = Pr\{y_i = 0\}$$
$$f_1(x_i) = Pr\{y_i = 1\}$$

Second step : Maximization step

The class label of each sample x_i is updated and new clusters C_0 and C_1 are formed according to the maximum a posteriori classification rule via

$$C_0 \leftarrow \{x_i | f_0(x_i) \geq 0.5\}$$
$$C_1 \leftarrow \{x_i | f_1(x_i) < 0.5\}$$

This procedure aims to create two distinct clusters that are as separate as from each other possible. The goodness of the resulting cluster is evaluated by an overlap measure $c(C_0, C_1)$ defined by

$$c(C_0, C_1) = \frac{1}{N_0} \sum_{x_i \in C_0} f_1(x_i) + \frac{1}{N_1} \sum_{x_i \in C_1} f_0(x_i) \tag{3.1}$$

where N_0 and N_1 denote the number of sampled assigned to class C_0 and C_1, respectively. Note this measure calculates a balanced estimate of the overlap between C_0 and C_1, since it corresponds to the normalized sum of C_0 presence in C_1 samples and C_1 presence in C_0 samples.

Since we have developed a binary hierarchical clustering scheme, we need to control the algorithm's progression before it invokes further divisions. To this end, we have used the overlap measure $c(C_0, C_1)$ as a division cost and compared the division cost obtained from a clustering with the division cost obtained from the clustering of its parent cluster. If the division cost is greater than the parent cluster division cost, the division process stops and division is rejected; otherwise, the division is accepted and the algorithm continues to divide each children into sub clusters.

We explain the methodology for this algorithm in a block diagram in Figure 3.1. Also in Figure 3.2, we show the binary division of a Gaussian mixture with 3 clusters on a tree model. At the top, we have the parent data containing all three clusters. Our algorithm divides this data into two groups, shown in red and blue, with a division cost of 0.0908. Then the child clusters, namely Estimated Cluster 1 and Estimated Cluster 2, are divided into two clusters of their own and the respective division costs are calculated. Since Estimated Cluster 1 does not contain sub clusters, its division cost is greater than parent data division cost, leading to the rejection of this division. On the other hand, Estimated Cluster 2 has two distinct clusters. This provides a division cost of 0.0570 less than that of its parent's division leading to the acceptance of the division. As a find verification of all clusters at the end of the algorithm we have checked whether the union of any two cluster forms a compact cluster or not. To do that, we have calculated the overlap measure between all pairs, then merged the clusters for which the overlap measure is larger than the first overlap measure.

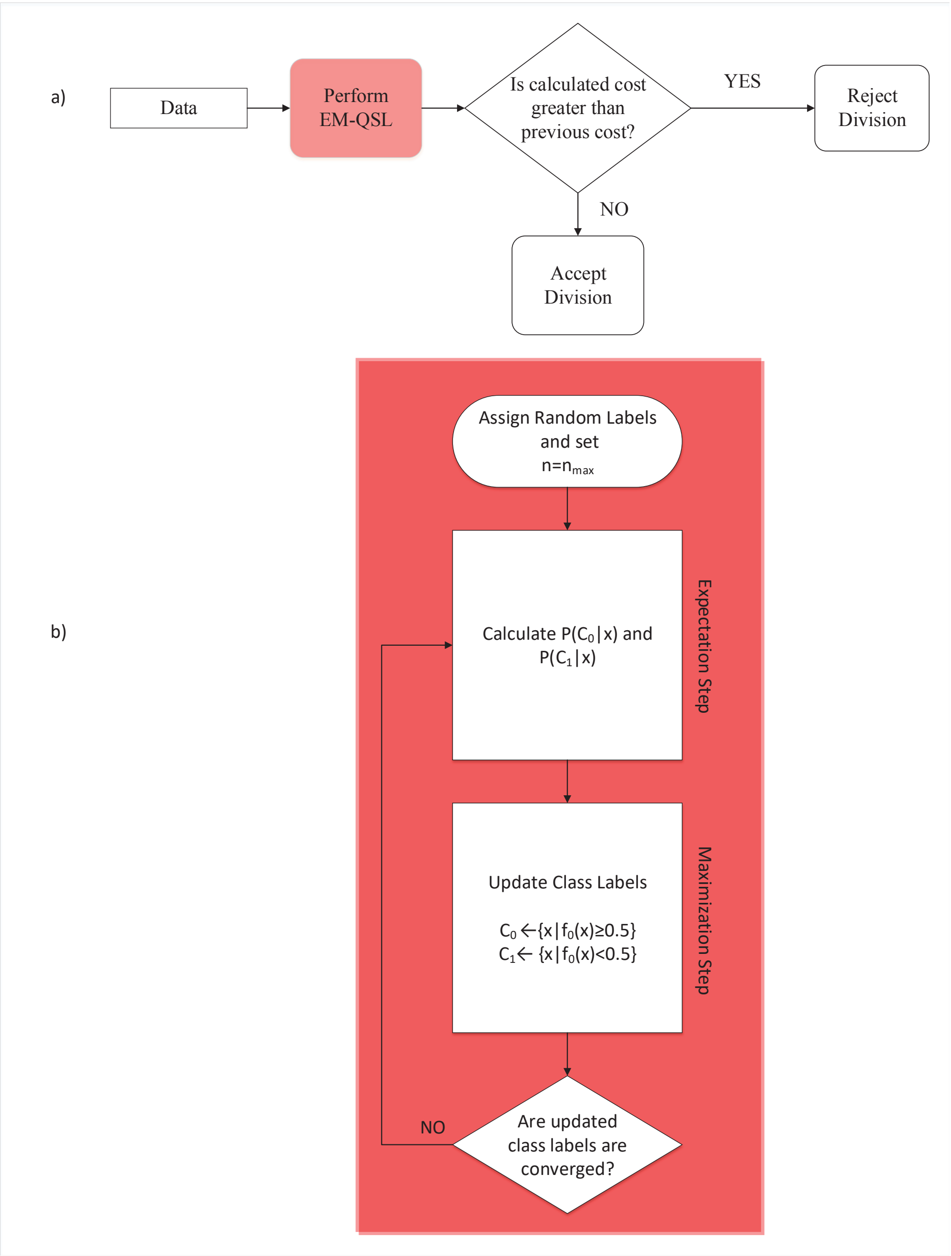

a)
Data
Perform EM-QSL
Is calculated cost greater than previous cost?
YES
Reject Division
NO
Accept Division
b)
Assign Random Labels and set $n=n_{max}$
Calculate $P(C_0|x)$ and $P(C_1|x)$
Expectation Step
Update Class Labels
$C_0 \leftarrow \{x|f_0(x) \geq 0.5\}$
$C_1 \leftarrow \{x|f_0(x) < 0.5\}$
Maximization Step
NO
Are updated class labels are converged?

Figure 3.1. Block Diagram of Binary Divisive Clustering using Model-free Expectation-Maximization Algorithm (a). The Model-free Expectation-Maximization Algorithm is detailed in (b).

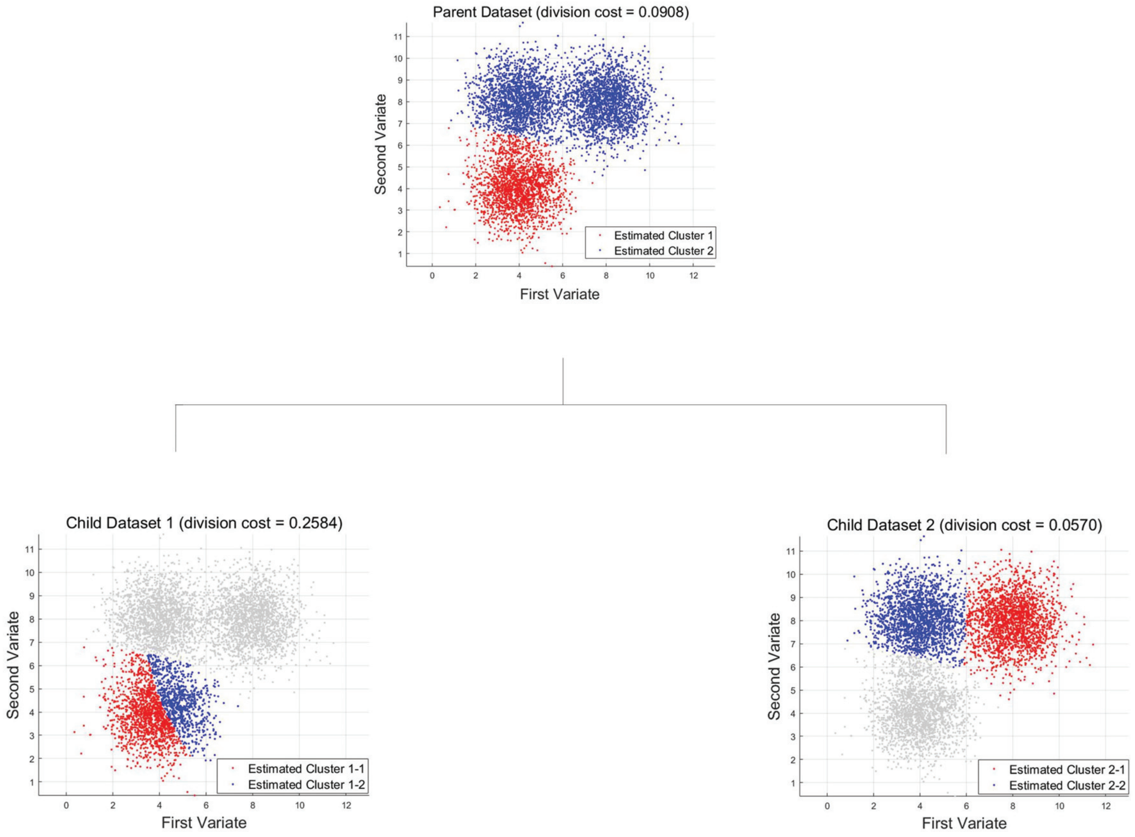


Figure 3.2. Tree Model of the clustering results for the toy problem involving a Gaussian mixture with three components.

3.2. Performance Evaluation

Clustering aims to divide dataset X into K meaningful groups. There are two basic questions that need to be answered in a clustering application. First one is *how many clusters are there in dataset* and the other one is *how real is the clustering itself* (Maulik and Bandyopadhyay, 2002). There are some performance metric measures that measure how well your clustering algorithm. A contingency table is a useful tool for measuring the performance of clustering. The entries of contingency table are generally frequency counts between real classes and estimated clusters.

Table 3.1. Contingency Table

		Estimated Clusters			
		$\widetilde{C_1}$	$\widetilde{C_2}$	$\dots$	$\widetilde{C_K}$
True Classes	C_1	c_{11}	c_{12}	$\dots$	c_{1K}
	C_2	c_{21}	c_{22}	$\ddots$	c_{2K}
	$\vdots$	$\vdots$	$\ddots$	$\ddots$	$\vdots$
	C_K	c_{K1}	c_{K2}	$\dots$	c_{KK}

Let C be a contingency table whose rows represent the true class labels while columns represent the estimated clusters. The entries of the contingency table are the number of samples that are assigned to cluster j while they were originally located in class i. The entries are denoted by $c_{11}, c_{12}, \dots, c_{KK}$, where first subscript represents the true class and the second subscript represents the estimated cluster.

A popular performance measure often used to evaluate automatic flow cytometry gating is the F-measure, defined as the harmonic mean of the *precision* and *recall* via

$$F_{measure} = \frac{2 \times Pr \times Re}{Pr + Re} \tag{3.2}$$

where precision (Pr) for a cluster equals the number of cells assigned to that cluster divided by the total number of cells assigned to that cluster, and Recall (Re) for a cluster is the number of samples that are correctly assigned to that cluster divided by the total number of samples of that cluster (Aghaeepour et al., 2013). For each true-estimated cluster pair, the corresponding

F-measure can be written as

$$F(C_i, \widetilde{C}_j) = \frac{2 \times Pr(C_i, \widetilde{C}_j) \times Re(C_i, \widetilde{C}_j)}{Pr(C_i, \widetilde{C}_j) + Re(C_i, \widetilde{C}_j)} \tag{3.3}$$

For each true cluster C_i, a set of F-measures against every predicted cluster $\widetilde{C}_j$ is calculated, and the best match with the highest F-measure is chosen and paired with C_i is reported. The sum of the highest scores for all true clusters produces a combined F-measure, defined as

$$F(C, \widetilde{C}) = \sum_{C_i \in C} \frac{C_i}{N} \max_{\widetilde{C}_j \in \widetilde{C}} \{F(C_i, \widetilde{C}_j)\} \tag{3.4}$$

3.3. Dataset & Results

We have applied our clustering method on both synthetically created Gaussian mixtures and real multi-color flow cytometry datasets. Real multi-color flow cytometry datasets are publicly available and obtained from FlowCAP-I Challange (Aghaeepour et al., 2013). We used Diffuse Large B-cell lymphoma dataset (DLBCL) and Hematopoietic Stem Cell Transplant (HSCT) for testing our clustering methodology.

3.3.1. Synthetically Generated Gaussian Mixture Dataset

Firstly, we have created a *toy* dataset with 3 distinct clusters, each modeled using two-dimensional Gaussian distribution with identity covariances but different means located at coordinates $[4\ 8]^T$, $[4\ 4]^T$, $[8\ 4]^T$, respectively. Samples were drawn from this mixture using different priors and different number of samples, denoted by N_1, N_2 and N_3, respectively. The dataset was clustered using both the proposed method and the conventional expectation maximization routine. The same binary division scheme was followed in both strategies.

Clustering performance for different sample size mixtures are presented in Table 3.2. Results show that our proposed methods performance is very close to the conventional expectation maximization algorithm. Since the mixture components are Gaussian, it is not surprising that the conventional expectation maximization algorithm is successful in identifying the Gaussian components.

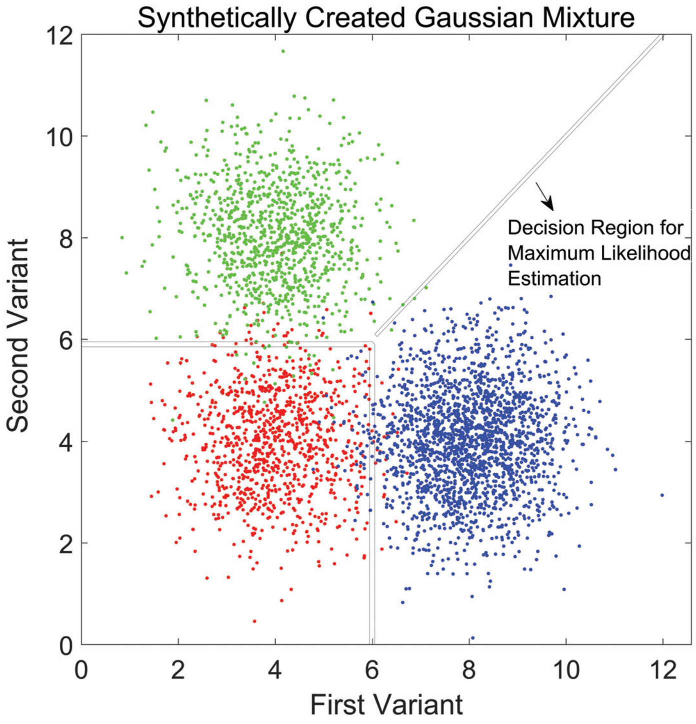


Figure 3.3. Synthetically Created Gaussian Mixture and Respective Class Labels

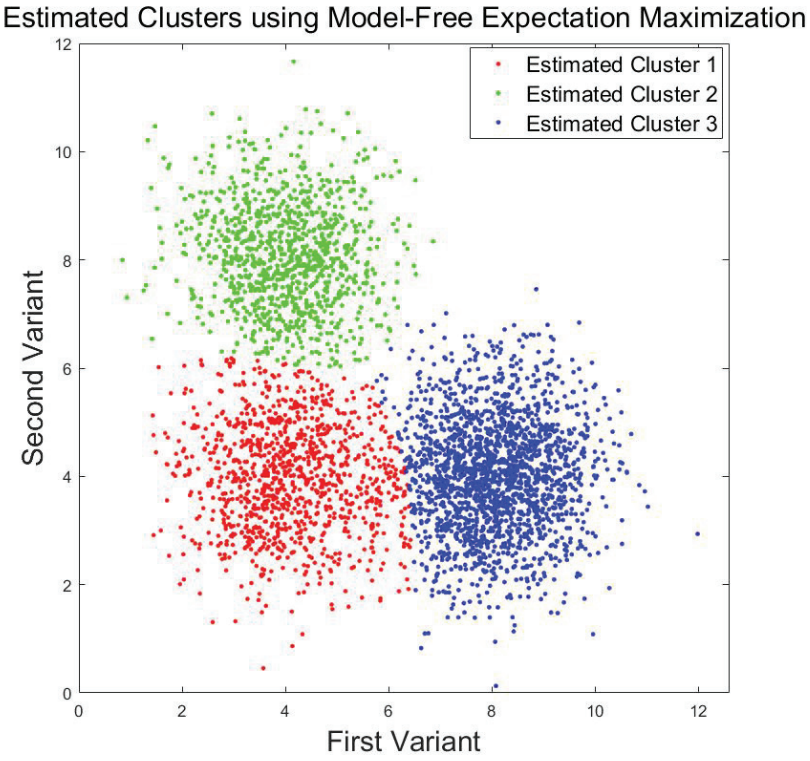


Figure 3.4. Estimated Clusters using MFEM on Synthetically Created Gaussian Mixture

Table 3.2. F-measures for the algorithm performance on synthetic dataset for different sample sizes in clusters

N_1	N_2	N_3	MFEM	conventional EM
500	500	1000	0.9622	0.9704
500	1000	2000	0.9633	0.9692
1000	500	1000	0.9694	0.9671
1000	1000	1000	0.9594	0.9682
2000	500	2000	0.8573	0.9746

3.3.2. Diffuse Large B-cell lymphoma (DLBCL) Dataset

Lymphoma is a type of cancer and it occurs when lymphocytes divide in an uncontrolled rate. The lymphoma cells look much larger compared to healthy lymphocytes. Diffuse large B-cell lymphoma (DLBCL) is a fast-growing type of lymphoma, so early diagnosis is very important for treatment.

In this study, we have used multi-color flow cytometry dataset provided by Flow Cap-I Challange Committee. The dataset contains 12369 samples in three expert-marked clusters. The manual gating procedure used to label the cells involved creating two-dimensional scatter plots of all possible parameter (fluorochrome) pairs (FL1vs2, 1vs3, 1vs4, 2vs3, 2vs4, 3vs4) and choosing the one in which the distinctions between the different clusters is most conspicuous for manual gating. In accordance with this approach, we have also used the same parameter pairs (FITC and PE Channels) to carry out the clustering experiments.

We have applied both our proposed binary divisive methodology using expectation maximization with quasi-supervised learning and the conventional expectation maximization algorithm. Figure 3.5 shows the data with manual gating representing the labels provided by the expert knowledge based labels. Figure 3.6 shows the clustering results of our proposed methodology and our algorithm identified two large clusters and our algorithm f-measure is determined as 0.9051. Also, conventional expectation maximization algorithm captured two large clusters too, and algorithm f-measure is 0.9040. The inability of both strategies to identify the third cluster is linked with the size of cluster: In this dataset, the third cluster has only 25 samples and located between two larger clusters. As a result, there is no statistically significant information for automatically identifying this smallest cluster.

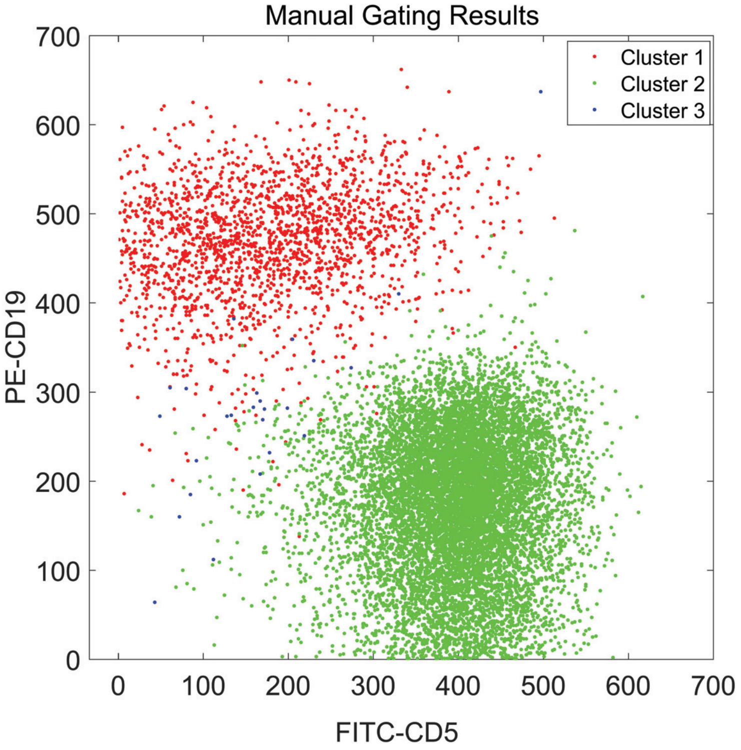


Figure 3.5. Manual Gating Results for The Diffuse Large B-cell Lymphoma (DLBCL) Dataset

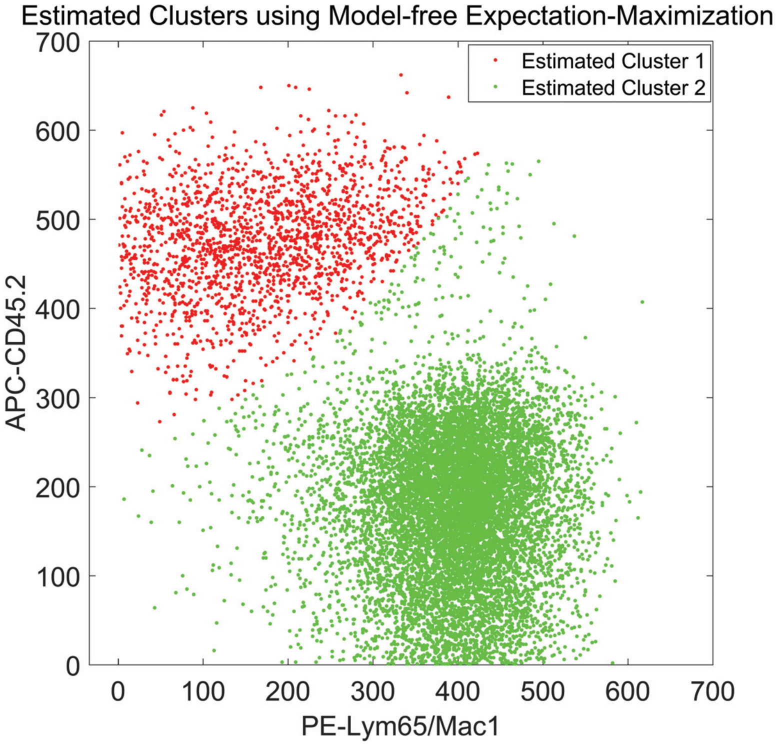


Figure 3.6. Automated Gating Results for The Diffuse Large B-cell Lymphoma (DLBCL) Dataset Using MFEM

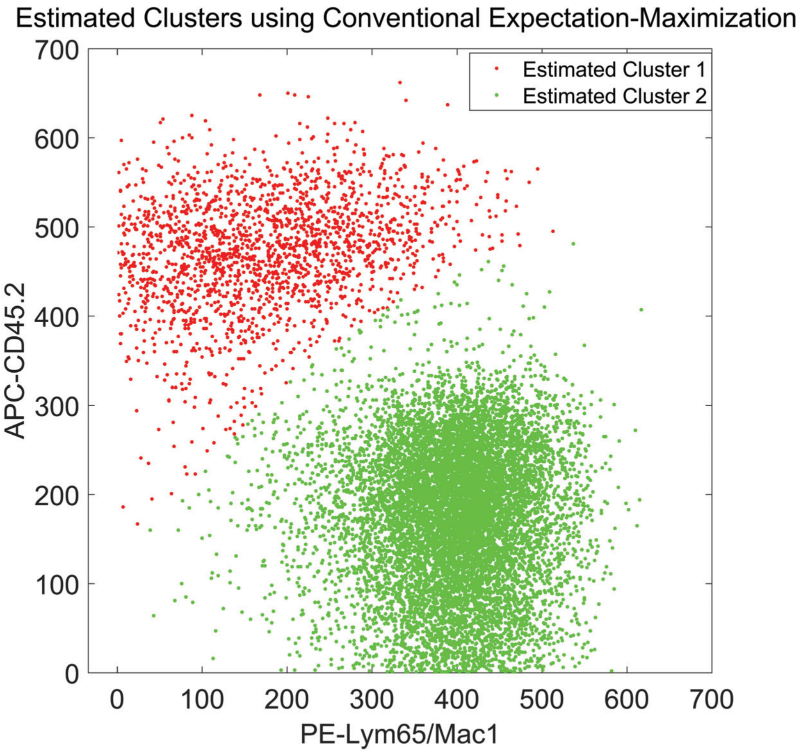


Figure 3.7. Automated Gating Results for The Diffuse Large B-cell Lymphoma (DL-BCL) Dataset Using Conventional Expectation Maximization

3.3.3. Hematopoietic Stem Cell Transplant (HSCT) Dataset

Hematopoietic stem cell transplantation (HSCT) is a treatment that involves intravenous infusion of stem cells for lymphoma, leukemia, immune-deficiency illnesses, congenital metabolic defects etc (Couri et al., 2009). It was identified the most efficient approach for some lymphohematopoietic neoplasms and for some solid tumors as well as non-malignant disorders (Voltarelli, 2000). Flow cytometry has an important role in monitoring the treatment.

In this experiment, we have used a multi-color flow cytometry dataset associated with a mouse hematopoietic stem cell transplantation provided by Flow Cap-I Challenge Committee that contains 8914 samples in four expert marked clusters. Figure 3.8 shows the dataset along with the labels. Figure 3.9 shows the clustering results of our proposed methodology; it identified three of the four clusters with a f-measure of 0.8087. The conventional expectation maximization algorithm determined only two clusters, at a level of f-measure is 0.5947. Carrying out the original expectation maximization algorithm outside of the binary division framework assuming four clusters failed since the algorithm could not capture the smallest cluster. However, assuming three clusters produced a better clustering with a f-measure level of 0.9706. In parallel with the earlier dataset the missing cluster was the one with a small size: the fourth cluster has only 100 samples and located between two larger clusters. Thus, there is no statistically significant information for automatically identifying this small cluster.

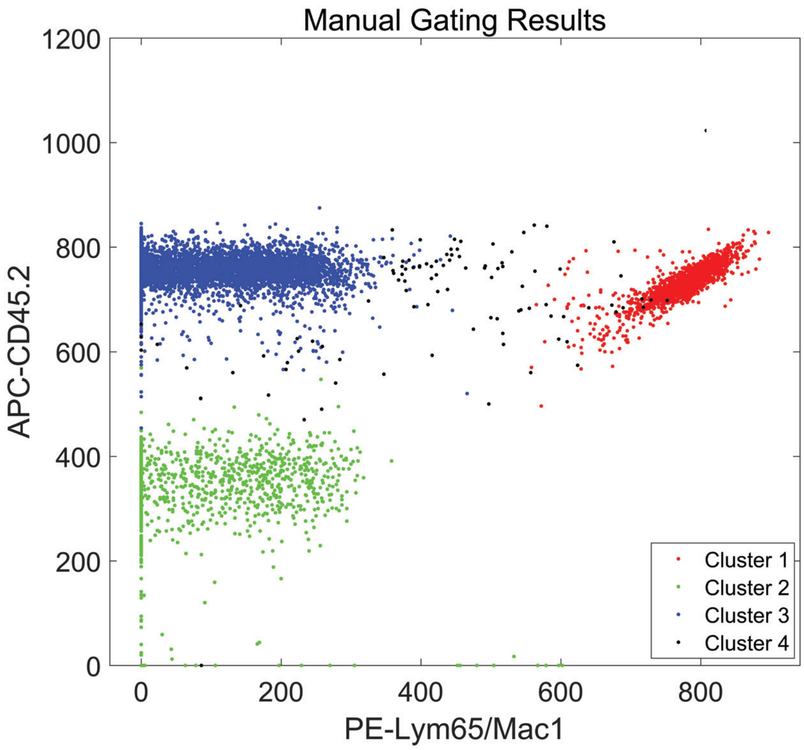


Figure 3.8. Manual Gating Results for The Hematopoietic Stem Cell Transplant (HSCT) Dataset

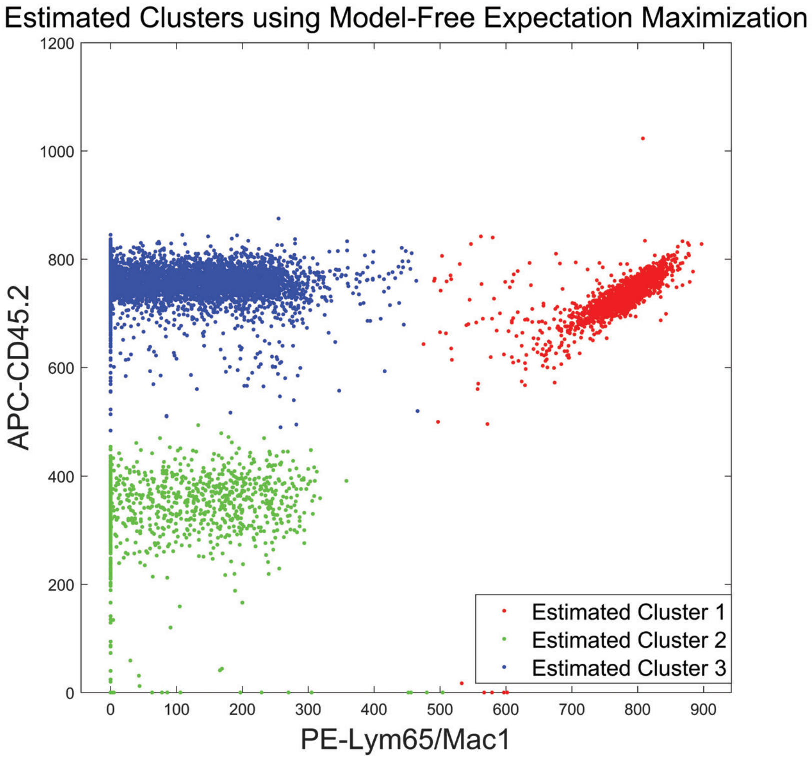


Figure 3.9. Automated Gating Results for Hematopoietic Stem Cell Transplant (HSCT) Dataset using MFEM

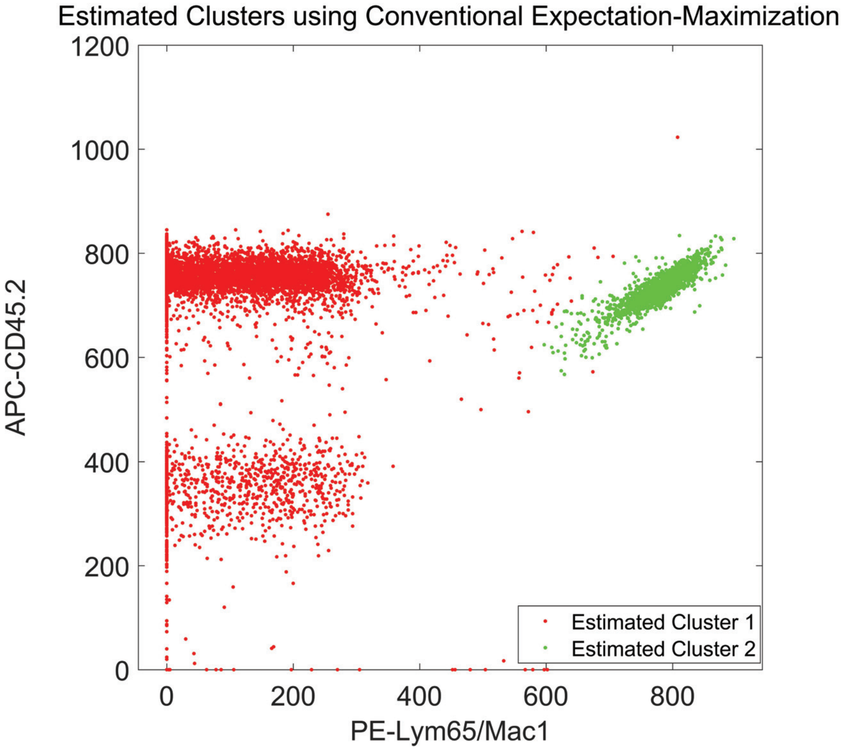


Figure 3.10. Automated Gating Results for Hematopoietic Stem Cell Transplant (HSCT)) Dataset Using Conventional Expectation Maximization

3.4. Discussion

We developed a novel clustering algorithm based on a recursive binary division scheme that does not require any knowledge on the number of clusters or any parametric model assumption. Model-free expectation maximization uses the quasi-supervised learning algorithm for posterior probability estimation and avoids fitting distributions of unknown data. It begins by randomly assigning class labels (C_0 and C_1) to the dataset and iterates until the posterior probability of the classes match the class assignments of the samples. This procedure divides each cluster into two daughter clusters. Furthermore, it controls further divisions using a division cost that evaluates how separate the resulting clusters are. In the proceed, it automatically identifies the number of clusters and determines which sample belongs to which cluster.

The performance of the clustering algorithm is measured using F-measure parameter on synthetically created datasets and real multi-color flow cytometry datasets. The results show that the proposed model-free expectation-maximization algorithm has the ability to identify the number of clusters, through the performance depends on the distinctness of the clusters and the number of samples in the clusters: In particular, the algorithm misses small clusters with few samples, since there is insufficient statistical evidence to warrant their identification. Furthermore, while the MFEM algorithm can capture clusters on low dimensional datasets (2D or 3D) accurately, the performance decreases with increasing dimensionality.

In order to address these issues, we have developed a more robust clustering method, namely the annealing-based model-free expectation-maximization (ABMFEM). In the next chapter, we explain annealing-based model-free expectation-maximization clustering in detail and we demonstrate its performance on the same synthetically created dataset as well as the DLBCL dataset using all five fluorochrome channels and HSCT dataset using all six fluorochrome channels.

CHAPTER 4

ANNEALING-BASED MODEL-FREE EXPECTATION-MAXIMIZATION CLUSTERING

In the previous chapter, we have introduced our binary divisive hierarchical clustering algorithm, model-free expectation-maximization clustering. Annealing-based model-free expectation-maximization (ABMFEM) is an improved version of the model-free expectation-maximization algorithm (Köktürk and Karaçalı, 2016). Physical annealing process is the process of heating up a solid until it melts and then, the solid is cool down slowly until the temperature of the enviroment. The particles reaches the ground state energy level (Kirkpatrick et al., 1983). In each cooling step, temperature remains constant for a time and solid reaches thermal equilibrium. In liquid phase, all particles are located randomly, particles forms a highly structured lattice in ground state and this provides the minimizing energy. Simulated annealing algorithm is the adopted form of the physical annealing process to clustering problems (Selim and Alsultan, 1991). The simulated annealing is a powerful optimization technique to find the global minimum of a function, in clustering problems, we can think ”particles” are data variables and ”energy” is the objective function wanted to minimize (Brown and Huntley, 1992). In this perspective, we optimize model-free expectation-maximization algorithm using the simulated annealing. Its iterative perspective gives an opportunity to create more flexible decision regions and this improves the algorithm performance. In this chapter, we introduce the details of the annealing based model-free expectation-maximization algorithm and the results obtained on the same synthetically generated and real multi-color flow cytometry datasets described in the previous chapter.

4.1. Methodology

The proposed method begins with an initial random assignment of points into two clusters C_0 and C_1, followed by the model-free expectation maximization cycle that begins with a large value for the reference set size parameter n and computes the posterior probability of C_0 and C_1 at each sample. The algorithm proceeds by re-assigning the points to the cluster whose posterior is larger and iterates until convergence. After convergence, the procedure is re-applied to the data starting with the latest cluster assignments using a smaller n. The optimal cluster assignments are selected by tracking the cost function in equation 2.52 as n

decreases to 1, and identifying the level for which $E(n)$ is minimal (Köktürk and Karaçalı, 2016). Updating class labels using larger reference sets achieves flexible decision regions between clusters. This feature of the algorithm gives better clustering performance on high dimensional datasets.

The block diagram of the proposed method is shown in Figure 4.1. As seen from the block diagram, this method uses binary divisive scheme described in the previous chapter for each value of the n, initialized with the cluster assignments of the previous n.

The modified expectation maximization procedure that forms the basis of the proposed clustering method is summarized below:

```
for $i = n_{max} : -1 : 1$ do
    Expectation Step:
        Calculate $P(C_0|x)$ and $P(C_1|x)$
    Maximization Step:
        Update class labels
        $C_0 \leftarrow \{x|f_0(x) \geq 0.5\}$
        $C_1 \leftarrow \{x|f_1(x) < 0.5\}$
end for
```

The main difference between proposed method and the earlier model-free expectation-maximization is, annealing over the reference set size n: simulated annealing aims to find the global minimum of a cost function by decreasing the energy level of a system gradually as it converges to the desired solution (Kirkpatrick et al., 1983). In the proposed method, n represents the system energy, as large n produces a more flexible learning system, and $E(n)$ measures the complexity of the clustering obtained for a given n. At the level where $E(n)$ is minimal and the clusters exhibit smallest overlap the algorithm produces the best clustering result.

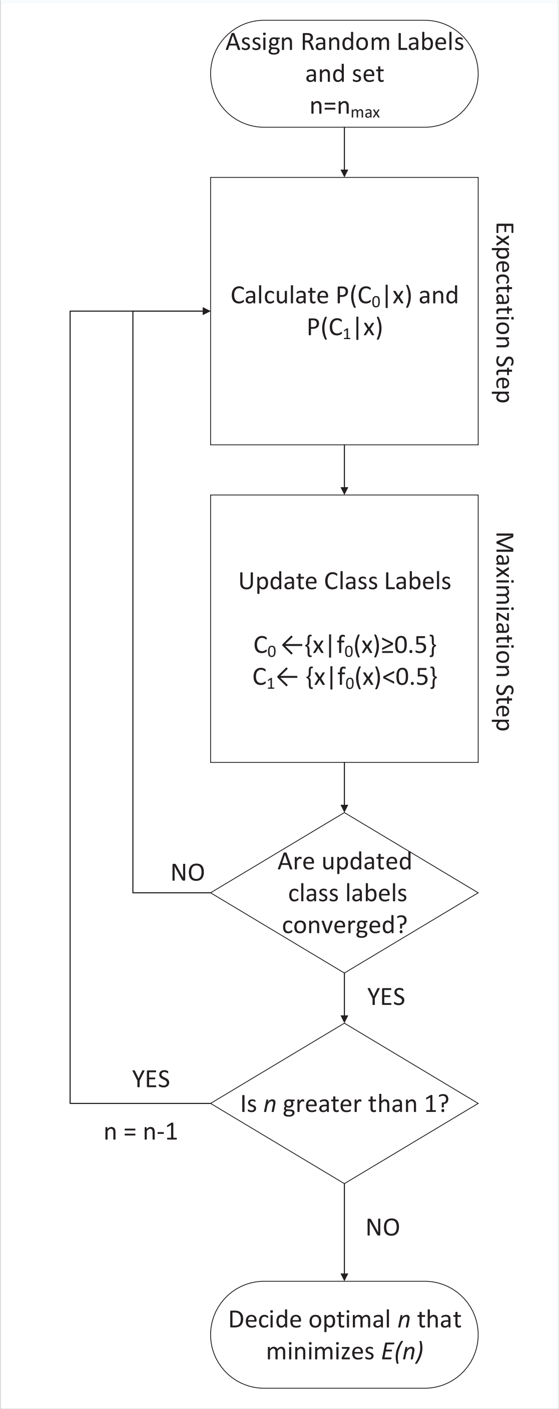

Assign Random Labels and set n=n_max
Calculate P(C_0|x) and P(C_1|x)
Expectation Step
Update Class Labels
C_0 ← {x|f_0(x)≥0.5}
C_1 ← {x|f_0(x)<0.5}
Maximization Step
NO
Are updated class labels converged?
YES
YES
n = n-1
Is n greater than 1?
NO
Decide optimal n that minimizes E(n)

Figure 4.1. Block Diagram of the Annealing Based Model-Free Expectation Maximization Algorithm

4.2. Dataset & Results

The proposed method was applied to the same datasets described in the previous chapter and results compared with those of model-free expectation-maximization algorithm. Note that, in the previous chapter we only used two-dimensional datasets since model-free expectation-maximization algorithm could not capture the clusters in other dimensions. But now, we used all dataset dimensions to evaluate our algorithm's performance.

4.2.1. Synthetically Generated Gaussian Mixture Dataset

We have used Gaussian mixture dataset that described previously. As a reminder, it had 3 distinct clusters, each modeled using two-dimensional Gaussian distributions with unit covariances but different means, at $[4\ 8]^T$, $[4\ 4]^T$, and $[8\ 4]^T$ respectively. Samples were drawn from this mixture using different priors. For each cluster, the number of samples are denoted by N_1, N_2 and N_3, respectively.

The division of this dataset using annealing-based model-free expectation-maximization algorithm is illustrated on Figure 4.2 along with the respective cost functions. The plots represent the final data labels were determined with the optimal n that minimized the cost function $E(n)$. The comparative accuracy table is presented on Table 4.1. The numbers represent average accuracies for the corresponding algorithms over 20 independent repeats. The results show that the annealing based approach provides an improvement of the clustering accuracy achieved by the earlier model-free expectation-maximization algorithm.

Table 4.1. F-measures for the algorithm performance on synthetic dataset for different sample sizes in clusters

N_1	N_2	N_3	MFEM	ABMFEM	Conventional EM
500	500	1000	0.9622	0.9659	0.9704
500	1000	2000	0.9433	0.9718	0.9692
1000	500	1000	0.9694	0.9701	0.9671
1000	1000	1000	0.9594	0.9646	0.9682

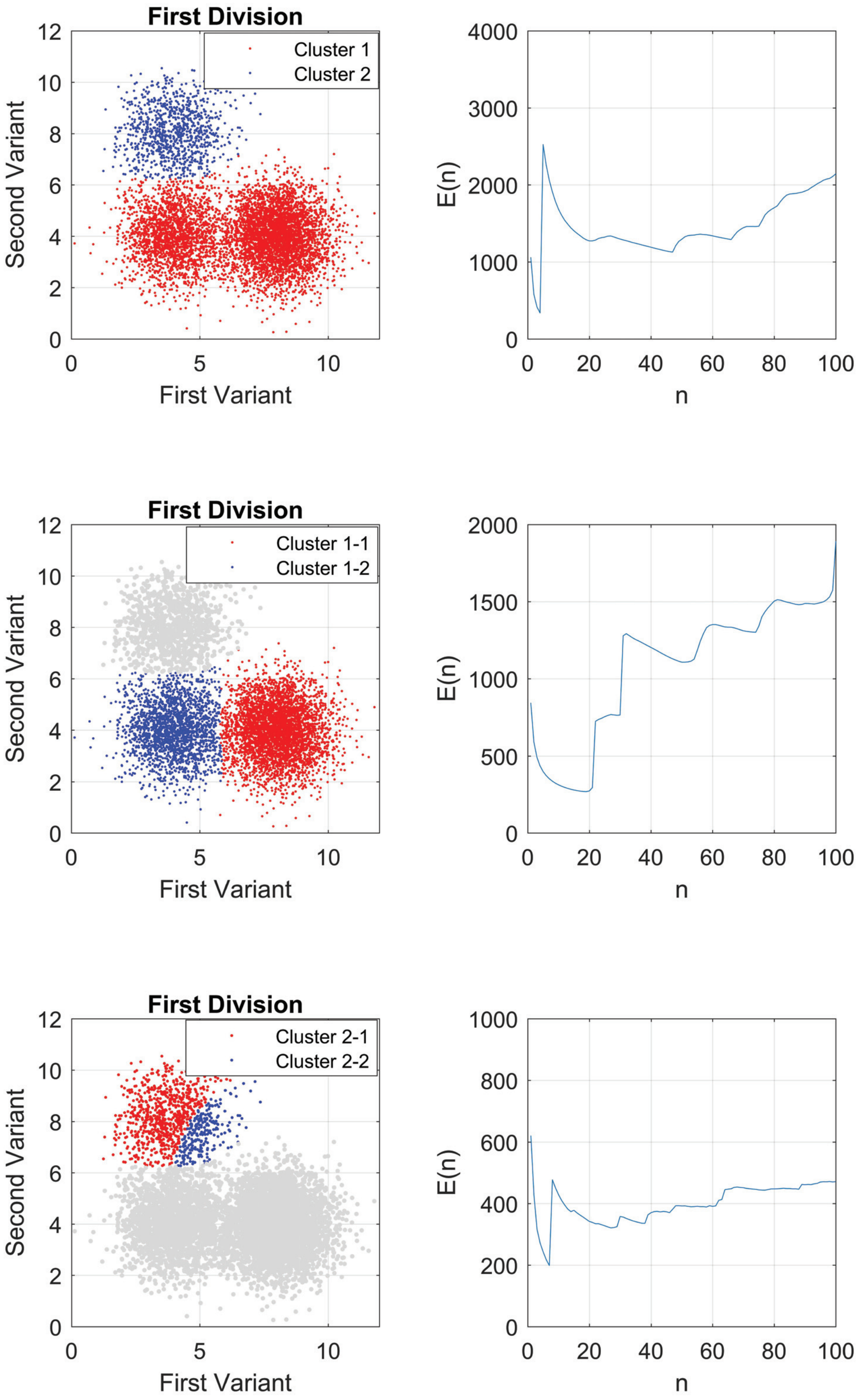


Figure 4.2. The data division is illustrated on the first column and respective $E(n)$ functions are given in the second column. The data is first divided into two clusters (upper row) and division cost is determined. Then, the procedure is applied on the daughter clusters (second and third rows).

4.2.2. Diffuse Large B-cell lymphoma (DLBCL) Dataset

We have used the same DLBCL dataset described in Section 3.3.2. However, in contrast to the earlier analysis that used only FITC and PE channels we used all five fluorochrome channels of the dataset since annealing-based model-free expectation-maximization clustering can capture distinct clusters in high dimensional datasets. The proposed method improved the clustering performance with an accuracy of 0.9959, while model-free expectation-maximization algorithm performance was 0.9051 in two dimensions and the conventional expectation maximization algorithm performance was 0.9040.

Note that the proposed algorithm also could not detect the third cluster that had only 25 samples, and this signifies the absence of statistical significance of the small cluster. But it identified the other two clusters samples with an increased overall accuracy. We have demonstrated annealing-based model-free expectation-maximization clustering results on a two-dimensional scatter plot to calculate a comparison metric even though the clustering was carried out on all 5 dimensions. Results of the proposed method on DLBCL dataset is represented in Figure 4.3.

4.2.3. Hematopoietic Stem Cell Transplant (HSCT) Dataset

We have also used the HSCT dataset, described in section 3.3.3, to evaluate the proposed method. We had used PE and APC fluorochrome channels only in the previous chapter for clustering. Now, we have used all six different fluorochrome channels. Annealing-based model-free expectation-maximization algorithm identified three distinct clusters and missed the smallest cluster like model-free expectation-maximization, that had only 100 samples of the 8914 samples. However, annealing-based model-free expectation-maximization algorithm performance is significantly higher compared to model-free expectation-maximization: model-free expectation-maximization performance was 0.8087 with three distinct clusters while annealing-based model-free expectation-maximization with a f-measure of 0.9827. The scatter plot showing the resulting clusters is shown in Figure 4.4.

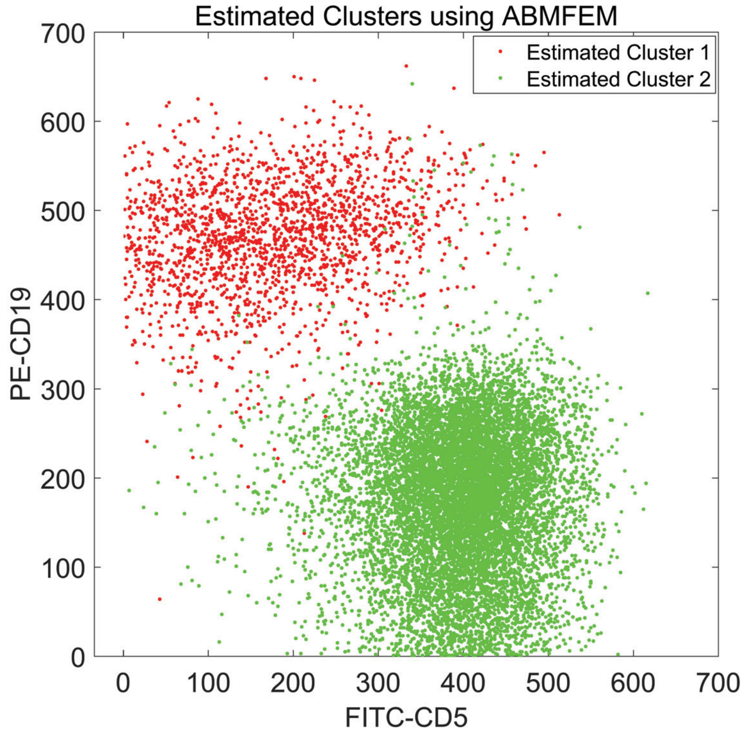


Figure 4.3. Estimated CLusters for The Diffuse Large B-cell Lymphoma (DLBCL) Dataset using ABMFEM

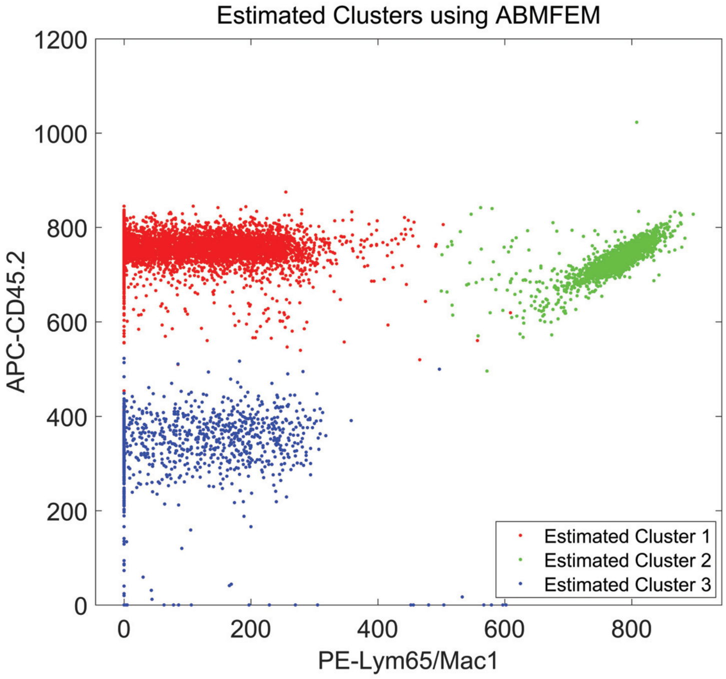


Figure 4.4. Estimated CLusters for The Hematopoietic Stem Cell Transplant (HSCT) Dataset using ABMFEM

4.3. Discussion

In this chapter, we introduced a recursive binary division algorithm for unsupervised clustering that does not require any knowledge about the data such as data distribution, model parameter, the number of clusters etc.The method operates by dividing the original dataset into two daughter clusters using the posterior probability estimates provided by the quasi-supervised learning algorithm in an expectation-maximization framework. In contrast to the earlier version, the algorithm uses an annealing-based approach to optimize the reference set size parameter n. It begins with a larger n and decreases n in each step until the entropy-based cost function minimized. The same procedure is then applied to the daughter clusters themselves and their daughter clusters and so on, until the division cost of the daughter clusters exceeds the division cost of the parent cluster by a significant margin.

In experiment results, the proposed method accurately identified the clusters of interest both on synthetic datasets as well as datasets collected from real multi-color flow cytometry experiments. The experiments also showed that clusters with too few samples were still at risk of being not recognized as separate clusters. From a statistical standpoint, it is not surprising that such small clusters are missed due to insufficient representation within the overall dataset. However, in applications where small clusters are of particular significance, additional measures are to be taken so that clusters with small representation are also recognized as such.

CHAPTER 5

AUTOMATIC COMPENSATION OF MULTICOLOR FLOW CYTOMETRY DATA USING JOINT DIAGONALIZATION

In multi-color flow cytometry experiments, manual analysis of the data becomes practically impossible as the number of parameter increases. To remedy the situation, flow cytometry data first needs a pre-processing step called *compensation*. Increasing the number of fluorochromes causes spillover, defined as the overlap between two or more fluorochromes' emission spectra. As a result the detectors cannot identify exclusively their target biomarkers when more than one fluorochrome are present in a cell with overlapping emission spectra. Compensation can be performed either during data collection on the flow cytometer, or after data collection in software. The compensation procedure is typically formalized as a linear algebra problem (Roederer, 2002; Bagwell and Adams, 1993a) described in Chapter 2, since the principal of superposition in measured ligth intensity applies through spillover parameters that can be measured using control samples. The most important goal in compensation is visualization of all distinct subpopulations as separate as possible from each other. To this end, research groups have been studying automatic compensation and automatic gating of multicolor flow cytometry data. However, in all proposed methods, control samples are needed to calculate the spillover coefficients (Roederer, 2001). Measuring the control samples and calibrating the flow cytometer for each experiment, on the other hand, is laborious work.

In this chapter, we describe our proposed methodology for automatic compensation and gating of high dimensional multi-color flow cytometry data. Our method is constructed on the premise that when properly compensated, all fluorochrome channels must be as orthogonal and independent from each other as possible. The algorithm begins with a data clustering part. Following the clustering of the uncompensated data, a joint diagonalization matrix is calculated over the identified clusters using Fast Frobenius Diagonalization (FFDIAG). Data is finally transformed using this matrix into a new coordinate system where all fluorochrome channels are approximately orthogonal to each other within each cell cluster.

5.1. Methodology

Flow cytometry gating aims to identify the distinct cell sub groups in a heterogeneous population. In a compensated dataset, these cell sub groups are placed in distinct quadrants and in such a way that the intensity values on each fluorochrome are approximately orthogonal to each other. Based on this premise, we have developed an automatic compensation procedure for multicolor flow cytometry datasets by combining our previously published annealing-based model-free expectation maximization algorithm and the FFDIAG algorithm. In addition, we have introduced gamma normalization for transformation of raw intensity measurements as it provides full automation in data transformation and achieves an optimal use of the dynamic range of values.

In the next sections, firstly we explain the gamma normalization and we demonstrate the effect of gamma normalization on the toy dataset used in Chapter 2. Next, we introduce the joint clustering and orthogonalization algorithm that achieves automated compensation and clustering of the flow cytometry data. Finally, we present results of this the algorithm on synthetically created Gaussian mixtures and on real multi-color flow cytometry data.

5.1.1. Gamma Normalization

We used gamma normalization for data visualization and processing of the flow cytometry intensity data because it allows calculating the operational parameters automatically from raw intensities to obtain an optimal use of the dynamic range. For a collection of raw intensities $\{x_i\}, i = 1, 2, \ldots, N$, the gamma normalized data $\widetilde{x_i}$ is defined as;

$$\widetilde{x_i} = (ax_i + b)^{\gamma} \tag{5.1}$$

where a is the scale parameter, b is the bias value and γ is the power factor. The parameters a and b are defined by the expressions

$$\begin{aligned} a &= \frac{(N-1)}{(N+1)(x_{(N)} - x_{(1)})} && (5.2) \\ b &= -ax_{(1)} + \frac{1}{N+1} && (5.3) \end{aligned}$$

where $x_{(i)}$ denotes the i^{th} smallest intensity values among $\{x_i\}$ with

$$x_{(1)} = \min_i x_i \tag{5.4}$$

and

$$x_{(N)} = \max_i x_i, \tag{5.5}$$

while γ is determined by a line search to achieve

$$\sum_{i=1}^{N} \left(\frac{i}{(N+1)} - (ax_i + b)^{\gamma} \right) = 0 \tag{5.6}$$

leading to an optimal use of the dynamic range between 0 and 1 without altering the inherent intensity information. We demonstrated four different transformation scales for flow cytometry data including gamma normalization and these described previously in Chapter 2. Compared to the original linear scale, after gamma normalization, all three clusters are placed distinctly and can therefore be identified with relative case using a statistical clustering method of choice. Since gamma normalization also provides full automation of the data transformation, we used it in this study as a preprocessing step on raw intensity data. We have demonstrated the gamma normalization of the toy dataset described in Chapter 2 in Figure 5.1.

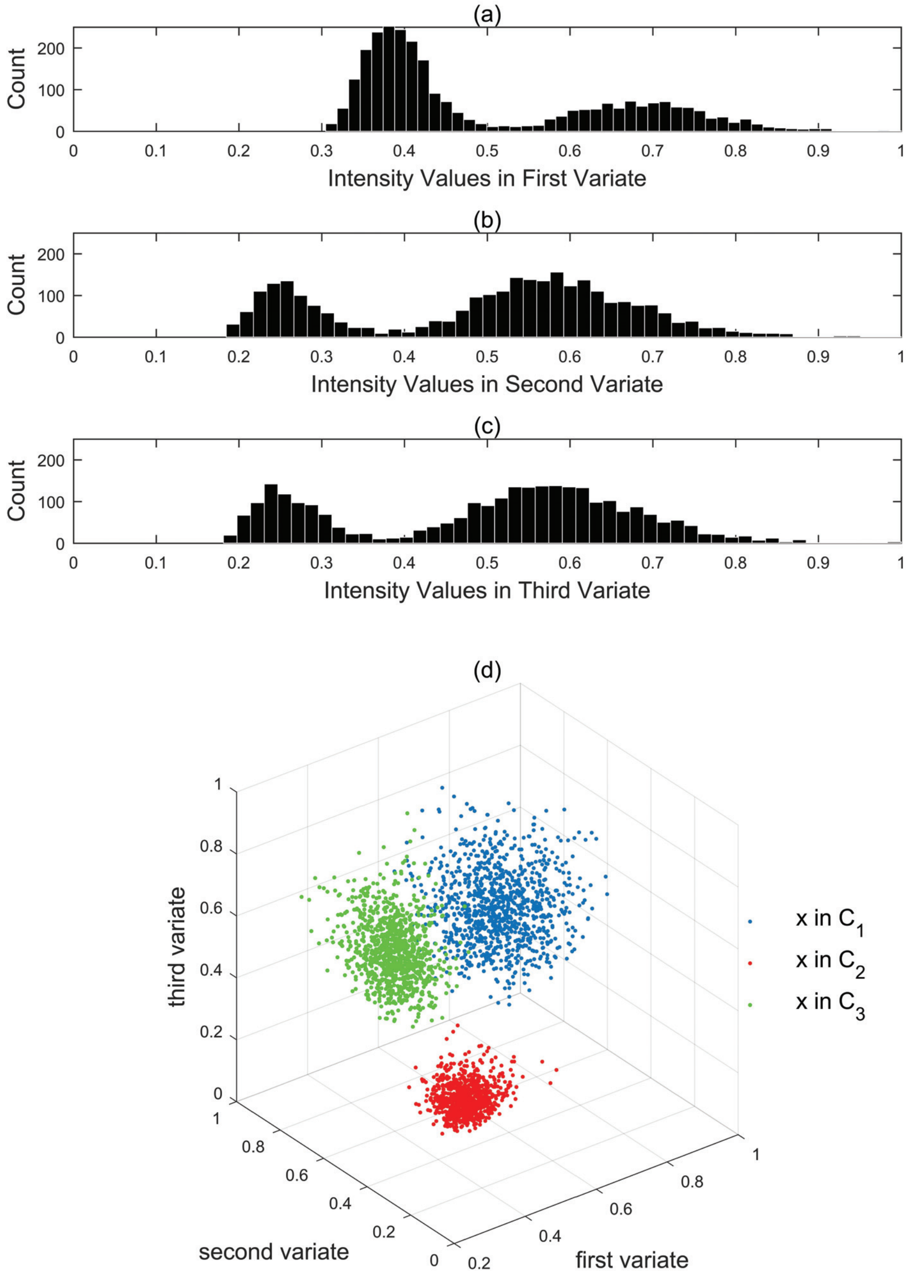


Figure 5.1. Gamma Normalized Instensity Values for Toy Dataset that Created in Chapter 2

5.1.2. Joint Clustering and Orthogonalization Algorithm

The main objection of our study is to achieve full automation in flow cytometry data analysis as it applies equally to both compensation and gating. To this end, we have evaluated two variants of the same underlying strategy; one is based on appliying the joint diagonalization procedure on the gamma normalized data, and the other is based on applying the joint diagonalization procedure on the raw data much like conventional compensation, following initial clustering.

Our algorithm block diagram is shown in Figure 5.2. First, the uncompensated dataset is gamma normalized and clustered using the annealing-based expectation maximization algorithm. The compensation matrix is obtained from the resulting clusters using Fast Frobenius Diagonalization (FFDIAG). The compensation matrix is then applied to the gamma normalized data and the compensated data is gamma normalized and clustered again one last time for better visualization and as final verification of the earlier clusters.

As an alternative, the second scheme carries out the joint diagonalization on the raw data since conventionally, the compensation matrix is obtained on the raw dataset instead of the gamma normalized data, we have compared the results in both cases obtained using both methods on synthetically created data as described below.

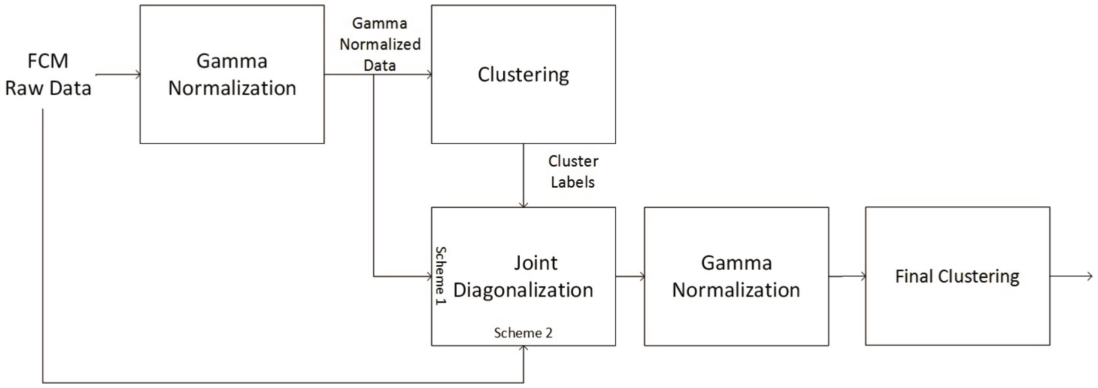


Figure 5.2. Block diagram of the proposed method

5.2. Results

We have illustrated our approach on the synthetically created dataset first and then on real multi-color flow cytometry data. Below, we first describe the specifics of the synthetic dataset and provide the results obtained using both alternatives. Next, we present results obtained on three different scenarios derived using real flow cytometry data.

5.2.1. Synthetically Created Dataset

We have created a *toy* dataset with 3 distinct clusters, C_1, C_2, C_3, each modeled using the exponential of three-dimensional Gaussian components with covariance matrices Σ_1, Σ_2, Σ_3 and with means μ_1, μ_2 and μ_3, respectively. Samples were drawn from this mixture using different priors. For each cluster the number of samples were N_1, N_2 and N_3, respectively. In the experiment, the covariance matrices were taken as

$$\Sigma_1 = \begin{bmatrix} 0.4 & 0.1 & 0.2 \\ 0.1 & 0.6 & 0.5 \\ 0.2 & 0.5 & 0.5 \end{bmatrix}, \qquad \Sigma_2 = \begin{bmatrix} 0.5 & 0.2 & 0.1 \\ 0.2 & 0.4 & 0.2 \\ 0.1 & 0.2 & 0.5 \end{bmatrix}, \qquad \Sigma_3 = \begin{bmatrix} 0.3 & 0.1 & 0.3 \\ 0.1 & 0.4 & 0.1 \\ 0.3 & 0.1 & 0.5 \end{bmatrix} \tag{5.7}$$

and with the mean vectors

$$\mu_1 = \begin{bmatrix} 3 & 5 & 2 \end{bmatrix}^T \tag{5.8}$$

$$\mu_2 = \begin{bmatrix} 3 & 4 & 3 \end{bmatrix}^T \tag{5.9}$$

$$\mu_3 = \begin{bmatrix} 3 & 3 & 5 \end{bmatrix}^T. \tag{5.10}$$

The toy data set illustrated in Figure 5.3. As seen from the figure, some transformation is required to identify the different clusters. Next, we have applied gamma normalization and clustered the dataset using annealing-based model-free expectation-maximization algorithm with f-measure 0.9645. The clustering results before the diagonalization on the gamma normalization scale is illustrated in Figure 5.4, while the estimated clusters on the original data domain are in Figure 5.5. Then, we have applied FFDIAG algorithm on gamma normalized scale as for the first alternative, to make clusters jointly diagonal and performed a final clustering. The diagonalized clusters are presented in Figure 5.6. Finally, we have calculated another diagonalization matrix on raw data scaleas for the second alternative, and finished with a final clustering. The results are shown in Figure 5.7.

Diagonalization results show that calculating the transformation matrix for compensation on the gamma normalized scale or raw data scale causes some differences. This is due to differences of the covariances in different domains. Since the elements of the covariance matrices contain large values on raw data domain, FFDIAG algorithm cannot converge to a good diagonalization matrix. While the conventional compensation performed on raw data scale, in our case diagonalization of clusters on raw data scales caused deformations on the clusters' shape. On the other hand, calculating the transformation matrix over the clusters on the gamma normalized scale creates visually better looking clusters with more prononced separation.

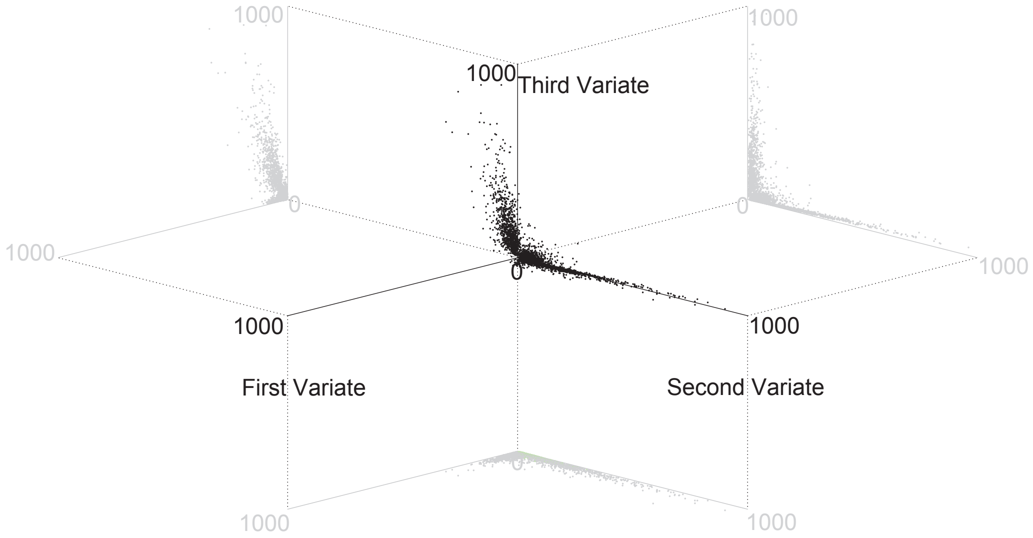


Figure 5.3. Original toy dataset after the exponential transformation. In calculations, this data was used to simulate raw intensity data for compensation and clustering.

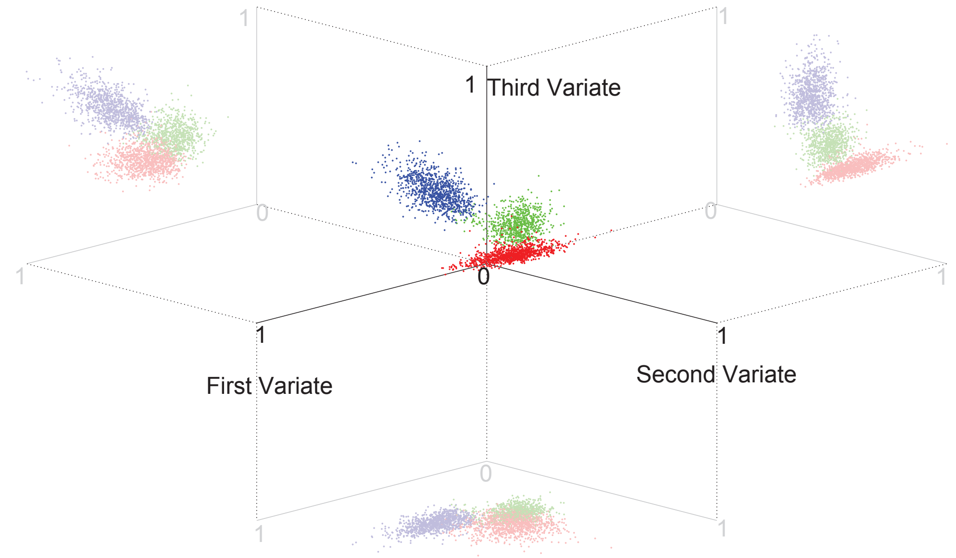


Figure 5.4. Gamma Normalized Toy Data with The Estimated Cluster Labels

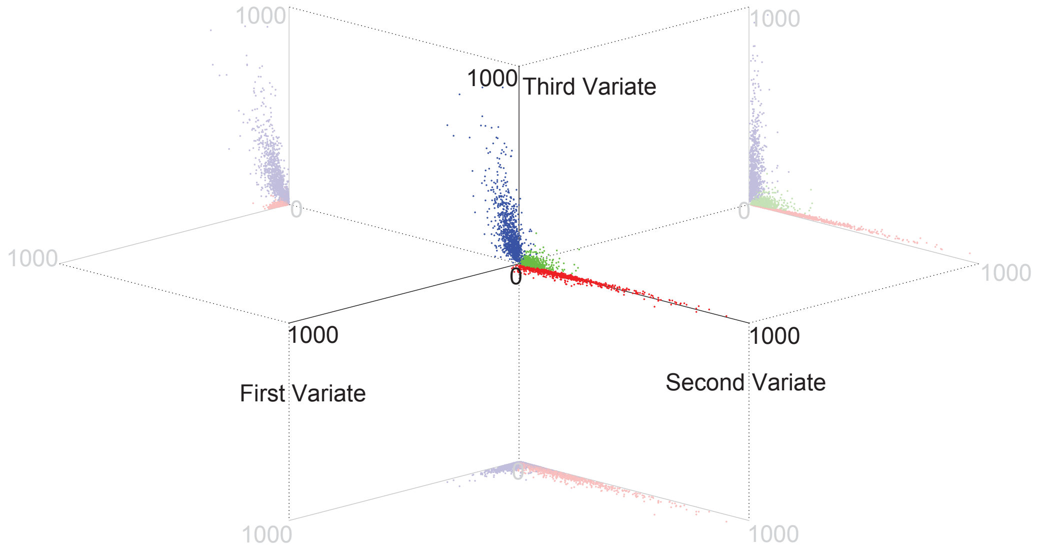


Figure 5.5. The Estimated Cluster Labels on Raw Data

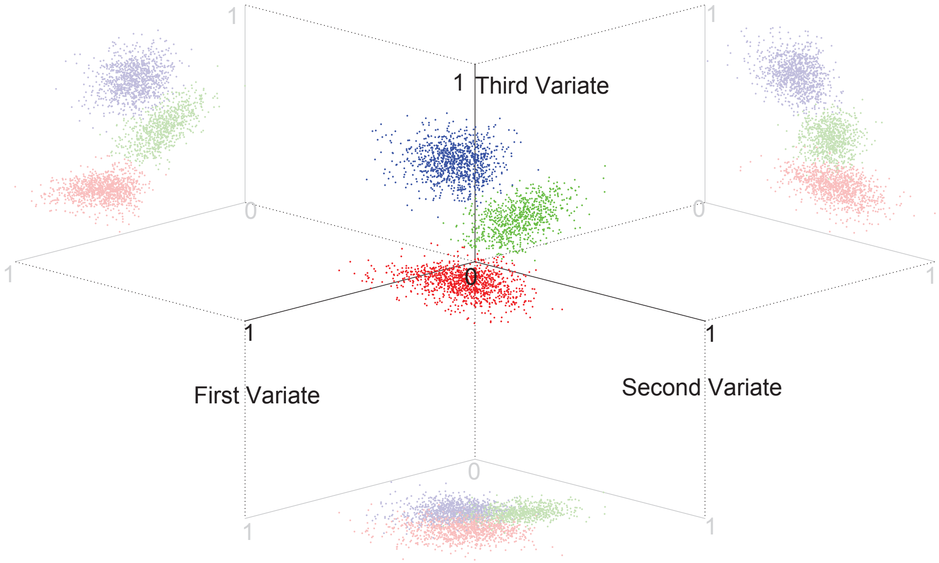


Figure 5.6. Gamma Normalized Data After Clustering and Joint Diagonalization (first alternative)

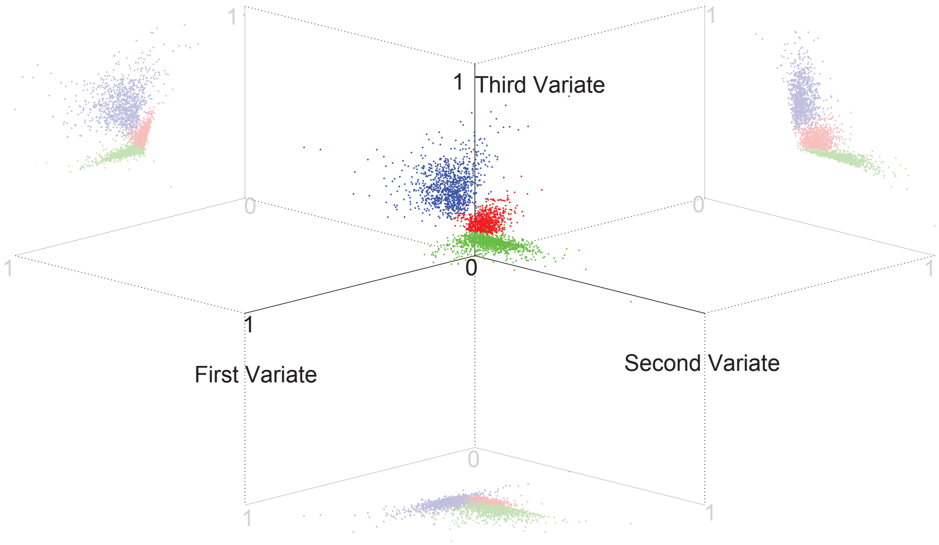


Figure 5.7. Gamma Normalized Data After Clustering and Joint Diagonalization (second alternative)

5.2.2. Results on the Real Multi-Color Flow Cytometry Dataset

The real multi-color flow dataset was obtained from Flow Repository database (Repository ID: FR-FCM-ZZWB (Gruetzkau, 2010)). Based on the experiment definition, whole blood leukocytes were taken from three healthy individuals and analyzed in an 8-color flow experiment. The cells were experimentally modified by the depletion of one particular cell type per sample, including granulocytes (using CD15-MACS-beads), monocytes (using CD14-MACS-beads), T lymphocytes (CD3-MACS-beads), T helper lymphocytes (using CD4-MACS-beads), and B lymphocytes (using CD19-MACS-beads).

We have chosen the lymphocyte population whose autofluorescence is smaller compared to other populations in our experiments, since, this allows a better match for the orthogonality premise for compensation: In flow cytometry experiments, cell autofluorescence often interference with the low level measurement of the bounded fluorescence (Alberti et al., 1987). The dataset has 8 different fluorochrome channels, and both uncompensated and compensated datasets are available. We have tested our algorithm on randomly chosen 10000 samples. Firstly, we took three fluorochrome channels with the most overlapping spectra. Then, we have also tested our algorithm on the three channels with the least overlap. In the final scenario, we compensated the dataset using all 8 channels. For validation, have manual compensation results in the dataset. However, this compensation was performed by taking into account the needs of the actual experiment where the expert adjusted the compensation parameters to identify interested cell populations using 2D scatter plots. As a result, we could not compare our results mathematically, because our algorithm evaluates the fluorescence intensities on all channels simultaneously. Nevertheless we have includeed all manual compensation results along with our algorithm results to compare the appearance of the cell sub-groups. Finally, we have compensated the datasets with both schemes described in the algorithm block diagram in Figure 5.2. We have clustered the dataset and determined a transformation matrix that diagonalizes the clusters over gamma-normalized values according to scheme 1, and over raw values according to scheme 2. After the diagonalization, we have performed a final clustering to improve the clustering accuracy.

The three most overlapping fluorochromes in terms of their emission spectra were Pacific Blue, AmCyan and FITC. We used these three fluorochrome channels to test our algorithm in the first scenario. In Figure 5.8, the uncompensated dataset on gamma normalization scale is presented. While, manual compensation result for these three channels is shown in Figure 5.9. Note that manual compensation results are clearly inaccurate for these three channels, even tough manually compensated values were obtained using all 8-colors. The deleterious effects of the other channels' compensation parameters can be observed in these three channels. This is caused by the difficulty of proper multi-color compensation using manual techniques. The results of our proposed method for both schemes are presented in Figures 5.10 and 5.11. Despite the fact that these are the most overlapping channels, in both cases, our algorithm successfully determines the cell sub-groups and align them over the non-discriminant channels.

In the second scenario, we chose the three channels corresponding to fluorochromes emission spectra overlapped minimally (Pacific Blue, APC-Cy7 and PE). We have applied our algorithm again using both schemes. Note that both uncompensated and manual compensated data looks good as expected in Figures 5.12 and 5.13 respectively. Since the spillover between these channels is minimal using scheme 1, the transformation matrix obtained over gamma normalized data is quite similar in the both uncompensated and manual compensated data since the clusters are well-placed and already close to being orthogonal in the gamma normalized space (Figure 5.14). However, using scheme 2 where the transformation matrix is obtained from raw data, the results are problematic: As seen from Figure 5.15, some clusters appear to have undergone undesired deformations. This is caused by the same phenomenon observed in the toy dataset, where the joint diagonalization algorithm could not converge to a proper transformation matrix. Poor convergence of the joint diagonalization on the linear data scale can be attributed to the presence of cells with extensively high fluorescence values that dominate the covariance calculations. For this reason, we have used our proposed algorithm under scheme 1 for a full 8-color data compensation in the final scenario. Figures 5.16 and5.17 present all pairwise 2D scatter plots of the uncompensated data and manual compensated data in all channels, respectively. As discussed before, serious short comings of the manual compensation in multi-color flow cytometry are visible Figure 5.17, evidenced by strong residual correlations with cells positioned along the main diagonal (e.g. APC-Cy7-A and Pacific Blue-A Scatter Plot). We have presented the automated compensation results with identified cell sub-groups in Figure 5.18. The results show that our algorithm can identify the cell-sub groups when the statistical information is sufficient as before, and by joint diagonalization of these clusters, proper compensation can be achieved for multi-color flow cytometry data.

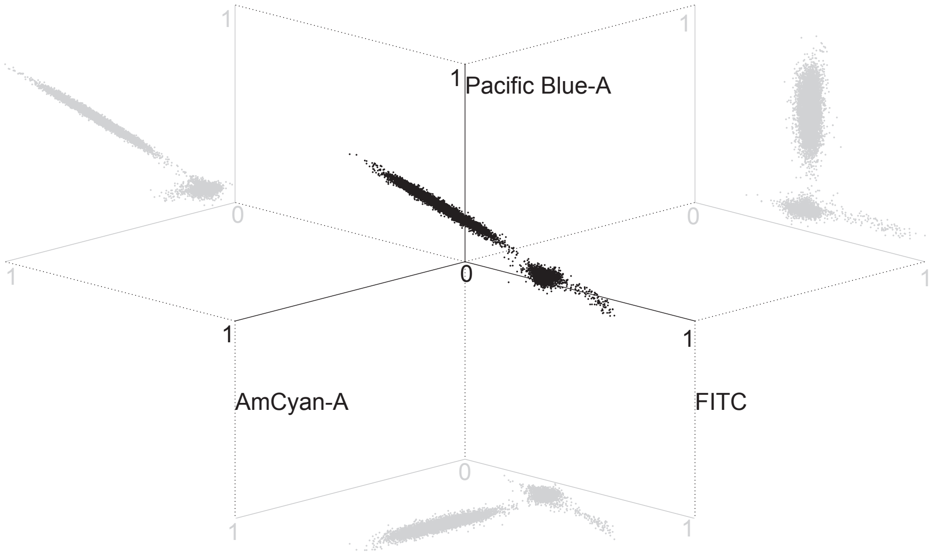


Figure 5.8. Scatter plots of the most overlapping channels (uncompensated)

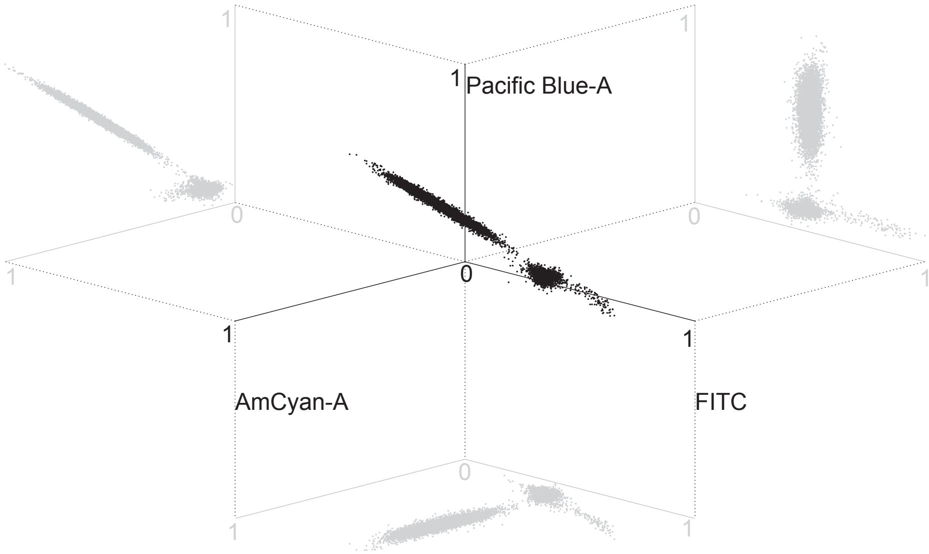


Figure 5.9. Scatter plots of the most overlapping channels (manually compensated)

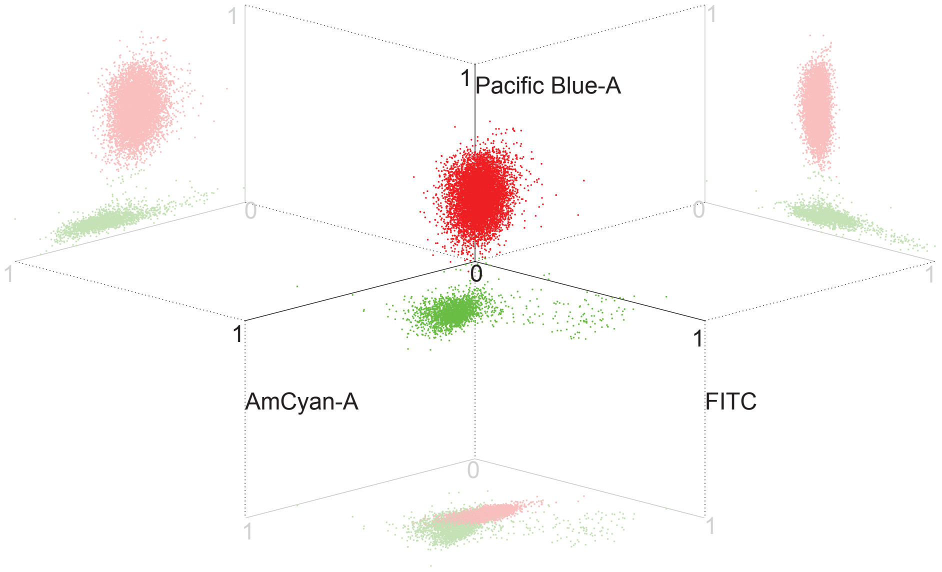


Figure 5.10. Scatter plots of the most overlapping channels (compensated according to scheme 1)

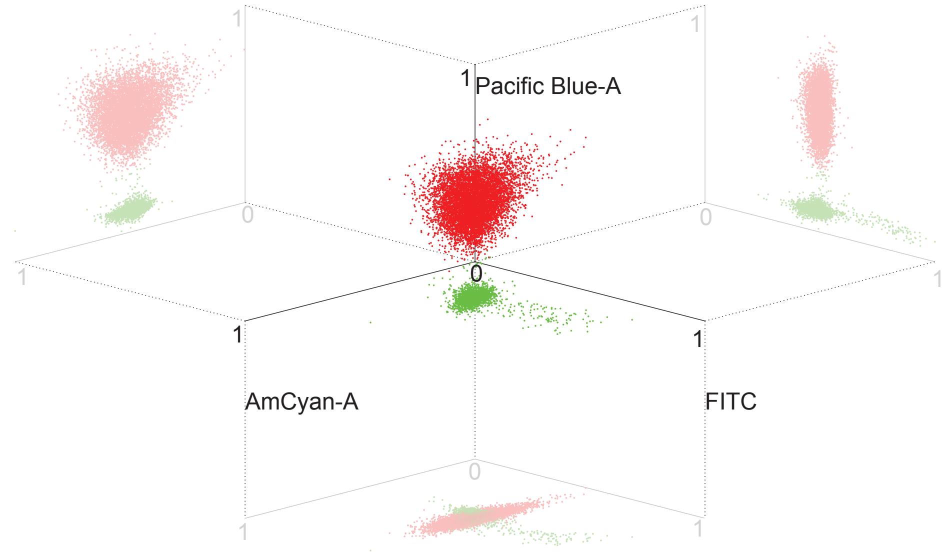


Figure 5.11. Scatter plots of the most overlapping channels (compensated according to sheme 2)

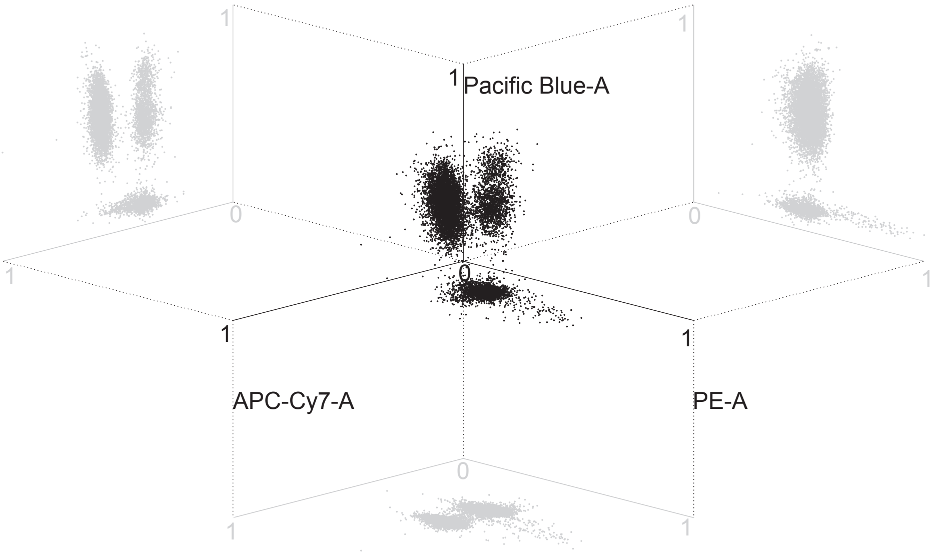


Figure 5.12. Scatter plots of the least overlapping channels (uncompensated)

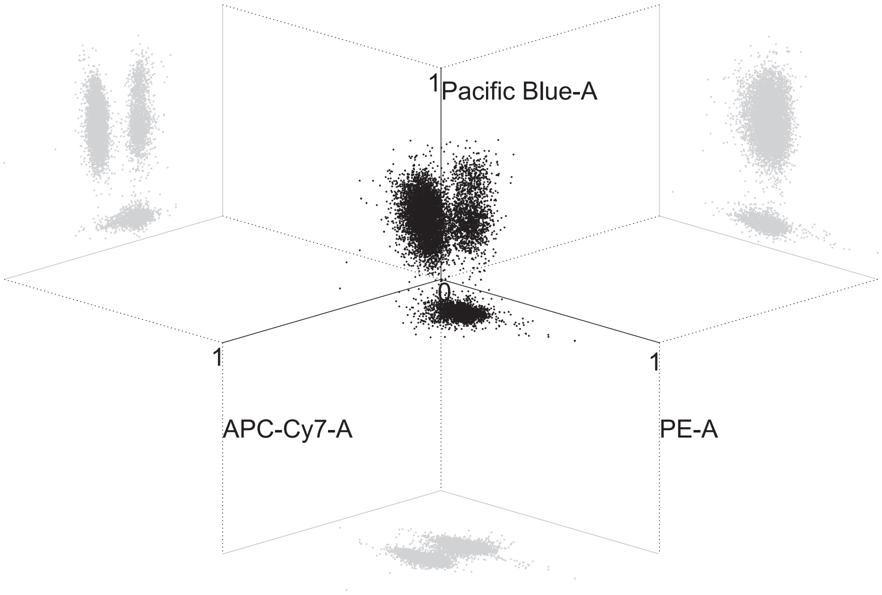


Figure 5.13. Scatter plots of the least overlapping channels (manual compensated)

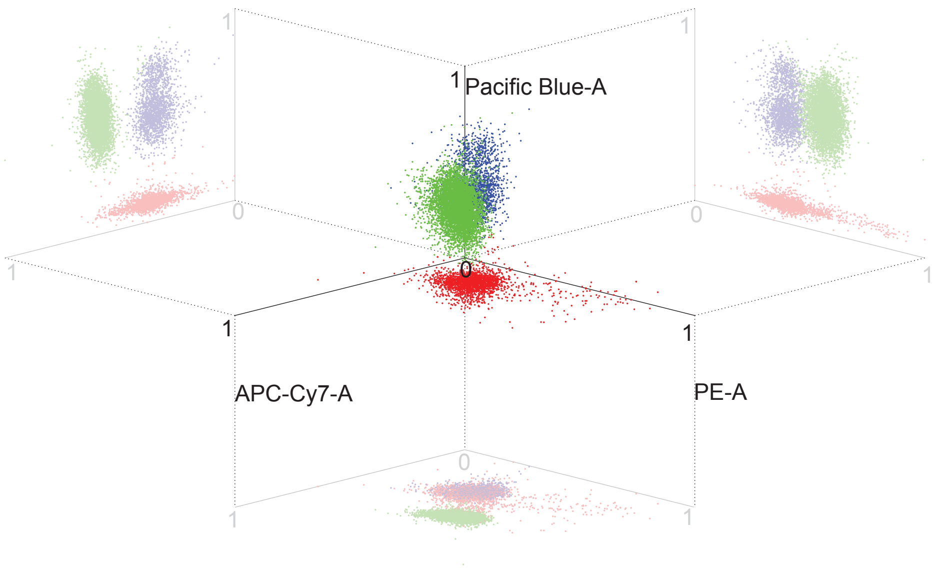


Figure 5.14. Scatter plots of the least overlapping channels (compensated according to scheme 1)

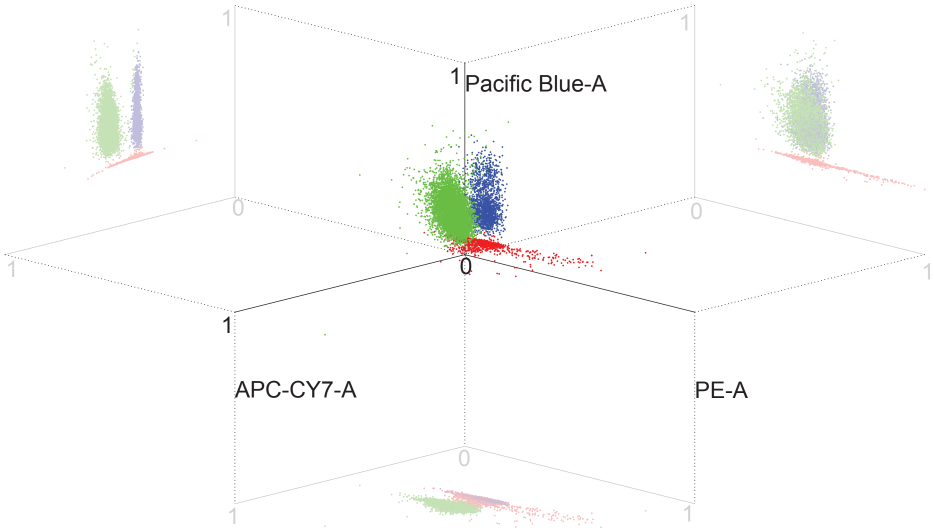


Figure 5.15. Scatter plots of the least overlapping channels (compensated according to scheme 2)

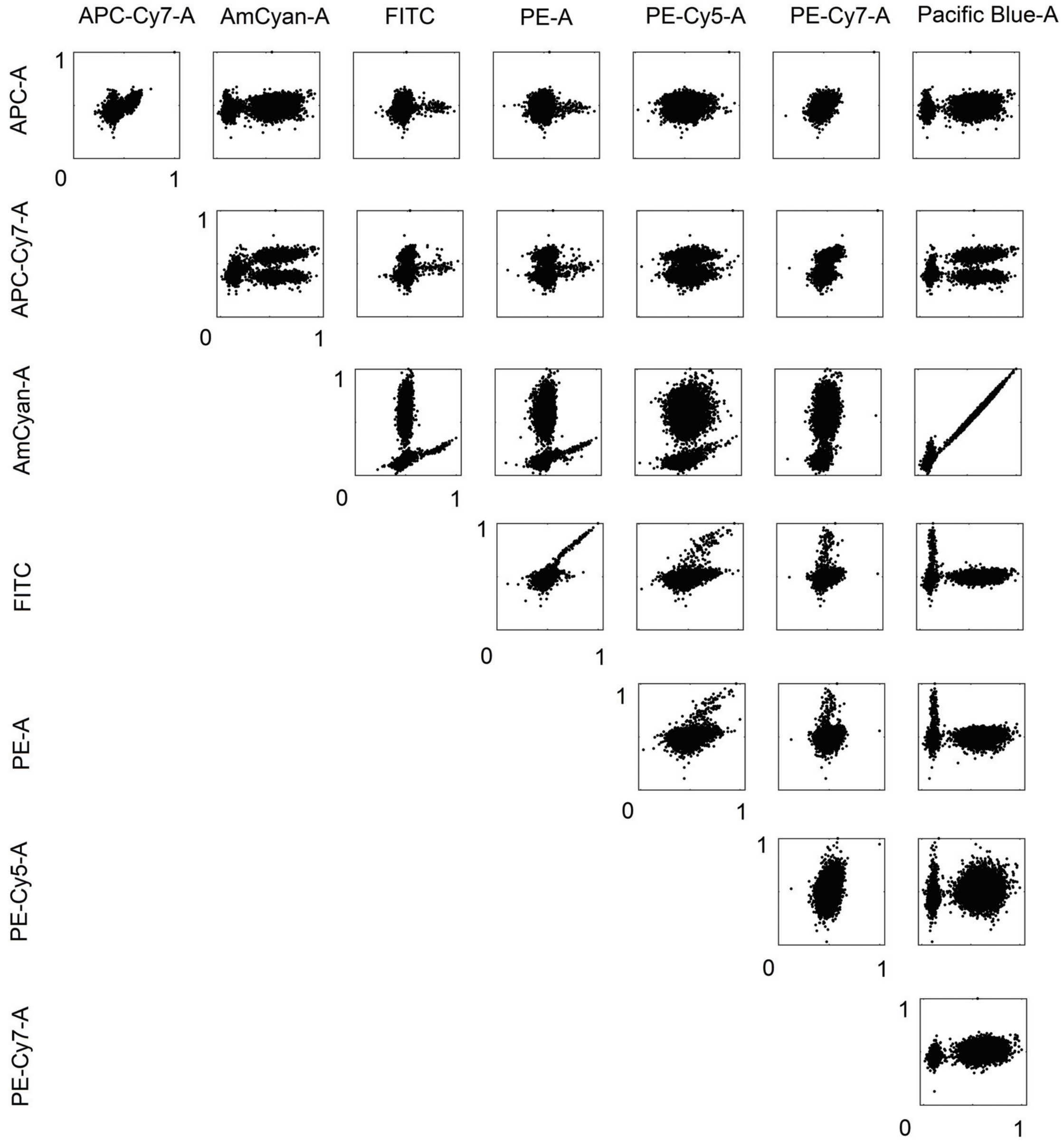


Figure 5.16. Pairwise scatter plots of the uncompensated dataset after gamma normalization. All values are within the unit interval.

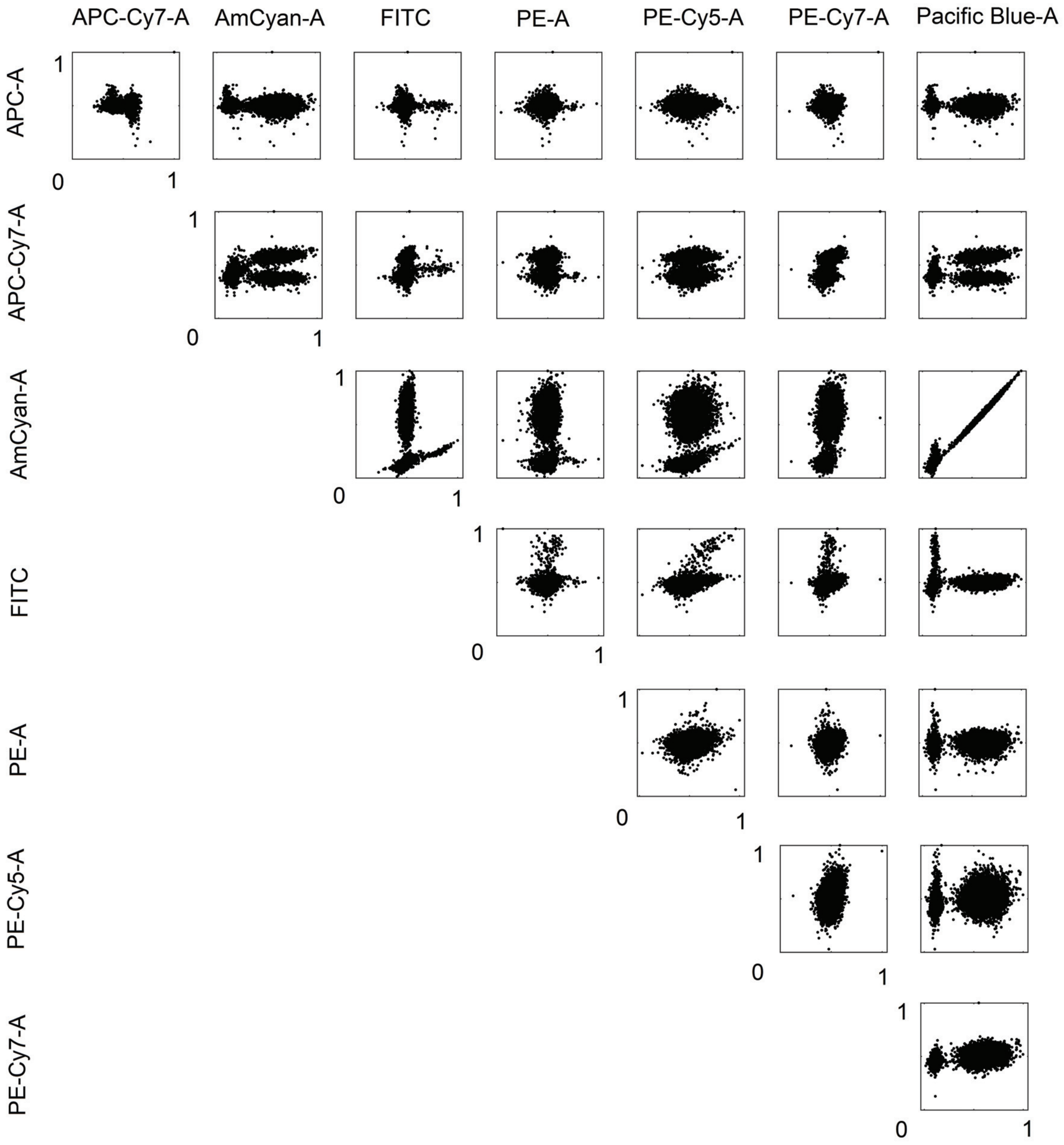


Figure 5.17. Pairwise scatter plots of the manually compensated dataset

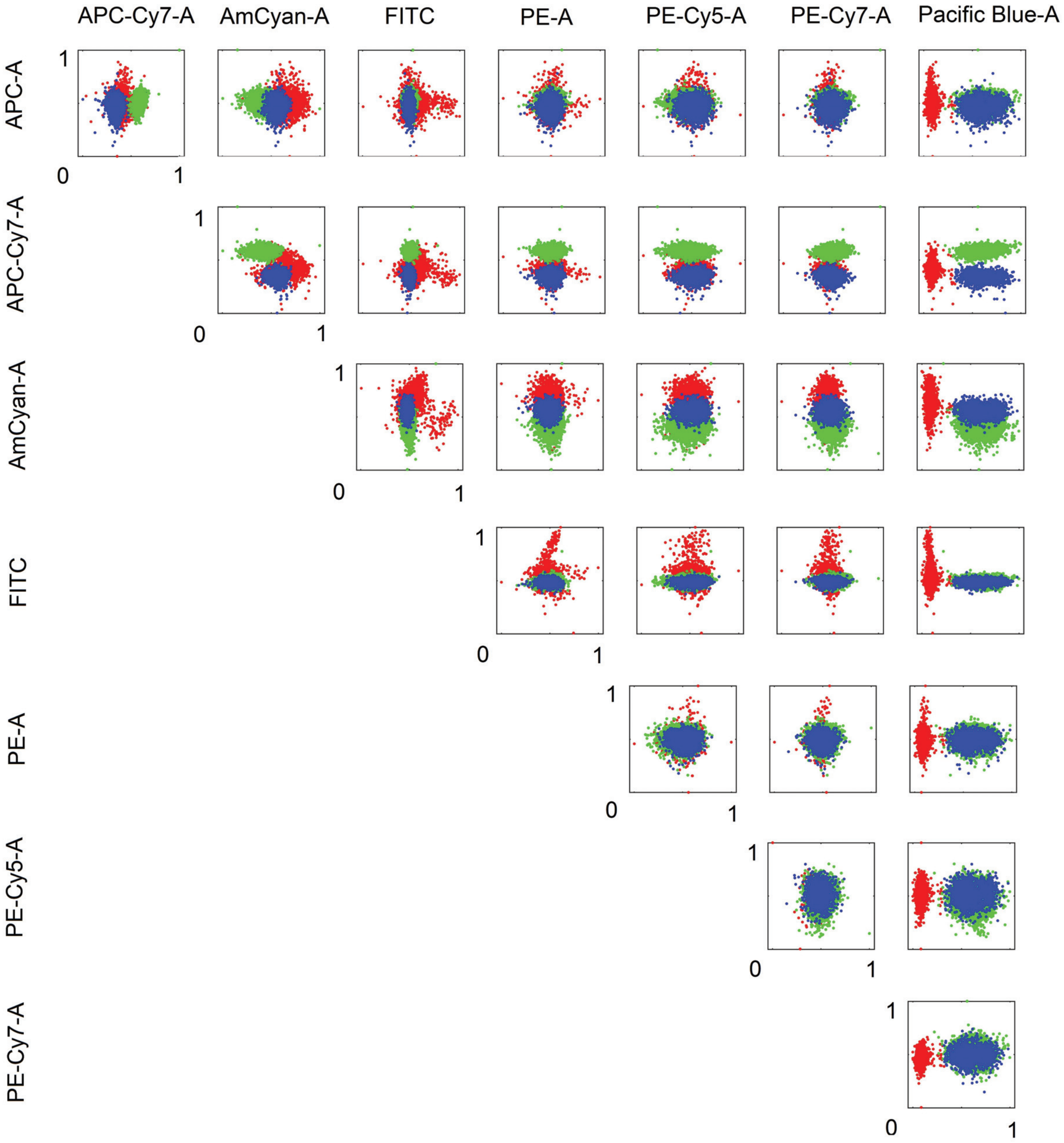


Figure 5.18. Pairwise scatter plots of automatically compensated dataset with the identified cell sub-groups. In total, three cell subgroups were identified, indicated by the red, green and blue colors respectively.

We have applied the proposed method to another real multi-color flow dataset obtained from the Flow Repository Database, too (Repository ID: FR-FCM-ZZ6B, Craig et al., 2014) designed to distinguish germinal B-cell lymphoma from reactive lymphoid tissue. In particular, we chose the datasets LNA31 and LNA77. The LNA31 data was obtained from a healthy subject and the LNA77 data was obtained from a subject who had lymphoma. All experiments were performed using 8 fluorochromes to stain the biomarker: anti-kappa, anti-lambda, CD19, CD20, CD10, CD5, CD38 and CD45. Both datasets contain only lymphocyte populations.

In accordance with the aim of the experiment, the authors label germinal B-cell lymphoma cells in a multi-step procedure. First, they label CD10 positive cells to identify germinal center (GC) reaction. Then, they investigate the expression of typical B-cell markers (**?**). In contrast, we have applied our algorithm on the full dataset to label all cell subgroups. In addition, manually compensation was performed by the authors to suit the needs of the experiment (Craig et al., 2014). As a result, proper manual compensation for all fluorochromes cannot be claimed. For comparison purposes,we have included all manual compensation results as provided by the dataset along with our algorithm results to compare the appearance of the cell sub-groups.

Figure 5.19 and Figure 5.20 represents the uncompensated and manually compensated pairwise scatter plots of the LNA31 dataset in gamma normalization scale. Our algorithm results along with the identified cell sub groups are illustrated in Figure 5.20. Our algorithm identified four distinct cell subgroups with proper alignment. Algorithm results can best be seen in the following scatter plots: PE/FITC, PE-Cy7/PE, PerCp-Cy5.5/PE.

Figure 5.22 ande Figure 5.23 show the uncompensated and manually compensated intensity values of the LNA77 dataset in gamma normalization scale respectively. Figure 5.24 represents the cell subgroups, three in this case, and the corresponding cell placement. In some scatter plots, especially in PerCp-Cy5.5 channel, it appears that the blue group may in fact include two cell subgroups. However, our clustering algorithm can not find statistical evidence to divide the blue population into two groups, that one the left as one single cell cluster.

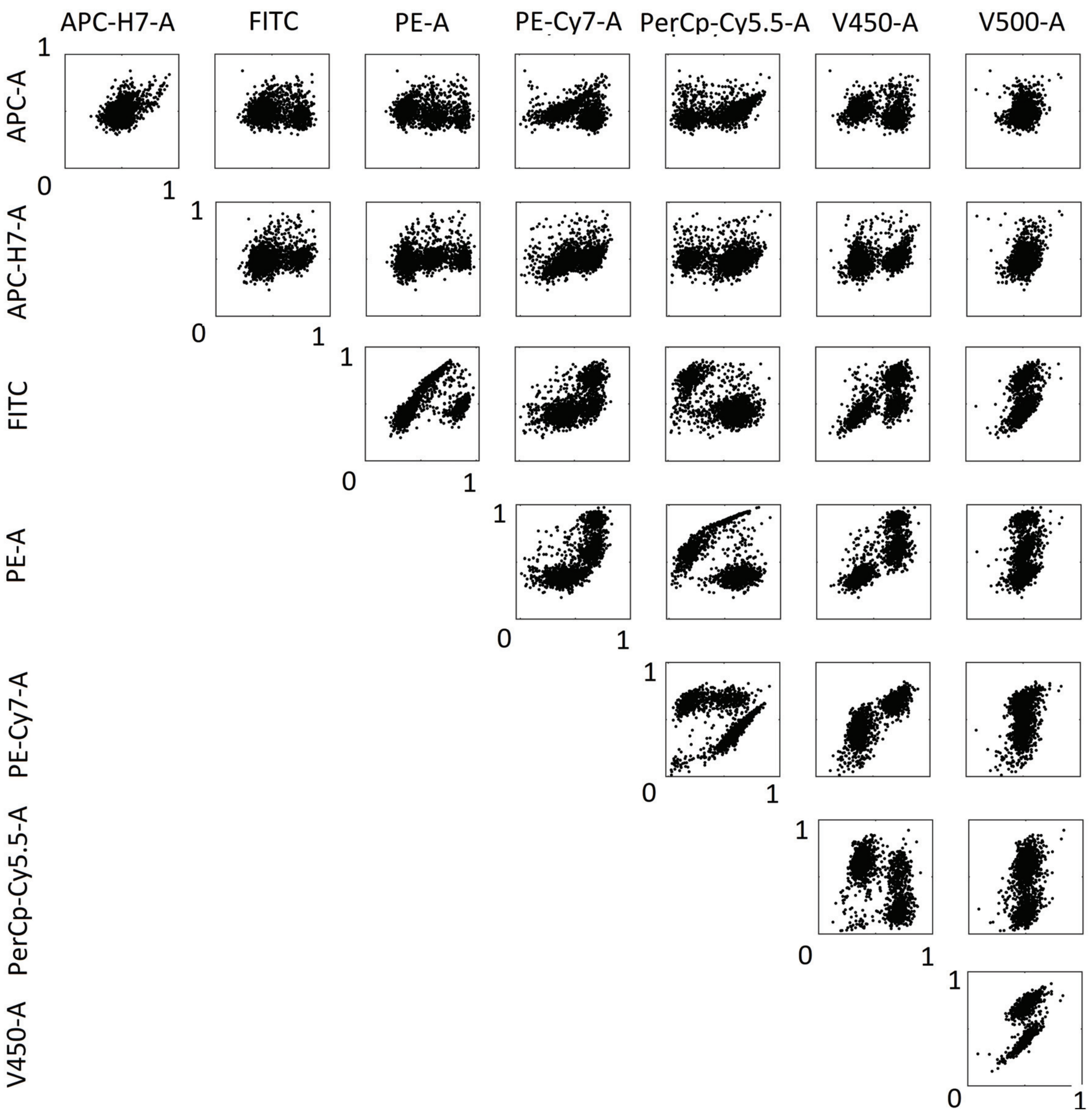


Figure 5.19. Pairwise scatter plots of the uncompensated LNA31 dataset after gamma normalization. All values are within the unit interval.

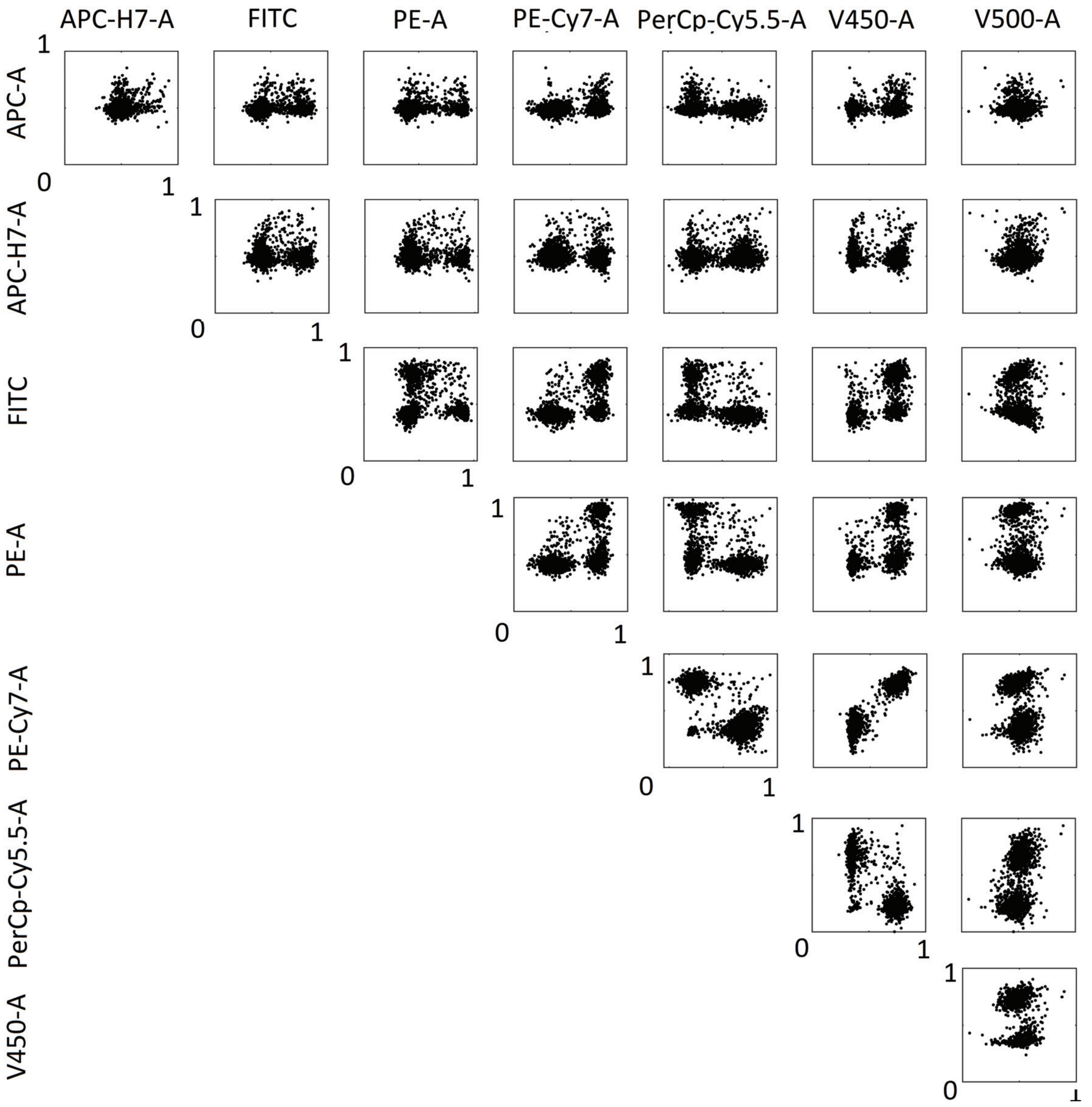


Figure 5.20. Pairwise scatter plots of the manually compensated LNA31 dataset after gamma normalization. All values are within the unit interval.

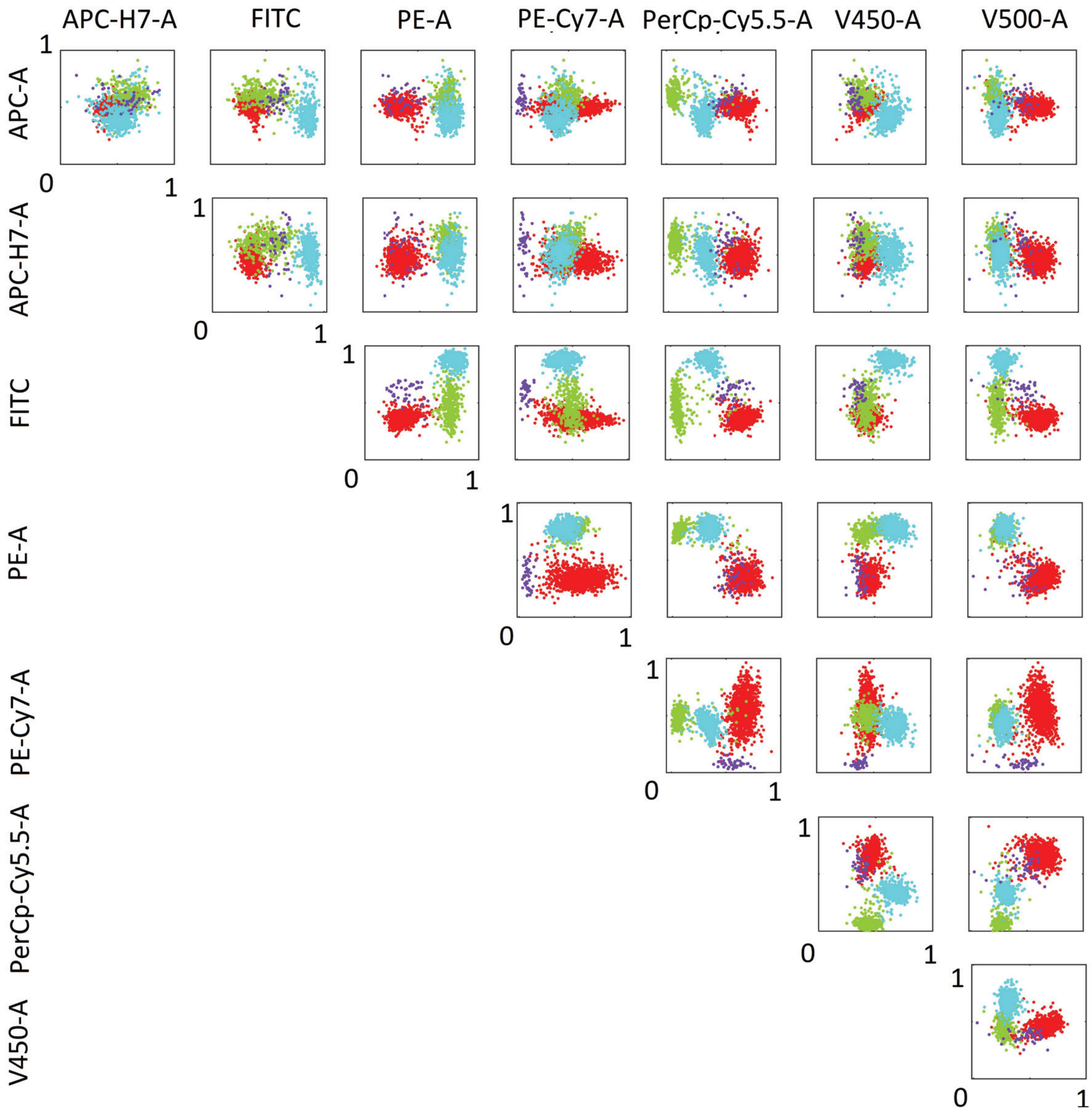


Figure 5.21. Pairwise scatter plots of automatically compensated LNA31 dataset with the identified cell sub-groups. In total, four cell subgroups were identified, indicated by the red, green, blue and purple colors respectively.

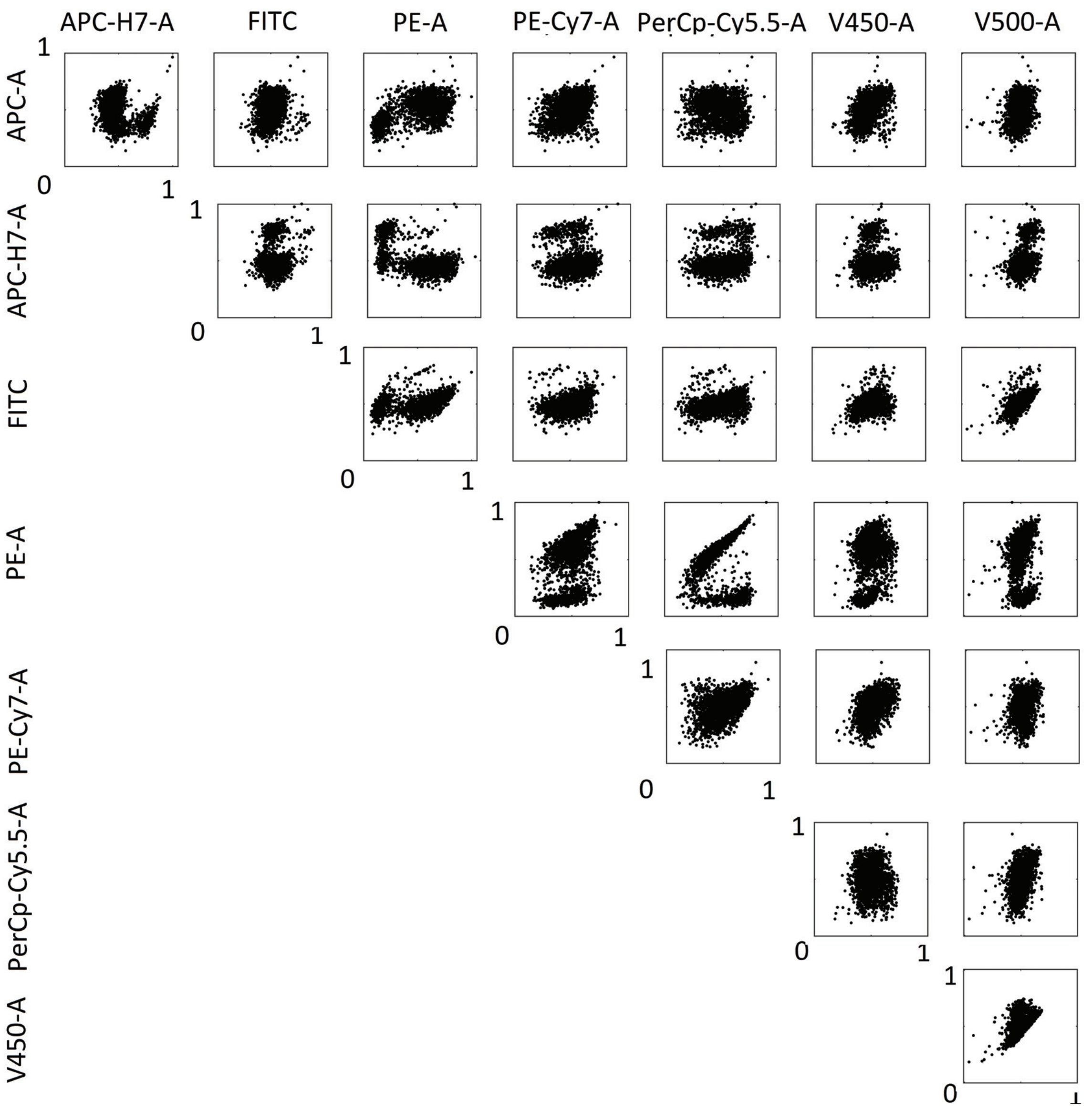


Figure 5.22. Pairwise scatter plots of the uncompensated LNA77 dataset after gamma normalization. All values are within the unit interval

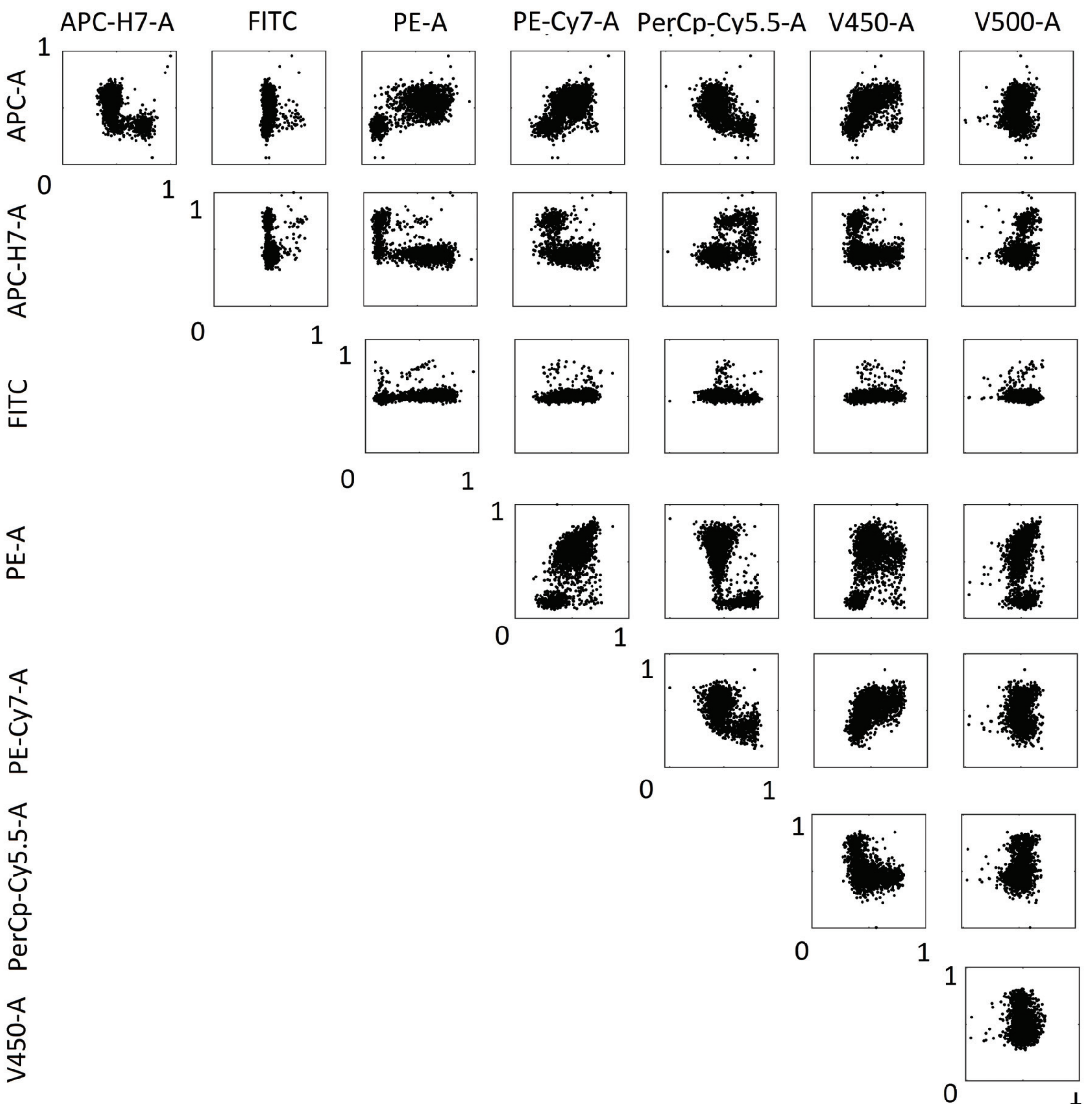


Figure 5.23. Pairwise scatter plots of the manually compensated LNA77 dataset after gamma normalization. All values are within the unit interval.

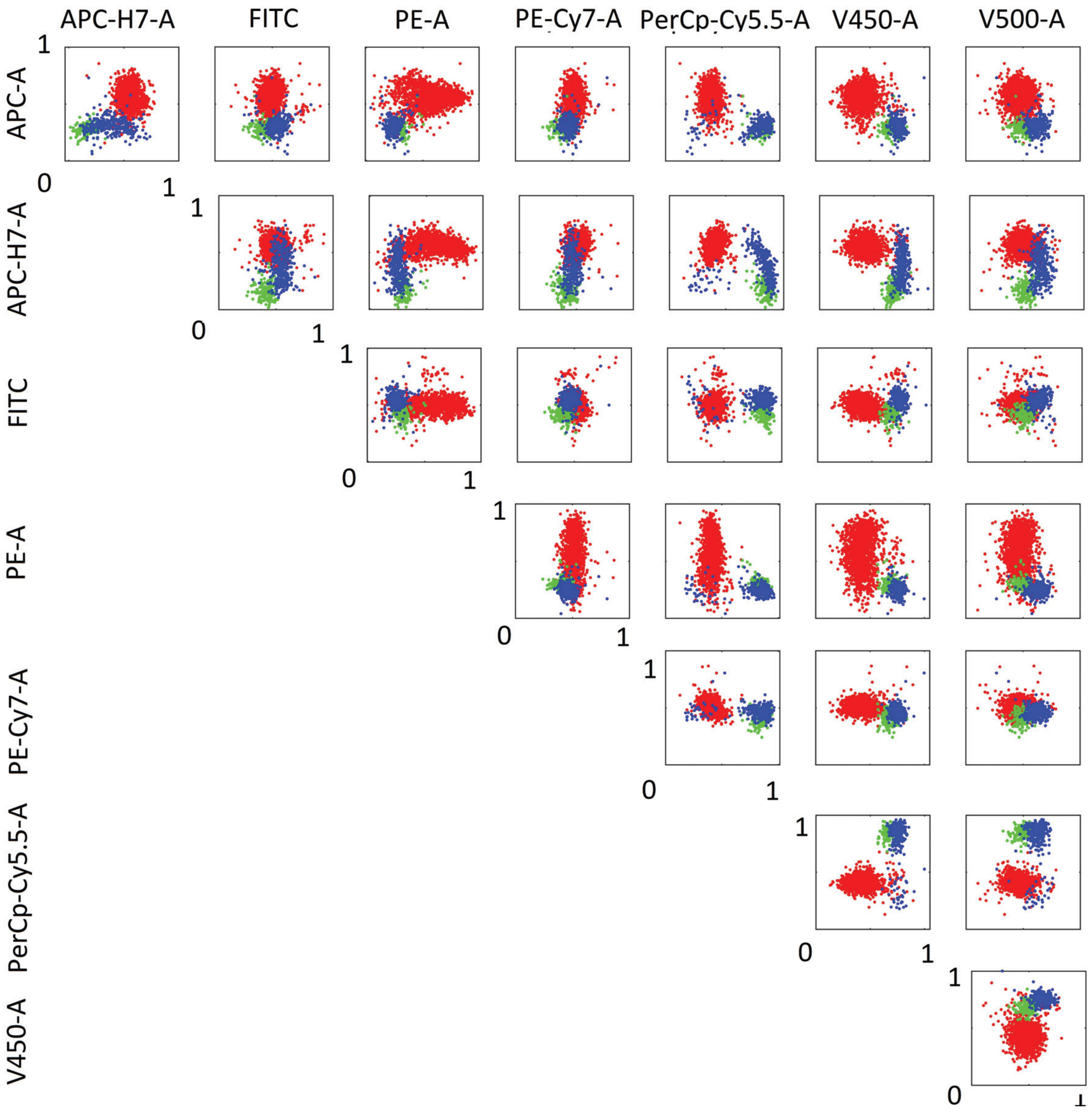


Figure 5.24. Pairwise scatter plots of automatically compensated LNA77 dataset with the identified cell sub-groups. In total, three cell subgroups were identified, indicated by the red, green and blue colors respectively.

5.3. Discussion

In this study, we have developed a joint automatic compensation and gating algorithm for multi-color flow cytometry. The algorithm begins by clustering multi-color flow cytometry data using annealing-based model-free expectation maximization algorithm. Joint diagonalization of these clusters calculates the compensation matrix. The compensated data is clustered one final time to revise the initial cell sub groups.

We have tested our algorithm on three real 8-color flow cytometry dataset. In the first one we have tested our algorithm on three different scenarios: intensity data from three fluorochrome channels with the largest overlap in their emission spectra, intensity data from another set of three fluorochrome channels with the least spectral overlap, and the full 8-channel data. The results show that our algorithm successfully performed gating and compensation in all three cases. It identified distinct cell sub populations and aligned them over non-differentiating fluorochromes. Interestingly, in our experiments, automatic compensation using joint diagonalization was more successful on gamma-normalized data than on raw intensities, in spite of the fact that gamma normalization induces substantial non-linearity on the intensities. This suggest that the linearity of the original compensation problem may not always hold, potentially due to inherent non-linearities in the flow cytometry detector circuitry. Additional colors may also be exacerbating this situation. In addition, the final re-clustering of the data following automatic compensation was of little consequence as the fraction of all cells that were assigned to different clusters than original was less than 0.5%

In the second and third datasets, we have performed our algorithm on full 8-color intensity data and our algorithm identified distinct cell subgroups and aligned them properly.

It should be noted that the evaluation of the proposed method was carried out over lymphocytes, that are a lower intensity cell group, and are therefore expected to possess uncorrelated fluorochrome intensities when properly compensated. The performance of the method described here on the other cells, such as monocytes, is currently under investigation. The performance of the algorithm can be increased by the adaptation of the update matrix with prior fluorochrome information.

CHAPTER 6

CONCLUSION

This thesis presents an automated approach for compensation and gating of multi-color flow cytometry data. The methods developed to this end include two clustering algorithms for automated gating and one joint diagonalization procedure for automated compensation. The respective performances of the proposed methods were evaluated on synthetically created datasets as well as real multi-color flow cytometry datasets.

We have started by developing model-free expectation-maximization (MFEM) clustering for automatic gating in Chapter 3. Model-free expectation-maximization is a binary divisive clustering for low dimensional datasets. It determines the number of clusters and respective clusters for each sample without any model assumption. The performance of the clustering algorithm was measured using f-measure, and the results show that model-free expectation-maximization clustering identifies the number of clusters and assigns samples to the true clusters with remarkable accuracy, especially in low dimensional data. We have also compared our algorithm results by applying conventional expectation-maximization (EM) in the same manner. The results show that model-free expectation-maximization clustering was better than conventional expectation-maximization in determining the number of clusters. Clustering accuracy results were very close when the number of clusters were estimated the same in both model-free expectation-maximization and conventional expectation-maximization. The algorithm only missed small clusters with too few samples as there was little statistical evidence for their existence. Furthermore, model-free expectation-maximization clustering was more successful for low dimensional datasets. To overcome this problem, we have combined model-free expectation-maximization clustering with a simulated annealing approach and developed the annealing-based model-free expectation-maximization clustering (ABMFEM), described in Chapter 4.

Simulated annealing is a powerful optimization technique to find the global minimum of a function. Accordingly, we began annealing-based model-free expectation-maximization clustering with a large reference set size that produced flexible decision regions, and decreased the number of samples in the reference set in each iteration to find optimal reference set size while estimating posterior probabilities. The annealing based approach achieved better clustering results on both low and high dimensional datasets. The experiments further showed that small clusters were still at risk of being missed. This indicates that further research is required to identify clusters with few samples, especially in cases when such clusters may potentially carry biological or clinical significance.

Finally, we have developed an automatic compensation procedure that identifies the cell sub groups and jointly diagonalizes them in Chapter 5. The algorithm begins by clustering multi-color flow cytometry data using annealing-based model-free expectation maximization algorithm. To remove the spillover between fluorescence channels, the algorithm finds a transformation matrix that makes each cluster orthogonal over all channels using joint diagonalization with non-orthogonal transformation (FFDIAG) algorithm. The compensated data is clustered one final time to revise the initial cell sub groups. We have also introduced the gamma normalization for transformation of raw intensity measurements as it provides full automation in data transformation and achieves an optimal use of the dynamic range of values. In experiments, we have used only the lymphocyte population of the real multi-color flow cytometry dataset since they have lower autofluorescence compared to other cell types and this suits our orthogonality premise for compensation.

We have tested our algorithm on both synthetically created and real multi-color flow cytometry data in different scenarios and schemes. Firstly, we have identified the three flurochrome channels overlapping maximally in their emission spectra and clustered them on the gamma normalized scale. We have used two different schemes in the orthogonalization: The first one diagonalized cell clusters on gamma normalized scale and the other one orthogonalized the clusters on raw intensity linear domain. Altough these channels overlapped with each other substantially, our algorithm could compensate the data in both scales. In the second scenario, we have identified the three channels with minimal overlap and applied the same procedure again with two schemes. Since the spectral leakage between these fluorochrome channels was small, both the uncompensated and the compensated data look properly compensated. Our compensation algorithm was also successful when we calculated the transformation matrix on the gamma normalized scale. However, calculating the transformation matrix over raw data caused deformations on the identified clusters, as the joint diagonalization algorithm could not produce a matrix that minimized non orthogonal elements of the covariance matrices of the cell clusters due to large intensity values. In the last scenario, we tested our algorithm using scheme 1 on the full 8-color flow cytometry data. The results were satisfactory as all clusters were placed well and aligned to the others over the non-discriminating fluorescence channels. On the other hand, we could not compare our results with manual compensation, because in the flow cytometry experiment, the expert compensated the data according to the experiment needs to uncover interested cell populations, leaving an interested channels properly compensated.

The body of research summarized above describes a way for automatic compensation and gating of multi-color flow cytometry data. One of the problems in this study is identification of small cell groups with few samples. From a statistical standpoint, it is not surprising that such small clusters are missed due to insufficient representation within the overall dataset. However, in applications where small clusters are of particular significance, additional measures are to be taken so that clusters with small representation are also recognized as such. The other problem is automated compensation of the cell types whose autofluorescence is greater than lymphocytes. The performance of the method described here on the other cells, such as monocytes, can be studied further to fill in between the major research components presented in this dissertation

REFERENCES

(1990). Data file standard for flow cytometry. *Cytometry 11*(3), 323–332.

Aghaeepour, N., G. Finak, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, R. H. Scheuermann, F. Consortium, D. Consortium, et al. (2013). Critical assessment of automated flow cytometry data analysis techniques. *Nature methods 10*(3), 228–238.

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

Alberti, S., D. R. Parks, and L. A. Herzenberg (1987). A single laser method for subtraction of cell autofluorescence in flow cytometry. *Cytometry 8*(2), 114–119.

Bagwell, C. B. and E. G. Adams (1993a). Fluorescence spectral overlap compensation for any number of flow cytometry parameters. *Annals of the New York Academy of Sciences 677*(1), 167–184.

Bagwell, C. B. and E. G. Adams (1993b). Fluorescence spectral overlap compensation for any number of flow cytometry parameters. *Annals of the New York Academy of Sciences 677*(1), 167–184.

Baumgarth, N. and M. Roederer (2000). A practical approach to multicolor flow cytometry for immunophenotyping. *Journal of immunological methods 243*(1), 77–97.

Box, G. E. and D. R. Cox (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 211–252.

Brown, D. E. and C. L. Huntley (1992). A practical application of simulated annealing to clustering. *Pattern Recognition 25*(4), 401–412.

Brown, M. and C. Wittwer (2000). Flow cytometry: principles and clinical applications in hematology. *Clinical chemistry 46*(8), 1221–1229.

Bunse-Gerstner, A., R. Byers, and V. Mehrmann (1993). Numerical methods for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications 14*(4), 927–949.

Couri, C. E., M. C. Oliveira, A. B. Stracieri, D. A. Moraes, F. Pieroni, G. M. Barros, M. I. A. Madeira, K. C. Malmegrim, M. C. Foss-Freitas, B. P. Simões, et al. (2009). C-peptide levels and insulin independence following autologous nonmyeloablative hematopoietic stem cell transplantation in newly diagnosed type 1 diabetes mellitus. *Jama 301*(15), 1573–1579.

Craig, F. E., R. R. Brinkman, S. T. Eyck, and N. Aghaeepour (2014). Computational analysis optimizes the flow cytometric evaluation for lymphoma. *Cytometry Part B: Clinical Cytometry 86*(1), 18–24.

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 (methodological)*, 1–38.

Ermolaev, V. and V. Lubimtsev (1987). The jablonski diagram and the electronic-energy relaxation processes from the high excited singlet levels of organic-molecules in solutions. *ACTA PHYSICA POLONICA A 71*(5), 731–741.

Finak, G., A. Bashashati, R. Brinkman, and R. Gottardo (2009). Merging mixture components for cell population identification in flow cytometry. *Advances in Bioinformatics 2009*.

Finak, G., J.-M. Perez, A. Weng, and R. Gottardo (2010). Optimizing transformations for automated, high throughput analysis of flow cytometry data. *BMC bioinformatics 11*(1), 546.

Golub, G. H. and C. F. Van Loan (2012). *Matrix computations*, Volume 3. JHU Press.

Gruetzkau, A. (2010). Fr-fcm-zzwb repository.

Guo, C., H. Fu, and W. Luk (2012). A fully-pipelined expectation-maximization engine for gaussian mixture models. In *Field-Programmable Technology (FPT), 2012 International Conference on*, pp. 182–189. IEEE.

Güven, M. (2010). Detection of man-made structures in aerial imagery using quasi-supervised learning and texture features. Master's thesis, İzmir Institute of Technology.

Hahne, F., N. LeMeur, R. R. Brinkman, B. Ellis, P. Haaland, D. Sarkar, J. Spidlen, E. Strain, and R. Gentleman (2009). flowcore: a bioconductor package for high throughput flow cytometry. *BMC bioinformatics 10*(1), 106.

Horn, R. A. and C. R. Johnson (2012). *Matrix analysis*. Cambridge university press.

Jain, A. K., M. N. Murty, and P. J. Flynn (1999). Data clustering: a review. *ACM computing surveys (CSUR) 31*(3), 264–323.

Karaçalı, B. (2010). Quasi-supervised learning for biomedical data analysis. *Pattern Recognition 43*(10), 3674–3682.

Kirkpatrick, S., C. D. Gelatt, M. P. Vecchi, et al. (1983). Optimization by simulated annealing. *science 220*(4598), 671–680.

Köktürk, B. E. and B. Karaçalı (2012). Automated labeling of electroencephalography data using quasi-supervised learning. In *Signal Processing and Communications Applications Conference (SIU), 2012 20th*, pp. 1–4. IEEE.

Köktürk, B. E. and B. Karaçalı (2013). Quasi-supervised learning on dna regions in colon cancer histology slides. In *Signal Processing and Communications Applications Conference (SIU), 2013 21st*, pp. 1–4. IEEE.

Köktürk, B. E. and B. Karaçalı (2014). Model-free expectation maximization for divisive hierarchical clustering of multicolor flow cytometry data. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, pp. 267–272. IEEE.

Köktürk, B. E. and B. Karaçalı (2016). Annealing-based model-free expectation maximisation for multi-colour flow cytometry data clustering. *International Journal of Data Mining and Bioinformatics 14*(1), 86–99.

Lee, G. (2011). *Machine Learning for Flow Cytometry Data Analysis*. Ph. D. thesis, The University of Michigan.

Lo, K., R. R. Brinkman, and R. Gottardo (2008). Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A 73*(4), 321–332.

Maecker, H. T., J. P. McCoy, and R. Nussenblatt (2012). Standardizing immunophenotyping for the human immunology project. *Nature Reviews Immunology 12*(3), 191–200.

Matton, E. (2014). Automating flow cytometry data analysis using clustering techniques. Master's thesis, University of Gent.

Maulik, U. and S. Bandyopadhyay (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*(12), 1650–1654.

Miller, D. T., B. C. Hunsberger, and C. B. Bagwell (2012). Automated analysis of gpi-deficient leukocyte flow cytometric data using gemstoneâä¢. *Cytometry Part B: Clinical Cytometry 82*(5), 319–324.

Moon, T. K. (1996). The expectation-maximization algorithm. *Signal processing magazine, IEEE 13*(6), 47–60.

Noble, B. and W. Daniel (1977). Applied matrix algebra.

O'Donnell, E. A., D. N. Ernst, and R. Hingorani (2013). Multiparameter flow cytometry: advances in high resolution analysis. *Immune network 13*(2), 43–54.

Onder, D., S. Sarioglu, and B. Karacali (2013). Automated labelling of cancer textures in colorectal histopathology slides using quasi-supervised learning. *Micron 47*, 33–42.

Parks, D. R. and W. A. Moore (2005, October 11). Methods and systems for data analysis. US Patent 6,954,722.

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.

Pyne, S., X. Hu, K. Wang, E. Rossin, T.-I. Lin, L. M. Maier, C. Baecher-Allan, G. J. McLachlan, P. Tamayo, D. A. Hafler, et al. (2009). Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Sciences 106*(21), 8519–8524.

Qian, Y., C. Wei, F. Eun-Hyung Lee, J. Campbell, J. Halliley, J. A. Lee, J. Cai, Y. M. Kong, E. Sadat, E. Thomson, et al. (2010). Elucidation of seventeen human peripheral blood b-cell subsets and quantification of the tetanus response using a density-based method for the automated identification of cell populations in multidimensional flow cytometry data. *Cytometry Part B: Clinical Cytometry 78*(S1), S69–S82.

Quinn, J., P. W. Fisher, R. J. Capocasale, R. Achuthanandam, M. Kam, P. J. Bugelski, and L. Hrebien (2007). A statistical pattern recognition approach for determining cellular viability and lineage phenotype in cultured cells and murine bone marrow. *Cytometry Part A 71*(8), 612–624.

Roederer, M. (2001). Spectral compensation for flow cytometry: visualization artifacts, limitations, and caveats. *Cytometry 45*(3), 194–205.

Roederer, M. (2002). Compensation in flow cytometry. *Current Protocols in Cytometry*, 1–14.

Sa, Y., Y. Feng, K. M. Jacobs, J. Yang, R. Pan, I. Gkigkitzis, J. Q. Lu, and X.-H. Hu (2013). Study of low speed flow cytometry for diffraction imaging with different chamber and nozzle designs. *Cytometry Part A 83*(11), 1027–1033.

Selim, S. Z. and K. Alsultan (1991). A simulated annealing algorithm for the clustering problem. *Pattern recognition 24*(10), 1003–1008.

Shafer, G. et al. (1976). *A mathematical theory of evidence*, Volume 1. Princeton university press Princeton.

Sugár, I. P., J. González-Lergier, and S. C. Sealfon (2011). Improved compensation in flow cytometry by multivariable optimization. *Cytometry Part A 79*(5), 356–360.

Van Der Vorst, H. A. and G. H. Golub (2001). 150 years old and still alive: Eigenproblems. *The state of the art in numerical analysis*, 93–119.

Voltarelli, J. C. (2000). Applications of flow cytometry to hematopoietic stem cell transplantation. *Memórias do Instituto Oswaldo Cruz 95*(3), 403–414.

Ziehe, A., P. Laskov, G. Nolte, and K.-R. MÃžller (2004). A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research 5*(Jul), 777–800.

VITA

Date and Place of Birth: 24.07.1987, Muğla-TURKEY

EDUCATION

2012 - 2017 Doctor of Philosophy in Electrical and Electronics Engineering
Graduate School of Engineering and Sciences, İzmir Institute of Technology,
İzmir -Turkey
Thesis Title: Development of A Comparative Analysis Framework for
High Dimensioanl Data Based on Quasi-supervised Learning
Supervisor: Prof.Dr. Bilge KARAÇALI

2009 - 2012 Master of Science in Electrical and Electronics Engineering
Graduate School of Engineering and Sciences, İzmir Institute of Technology
İzmir -Turkey
Thesis Title: Separation of Stimulus-Specific Brain Activity Patterns
in Electroencephalography Data using Quasi-Supervised Learning
Supervisor: Prof.Dr. Bilge KARAÇALI

2005 - 2009 Bachelor of Electronics and Communication Engineering
Department of Electronics and Communication Engineering, Faculty of Engineering,
İzmir Institute of Technology
İzmir - Turkey

PUBLICATIONS

- Köktürk, Başak Esin and Bilge Karaçalı. ”Annealing-based model-free expectation maximisation for multi-colour flow cytometry data clustering.” International Journal of Data Mining and Bioinformatics 14.1 (2016): 86-99.

- Köktürk, Başak Esin and Bilge Karaçalı. ”Model-free expectation maximization for divisive hierarchical clustering of multicolor flow cytometry data.” Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on. IEEE, 2014.