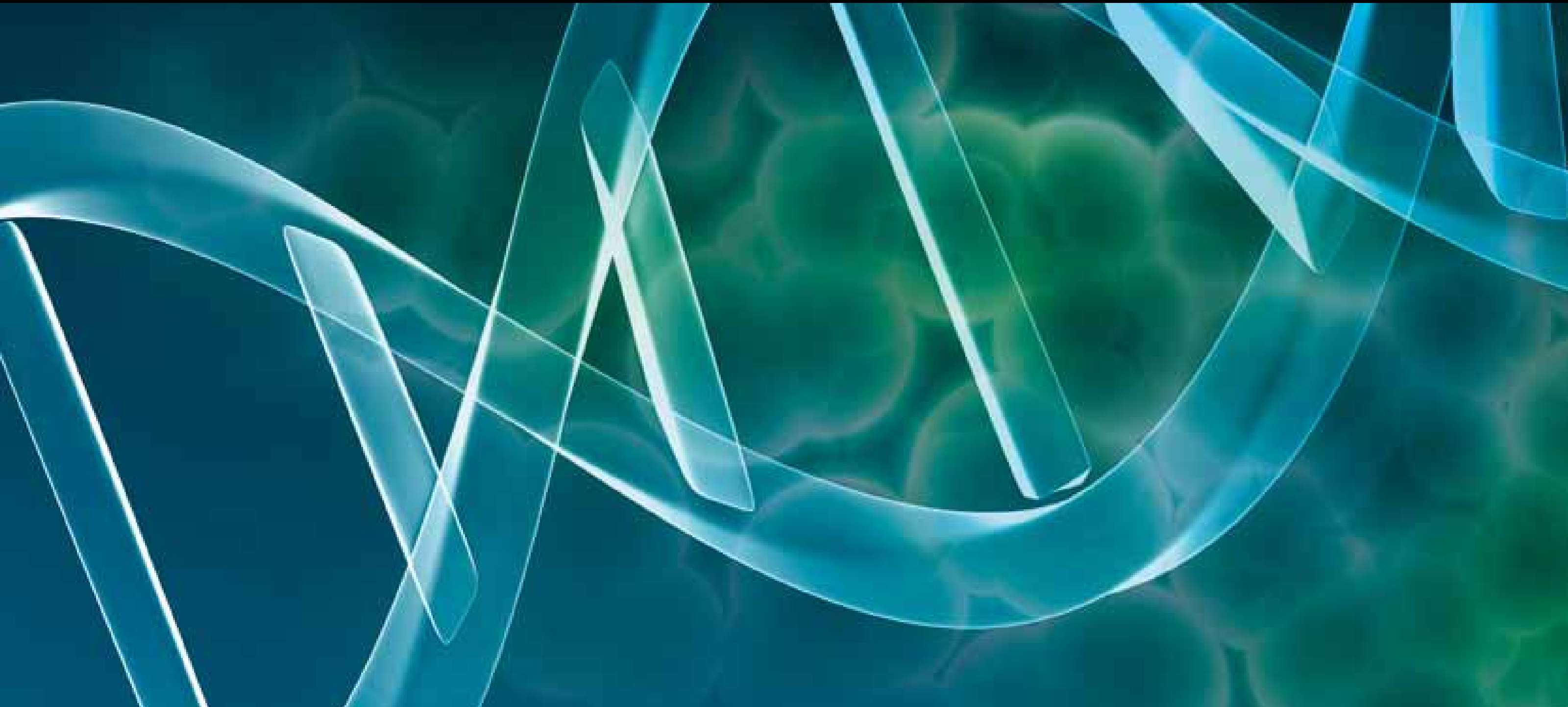
Advances in Bioinformatics

Recent Bioinformatics Advances in the Analysis of High Throughput Flow Cytometry Data

Guest Editors: Raphael Gottardo, Ryan R. Brinkman, George Luta, and Matt P. Wand



Recent Bioinformatics Advances in the Analysis of High Throughput Flow Cytometry Data

Advances in Bioinformatics

Recent Bioinformatics Advances in the Analysis of High Throughput Flow Cytometry Data

Guest Editors: Raphael Gottardo, Ryan R. Brinkman, George Luta, and Matt P. Wand

Advances in Bioinformatics

Editorial Board

Contents

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2009, Article ID 461763, 2 pages
doi:10.1155/2009/461763

Editorial

Recent Bioinformatics Advances in the Analysis of High Throughput Flow Cytometry Data

Raphael Gottardo,[1] Ryan R. Brinkman,[2] George Luta,[3] and Matt P. Wand[4]

[1] *Computational Biology Unit, Clinical Research Institute Montréal (IRCM), 110 Avenue des Pins Ouest, Montréal, Québec, Canada H2W 1R7*
[2] *BC Cancer Agency, 675 West 10th Avenue Vancouver, Canada BC V5Z 1L3*
[3] *Department of Biostatistics, Bioinformatics, and Biomathematics, Georgetown University, Medical Center, 4000 Reservoir Rd., NW Building D, Suite 180 Washington, DC 20057-1484, USA*
[4] *School of Mathematics and Applied Statistics, University of Wollongong, Northfields Avenue, Wollongong 2522, Australia*

Correspondence should be addressed to Raphael Gottardo, raphael.gottardo@ircm.qc.ca

Received 31 December 2009; Accepted 31 December 2009

For more than 30 years, the fluorescence-based technique of flow cytometry (FCM) has been widely used by clinicians, immunologists, and cancer biologists to distinguish different cell types in mixed cell subpopulations, based on the expression of cellular markers. In both health research and treatment, this analytical method is used for a variety of tasks, in particular the diagnosis and monitoring of cancer. This technology is also used for cross-matching organs for transplantation, and for research involving stem cells, vaccine development, apoptosis and phagocytosis.

In the last decade, advances in FCM instrumentation and reagent technologies have enabled simultaneous single cell measurement of surface and intracellular markers, including cellular-activation markers, intracellular cytokines, immunological signaling, and cytoplasmic and nuclear cell cycle and transcription factors, thus positioning FCM to play an even bigger role in health care and medical research.

However, the rapid expansion of FCM applications has outpaced the development of tools for storage, analysis, and data representation. For example, a typical FCM experiment may involve measurement of up to 20 different characteristics per cell, for hundreds of thousands of cells per sample. The increase in the amount of data generated by FCM techniques poses unique informatics and statistical challenges.

It is widely recognized that one basic challenge for FCM is to simplify the extraction of data and statistical information. To date, very few bioinformatic and statistical tools exist to manage, analyze, present, and disseminate FCM data. Current FCM data analysis methods involve the use of multiple applications, the output of which is often fragmented. There is a widespread demand for the development of integrated data analysis tools to organize, analyze, and exchange FCM data. Such development is lagging far behind the ability to collect and process samples via FCM, much to the detriment of health research.

This special issue aims to summarize the current state of bioinformatics research in FCM, to present the most recent developments in analytical tools and to open-up the field to new researchers to bring additional ideas and solutions to current bottlenecks. The issue includes several important contributions, which cover a wide range of approaches and techniques for FCM. These contributions are summarized as follows.

Bashashati and Brinkman review state-of-the-art FCM data analysis approaches that can be used in a typical analysis pipeline going from quality assessment to sample classification. Not only does their paper review current techniques and approaches but it also points out potential pitfalls of these approaches and discusses strategies to overcome these.

Much like with gene expression data, technical variation such as changes in the instrumentation channel voltages or changes in the specificity of the manufacturer of the antibodies can result in systematic biases. These biases need to be removed or at least minimized in order to allow proper data analysis and sample comparisons. Cichocki et al. present a novel normalization method to correct for time biases in large-scale flow cytometric analysis. They investigate two types of normalizing beads: broad spectrum and spectrum matched and propose two alternative normalization procedures that are usable in the absence of normalizing beads.

Once data have been properly normalized, a component of FCM analysis involves identifying immunophenotypically distinct sub-populations of cells within each patient; this is referred to as "gating" in the FCM community. Although gating has traditionally been done visually, automated approaches based on statistical modeling of the data are starting to emerge. Walther et al. present such an approach based on a nonparametric statistical model that aims to form cell subpopulations that can be delineated by the contours of high-density regions much like in manual gating. Because their approach is non-parametric it can reproduce non-convex subpopulations that are known to occur in FCM samples, but which cannot be produced with current parametric model-based approaches. Much like Walther et al., Finak et al. present a framework for the identification of cell subpopulations in FCM data based on merging mixture components using the *flowClust* methodology. In this new approach, several parametric clusters can represent a single sub-population, and the approach can thus accommodate complicated FCM data distributions (e.g., non-convex subpopulations).

Even though automated gating methods are becoming increasing popular, the majority of FCM experiments are still being analyzed visually, usually by serial inspection of one or two dimensions at a time. In order to improve and validate automated gating, it is important to compare automated gates to manual gates obtained by an expert. Gosink et al. introduce a Bioconductor package called flowFlowJo that can import gates defined by the commercial package FlowJo and work with them in a manner consistent with the other flow packages in Bioconductor. This work facilitates examination of gating robustness, allows one to combine manual and automated gating, and can be used to perform exploratory data analysis on manual gates.

Another major goal in clinical applications is the identification of biological changes (e.g., proportion of cells within a subpopulation) that correlate with a disease in order to predict the status (e.g., healthy/diseased) of a patient. Rogers and Holyst present flowFP, a Bioconductor package for fingerprinting flow cytometric data. flowFP provides tools to transform raw FCM data into a form suitable for direct input into conventional statistical analysis and empirical modeling software tools (e.g., supervised classification). Among other things flowFP is based on a multivariate binning approach and thus can bypass the gating stage, which can be an advantage for complex flow data.

In a similar clinical context, Eliot et al. investigate the use of tree-based methods for discovering associations between flow cytometry data and clinical endpoints. In particular, they compare a number of tree-based methods for their capability to select immunological predictors of CD4 reconstitution in HIV-infected subjects initiating anti-retroviral treatment. The authors show that tree-based methods can be successfully applied to flow cytometry data to better inform and discover associations that may not emerge in the context of a standard univariate analysis.

Even though Bioconductor is a great platform for FCM allowing computational statisticians and bioinformaticians to leverage the power of R and other contributed packages, it can remain difficult to be used by biologists and clinicians. Lee et al. have developed an open source, extensible graphical user interface (GUI) iFlow, which sits on top of the Bioconductor backbone, enabling basic analyses by means of convenient graphical menus and wizards. iFlow is easily extensible in order to quickly integrate novel methodological developments.

Finally, Strain et al. introduce plateCore, a new package that extends the functionality of core FCM Bioconductor packages to enable automated negative control-based gating and facilitate the processing and analysis of plate-based data sets from high-throughput FCM screening experiments.

Acknowledgments

The guest editors of this special issue are extremely grateful to all the reviewers who have contributed time and effort to review papers in a timely manner. Their critical reviews ensured that only high-quality papers were selected for publication. The guest editors also thank the authors who have contributed to this special issue and made it possible.

Raphael Gottardo
Ryan R. Brinkman
George Luta
Matt P. Wand

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2009, Article ID 584603, 19 pages
doi:10.1155/2009/584603

Review Article

A Survey of Flow Cytometry Data Analysis Methods

Ali Bashashati and Ryan R. Brinkman

Terry Fox Laboratory, British Columbia Cancer Agency, Vancouver, BC, Canada V5Z 1L3

Correspondence should be addressed to Ali Bashashati, abashash@bccrc.ca

Received 1 May 2009; Revised 20 July 2009; Accepted 22 August 2009

Recommended by George Luta

Flow cytometry (FCM) is widely used in health research and in treatment for a variety of tasks, such as in the diagnosis and monitoring of leukemia and lymphoma patients, providing the counts of helper-T lymphocytes needed to monitor the course and treatment of HIV infection, the evaluation of peripheral blood hematopoietic stem cell grafts, and many other diseases. In practice, FCM data analysis is performed manually, a process that requires an inordinate amount of time and is error-prone, nonreproducible, nonstandardized, and not open for re-evaluation, making it the most limiting aspect of this technology. This paper reviews state-of-the-art FCM data analysis approaches using a framework introduced to report each of the components in a data analysis pipeline. Current challenges and possible future directions in developing fully automated FCM data analysis tools are also outlined.

1. Introduction

Flow cytometry (FCM) is widely used in health research and treatment for a variety of tasks, such as providing the counts of helper-T lymphocytes needed to monitor the course and treatment of HIV infection, in the diagnosis and monitoring of leukemia and lymphoma patients, the evaluation of peripheral blood hematopoietic stem cell grafts, and many other diseases [1–8]. The technology is also used in cross-matching organs for transplantation, research involving stem cells, vaccine development, apoptosis, phagocytosis, and a wide range of cellular properties including phenotype, cytokine expression, and cell-cycle status [9–14]. Clinically, FCM is also used to analyze a wide array of immunological parameters in disease and to study the humoral and cellular response to vaccines.

FCM traditionally has been a tube-based technique limited to small-scale laboratory and clinical studies [15]. Due to recent hardware advances it is now possible to analyze thousands of samples per day. This has dramatically increased the efficiency and use of this technique and allowed the adoption of FCM to high-throughput settings.

It is widely recognized that data analysis is by far one of the most challenging and time-consuming aspects of FCM experiments as well as being a primary source of variation in clinical tests [7, 9, 10, 16–25]. Investigators have traditionally relied on intuition rather than on standardized statistical inference in the analysis of FCM data. The increased volume and complexity of FCM data resulting from the increased throughput greatly boosts the demand for reliable statistical methods and accompanying software implementations, for the analysis of these data [1–6, 16, 20, 23, 26–31]. This is because the ability to analyze FCM data is lagging far behind the ability to collect samples and to run FCM analyses, to the detriment of health research.

This article reviews published approaches for FCM data analysis in the context of a framework created to facilitate the reporting and review process.

2. Background

2.1. FCM Data Analysis. In FCM, intact cells and their constituent components are tagged with fluorescently conjugated monoclonal antibodies and/or stained with fluorescent reagents and then analyzed individually by a flow cytometer. In the instrument, hydrodynamic forces align the cells and the fluorescent molecules in/on each cell are excited by passing through the laser light at speeds exceeding 70 000 cells per second. Each cell passing through the beam also scatters light providing an indication of cell shape and size. A flow cytometer is capable of measuring up to 20 cell characteristics, for up to millions of individual cells per sample aliquot [26, 32]. This technology can be used to examine many cellular parameters on live or fixed cells, including surface, cytoplasmic, and nuclear proteins, DNA, RNA, reactive-oxygen species, intracellular pH, and calcium flux. Measurement of the expression of cellular-activation markers, intracellular cytokines, immunological signaling, and cytoplasmic and nuclear cell cycle and transcription factors can also be readily performed [9, 11, 12, 27, 28, 33–35].

Typical FCM data analysis involves

(1) gating (i.e., identification of homogenous cell populations that share a particular function),

(2) interpretation (i.e., finding (or using) correlations between some characteristics of the identified cell populations (e.g., percentages of cells in a cell population, median fluorescent intensity of a cell population for different markers) and clinical outcomes (e.g., diagnosis, survival).

Gating is a highly subjective process in which the investigators determine the regions in multiparametric space that contain the "interesting" data, based on their knowledge of the experimental factors and experience (Figure 1(a)). This is a tedious, time-consuming, and often inaccurate task typically accomplished using proprietary software provided by instrument manufacturers to serially select regions in one- and two-dimensional graphical representations of the data. Intersections or unions of polygonal regions in hyperspace are then used to filter data and define a subset or subpopulation of events for further analysis (Figure 1(b)). This low-dimensional subsetting ignores the high-dimensional multivariate nature of the data. While a variety of technical issues can confound the accurate positioning of gates, even relatively minor differences in gating can produce different quantitative results [36]. A recent study involving 15 institutions shows that the mean interlaboratory coefficient of variation ranged from 17–44%, even though the same samples and reagents were used and the preparation of samples was standardized. Even though all analyses were conducted by individuals with expertise in flow cytometry, most of the variation was attributed to gating [36].

2.2. Supervised and Unsupervised Learning Techniques. Supervised and unsupervised learning techniques can be used to address the problems faced in gating and interpretation of FCM experiments.

In supervised learning, the variables under investigation can be split into two groups: explanatory variables (e.g., measurements of events in FCM data) and one or more dependent variables (e.g., cell type). The goal here is to predict the labels of the input patterns (e.g., labels of the events in FCM data). This goal can be achieved by discovering an association between the explanatory variables and the dependent variable as is done in regression analysis. Once this association is discovered through the training stage, the algorithm can predict the dependent variable for any event of unknown label. To apply supervised data mining techniques the values of the dependent variable must be known for a sufficiently large part of the data set.

Unsupervised learning is closer to the exploratory spirit of data mining. In unsupervised learning situations all variables are treated in the same way; there is no dependent variable. However, there is still a goal to achieve. In automated gating of FCM data, the goal is to identify the events that are in the same cluster. Clusters contain groups of events that are more similar to each other than the events from other clusters.

The dividing line between supervised learning and unsupervised learning is the same that distinguishes discriminant analysis from cluster analysis. Supervised learning requires that the target variable is well defined and that a sufficient number of its values are given. For unsupervised learning typically the target variable is either unknown or has only been recorded for too small a number of cases.

3. Methods of Survey

FCM data analysis designs selected for this review include papers that met the following criteria.

(1) The keyword "flow cytometry" and one or more of the keywords "automated analysis", "automated gating", and "automated clustering" appeared in its title, abstract, or body using Google Scholar search engine.

(2) The work described one or more automated/semi-automated data analysis components. Papers that presented tutorials were not included. Papers that used manual gating procedures were included only if they employed automated analysis algorithms to analyze gating results. Papers that included simple statistical tests such as Student t-test on manual gating results and the papers that solely applied static gates to FCM data (without any other data processing component) were also not included.

(3) Only papers published in English in refereed international journals prior to March 2009 were included.

We use the framework presented in Section 3.1 to report components involved in FCM data analysis.

3.1. FCM Data Analysis Framework. Figure 2 depicts an FCM data analysis framework in which an FCM data file is analyzed through a series of analysis components. This framework has evolved from the study of FCM literature covered in this article and work in related fields, including statistics and computer science. This framework is constructed to report details of FCM data analysis studies in a systematic way to facilitate reporting and review process. The framework does not incorporate the hardware and software components used for FCM data collection.

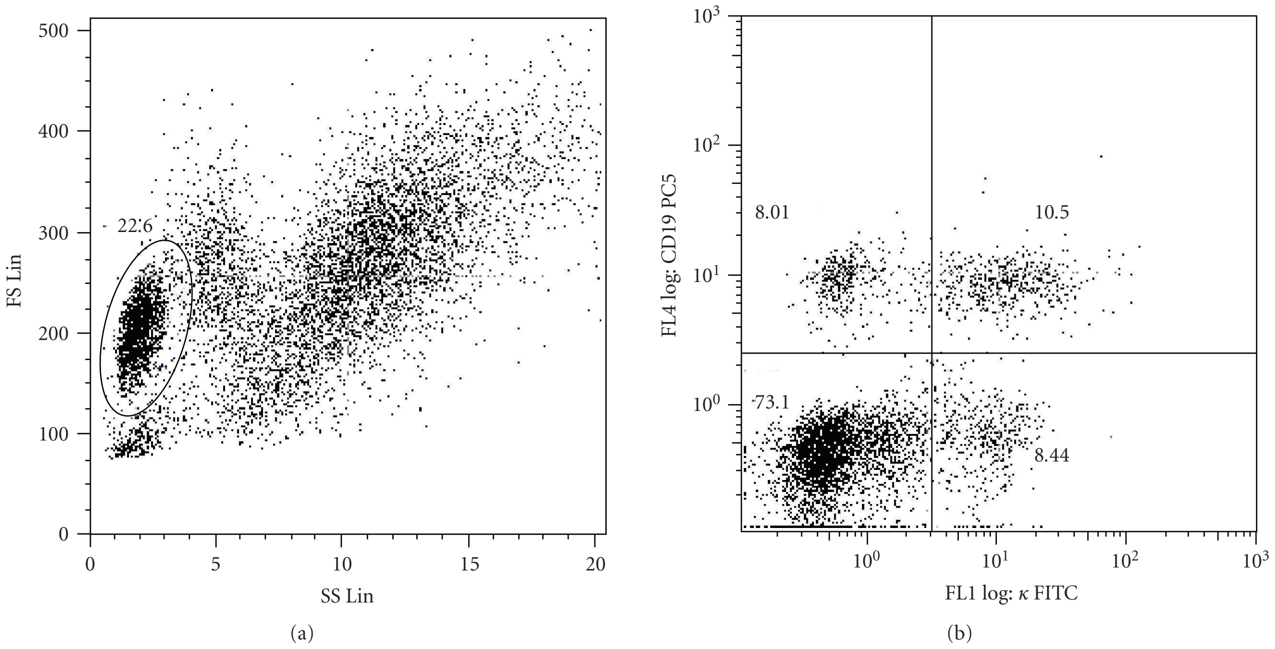


Figure 1: Two-dimensional sequential gating example. (a) Operator selects a subset of "interesting" events (shown within the ellipsoid region), (b) Selected events in (a) are observed and further analyzed using other dimensions of the data. The axes represent different parameters representing physical and chemical characteristics of the analyzed cells.

(1) *Quality Assessment.* Artifacts from sample preparation, handling, variations in instrument parameters, or other factors may confound experimental measurements and lead to erroneous conclusions. Therefore, quality assessment is a crucial step in the use of high-throughput flow cytometry and its associated information services [37–39]. The aim of data quality assessment could include detecting whether intersample variability measurements of samples are not likely to be biologically motivated. Such samples should be identified, investigated, and potentially removed from any further analyses.

(2) *Normalization.* Like all other high-throughput data sources, there is a substantial need for normalization steps to remove nonbiological variations so that the analysis can focus on the important and relevant biological variations between samples. Instrument variability (e.g., changes in laser power), experimental protocol changes (e.g., changes in voltage setting of the instrument), and reagent changes (e.g., using antibodies from different vendors) are examples of nonbiological factors that can introduce variability in the data and shift the location of cell populations. Such changes may affect the analysis of FCM data as the main prerequisite for automated FCM data analysis is a uniform, quantitative, and comparable raw data which can be addressed by developing normalization methodologies.

(3) *Outlier Removal.* Outliers refer to observations (events in the FCM data) that deviate to such a large extent from others so as to arouse suspicion that they do not belong to the same group of observations of interest. Cell debris, dead cells, and doublets (multiple events at the same time) often contaminate FCM data and give rise to outliers. Statistics derived from data sets that include outliers may be misleading. Therefore, it is crucial to identify outliers and account for their prevalence so as to minimize their effect on subsequent analysis.

(4) *Automated Gating.* Automated identification of homogenous cell populations that share a particular function is referred to as automated gating. The main purpose of automated gating is having an objective and systematic approach for classifying cells. Automated gating can be used to

(i) identify known cell populations,

(ii) discover new subpopulations of cells that might not be easily detected via standard manual gating methods. For example, cell populations may be missed due to limitations of two-dimensional manual gating.

(5) *Cluster Labelling.* Comparison of FCM samples is only possible if the same cell populations of different samples are compared against each other. For example, lymphocyte cells of two different samples can be compared against each other but it does not make sense to compare lymphocytes from one sample to granulocytes of another sample. Cluster labelling is referred to the procedure of finding similar cell populations *between* samples after automated gating. Depending on the automated gating approach used, cluster labelling may not be needed as it can be embedded in automated gating

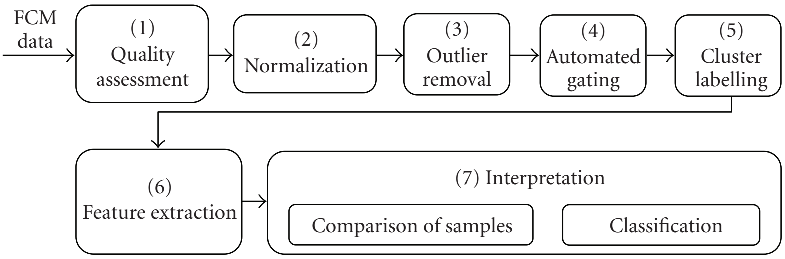


Figure 2: Proposed FCM data analysis framework.

procedure. Note that similar cells *within* each sample are identified through automated gating.

(6) *Feature Extraction.* This step involves computing measurable heuristic properties (also referred to as features) of the identified gates for further analysis. Percentages of cells with respect to the total number of cells, median, and standard deviations of fluorescent intensities of different markers for the events within each gate (or gates of interest) are examples of features that can be computed for the next step.

(7) *Interpretation.* Interpretation of gating results is highly dependent on what the objective of the study is. Usually, there are two major objectives in an FCM-based study: (a) statistical comparison of samples, where the samples are compared to see if they share similar characteristics; (b) classification, where the samples are labeled to predefined classes such as healthy versus patient or patients with short survival versus the ones with long survival time. Depending on the objectives of a study (comparison versus classification), unsupervised or supervised learning techniques can be used.

4. Results

In Table 1, we report the data analysis components of each paper according to the framework presented in Section 3. For the papers that reported multiple designs, multiple classifications were recorded. The designs were categorized based only on what was implemented and reported in each paper. Each column in Table 1 reports the details of each of the components of the FCM data analysis framework, including the following details of each automated gating algorithm

(i) capability of supporting multidimensional gating,

(ii) capability of the algorithm to determine the number of cell populations (gates) automatically,

(iii) whether or not the algorithm belongs to the category of supervised or unsupervised learning techniques.

All the studies covered in this review (except [40, 41]) use percentages of cells within the identified gates and/or median fluorescent intensities of cell populations as the properties (features) of the identified gates for further analysis. Furthermore, a few studies address quality assessment [42–44] and normalization [44] of FCM data. Therefore, for effective use of space, Table 1 does not report the quality assessment, normalization and feature extraction components of the framework for each study.

The entries that contain "E" refer to the term "embedded" meaning that either the cluster labelling, determining the number of cell populations, or outlier detection is embedded in the automated gating algorithm. Studies that did not implement a specific data processing component or do not have a specific capability (e.g., handling multidimensional data) have a "—" entry.

5. Discussion

Although a consensus among researchers for the need of a framework to describe FCM data analysis is not well documented, we feel that it can be a useful tool to facilitate research in this field. A common framework provides a reference, not only for researcher-to-researcher interaction but also for communication to persons in related fields and professions. It will also facilitate technology cross-fertilization, that is, the ability to recognize and integrate significant technological advancements made by others into one's own work. Therefore, during the course of reviewing FCM data analysis literature, we created a framework to report FCM data analysis approaches in a structured way, which facilitates the reporting and review process in the future. Our approach was to create an intuitive framework for organizing and documenting the key data analysis components described in a study and also provide a means to identify the data analysis components that have not been reported. Moreover, the use of this framework makes it easier to understand the differences between different data analysis pipelines.

Table 1 provides a summary of the survey, making it a quick reference to review the results. For example, a quick look at the first row in Table 1 shows the design components used by Jeffries et al. [45] in their analysis of FCM data. Moreover, if somebody is interested in designing or using automated gating approaches, he/she can quickly identify the studies that address automated gating of the FCM data by referring to the third column of Table 1. The proposed framework is flexible enough to encompass the range of data analysis approaches covered in this paper. However,

Table 1: Summary of survey (M: manual; Y: yes; E: embedded in gating; U: unsupervised; S: supervised; "—": not supported, not implemented, not applicable; "||": same as above). Note that this table does not report Quality Assessment, Normalization, and Feature Extraction components.

Paper	Outlier removal	Automated gating				Labelling	Interpretation (classification/ comparison of samples)
		Method	Supervised/ Unsupervised	Multidimensional	Automated # of clusters		
[45]	Logical and cleaning morphological operators applied to the corresponding image representation of FCM data	Logical operation on image representation of FCM data followed by thickening	U	—	—	Based on location and abundance of populations	—
	||	Majority operator applied to the image representation of FCM data followed by Soble edge detection	U	—	—	||	—
	||	Zero-degree B-Spline smoother applied to the 2-dimenisonal FCM data followed by break point detection	U	—	—	||	—
	||	Gath-Geva fuzzy clustering	U	—	—	||	—
[30]	Embedded in clustering (cluster membership weights can be used to exclude outliers)	Gaussian Mixture Models	U	Y	Y (using BIC)	M	—
[46]	Embedded in clustering (cluster membership weights can be used to exclude outliers)	t-Mixture Models	U	Y	Y (using BIC)	M	—
[47]	Embedded in clustering (excluding events that are far from Gaussian functions centers using a predefined cutoff value)	Mahalanobis distance from centroids of multivariate Gaussian functions used for classification task	S	—	—	E	—
[48]	—	Multilayer perceptron (MLP)	S	Y	—	E	—
[49]	—	Building templates for automated gating by using a cluster-finding algorithm (Beckton Dickinson's (BD) snap-to gate algorithm)	U	—	—	E (initially set by operator)	—
[50]	—	DKLL (an extension of the *k*-means algorithm to allow for non-spherical clusters)	U	Y	—	—	—
	—	Fuzzy *k*-means based on adaptive distance	U	Y	—	—	—
	—	Fuzzy *k*-means based on maximum likelihood	U	Y	—	—	—
	—	Fuzzy *k*-means based on minimum total volume	U	Y	—	—	—
	—	Fuzzy *k*-means based on sum of all normalized determinants	U	Y	—	—	—

Table 1: Continued.

Paper	Outlier removal	Automated gating				Labelling	Interpretation (classification/ comparison of samples)
		Method	Supervised/ Unsupervised	Multidimensional	Automated # of clusters		
[51]	—	M	—	—	—	M	Complete linkage hierarchical clustering
[52]	—	—	—	—	—	—	Comparing sample to a reference sample by probability binning algorithm
[53]	—	*k*-means	U	Y	Histogram feature guided	—	—
					Partition index guided		—
[17]	—	Frequency difference gating approach (defines a gate(s) that contains statistically significant more events in the test sample than the control sample)[1]	U	Y	—	—	—
[54]	—	MLP	S	Y	—	E	—
	—	Learning vector quantization (LVQ)	S	Y	—	E	—
	—	Radial basis function (RBF)	S	Y	—	E	—
	—	Asymetric RBF	S	Y	—	E	—
	—	Classification by modeling each class with Gaussian distributions	S	Y	—	E	—
	—	*k*-nearest neighbour method	S	Y	—	E	—
	—	Kohonen's self organizing map (SOM)	U	Y	—	M	—
[55]	—	Static gates applied to data	U	—	—	E (initially set by operator)	CLASSIF1 approach [56, 57]
[36]	—	Building templates for automated gating by using a cluster-finding algorithm (BD Snap-to gate algorithm)	U	—	—	E (initially set by operator)	—
[43][2]	—	M	—	—	—	M	Functional linear discriminant analysis
[58]	—	Building templates for automated gating by using a cluster-finding algorithm (BD's snap-to gate algorithm)	U	—	—	E (initially set by operator)	—
[59]	—	Gaussian Mixture Models	U	—	M	M	—
[60]	—	M	—	—	—	M	Average-linkage hierarchical clustering

Table 1: Continued.

Paper	Outlier removal	Automated gating				Labelling	Interpretation (classification/ comparison of samples)
		Method	Supervised/ Unsupervised	Multidimensional	Automated # of clusters		
[61]	—	M	—	—	—	M	Classification based on a semantic network of knowledge base through a hierarchical tree (if-then rule mechanism)
[62]	—	k-means	U	Y	—	M	—
	—	Calculating modes of density function (calculated by Kernel density estimation) followed by nearest neighbour heuristic	U	Y	—	M	—
	—	Gaussian mixture models using Markov chain Monte Carlo (MCMC)	U	Y	—	M	—
[63]	—	Building templates for automated gating by using a cluster-finding algorithm (BD's snap-to gate algorithm)	U	—	—	E (initially set by operator)	—
[64]	—	Automated gating using BD Simulset software	—	—	—	M	Correlation tests using Spearman's method
[65]	—	Image representation of randomly selected events from a group of flow data followed by smoothing, regional maxima detection and watershed algorithm to define the gates to apply to all the data	U	—	—	—	—
[66]	—	SOM	U	—	—	M	—
	—	Cluster analysis with Winlist (Verity Software House, USA))	U	—	—	||	—
[67]	—	Static gates applied to data and self adjusting gates (details not mentioned) for lymphocytes, monocytes, and granulocytes	U	—	—	E (initially set by operator)	CLASSIF1 approach [56, 57]
[68]	—	Fcom tool (an analysis tool in Winlist (Verity Software House, USA))	—	—	—	M	Average- linkage hierarchical clustering
[69]	—	Static gates applied to data and self adjusting gates for lymphocytes, monocytes, and granulocytes	U	—	—	E (initially set by operator)	CLASSIF1 approach [56, 57]
[70]	—	M	—	—	—	M	"Professor Fidelio" (a heuristic classification system that reasons on the basis of defined diagnostic patterns [71])

Table 1: Continued.

Paper	Outlier removal	Automated gating				Labelling	Interpretation (classification/ comparison of samples)
		Method	Supervised/ Unsupervised	Multidimensional	Automated # of clusters		
[72]	—	*k*-means followed by Murphy's cluster joining algorithm based on standard deviation of the data [73]	U	—	—	M	—
	—	*k*-means followed by a cluster joining algorithm based on modified spread of the data and modified distance between two clusters [72]	U	—	—	M	—
	—	Preclustering a subset of the data by *k*-means and assigning unclustered events to the closest cluster center followed by a cluster joining algorithm based on modified spread of the data and modified distance between two cluster [72]	U	—	—	M	—
[73]	E (excluding the events that were more than a set number of standard deviations away from the centroids of the clusters)	*k*-means followed by Murphy's cluster joining algorithm based on standard deviation of the data	U	Y	—	M	—
[74]	—	MLP	S	Y	—	E	—
[75]	—	RBF	S	Y	—	E	—
[76]	—	MLP	S	Y	—	E	—
	—	SOM	U	Y	—	M	—
	E (excluding the events that were more than a set number of standard deviations away from the centroids of the cluster)	*k*-means	U	Y	—	M	—
[77]	—	No gating—mean fluorescent intensities of antibodies were used for next stage of analysis	—	—	—	—	MLP
[78]	—	RBF	S	Y	—	E	—
[40]	—	—	—	—	—	—	Histogram of one parameter of FCM data followed by MLP
[79]	—	Classification and regression trees (CARTs)	S	Y	—	E	—
[80]	—	Support vector machine (SVM)	S	Y	—	E	—
	—	RBF	S	Y	—	E	—

Table 1: Continued.

Paper	Outlier removal	Automated gating				Labelling	Interpretation (classification/ comparison of samples)
		Method	Supervised/ Unsupervised	Multidimensional	Automated # of clusters		
[81]	—	RBF using radially symmetric basis function (based on Euclidean distance)	S	Y	—	E	—
	—	RBF using more general arbitrarily oriented ellipsoidal basis functions (based on Mahalanobis distance)	S	Y	—	E	—
[82]	—	Gaussian mixture model clustering	U	Y	—	—	—
[83]	Embedded in clustering (excluding events that are far from Gaussian functions centers using a predefined cutoff value)	Mahalanobis distance from the centroids of multivariate Gaussian functions used for classification task	S	—	—	E	—
[84]	—	M	—	—	—	—	Classification based on a shrunken centroids approach [85] Hierarchical clustering
[41]	—	M	—	—	—	—	Kernel density estimation followed by calculating differences between patients by Kulback-Leibler divergence to form a similarity matrix and then dimensionality reduction by multidimensional scaling for 2-dimensional visualization
[86]	—	RBF	S	Y	—	E	—
[87]	—	M	—	—	—	—	Hierarchical clustering Principal component analysis (PCA) for dimensionality reduction and visualization to see if classes are separable by looking at the first few principle components

[1] Closely related to probability binning algorithm introduced in [52].

[2] This study utilizes quality assessment strategy introduced in [42] that is based on comparison of density, ECDF (empirical cumulative distribution function), box plots, and two types of bivariate plots of similar samples.

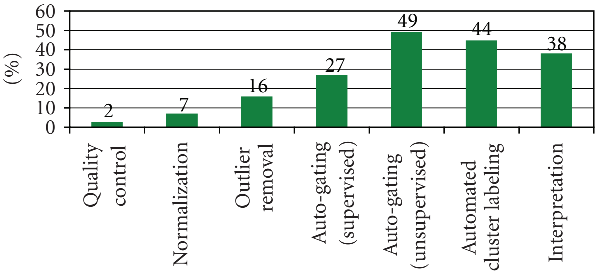


Figure 3: Percentages of studies that address different data analysis components according to the proposed framework. Note that cluster labeling approaches that are embedded in gating stage are counted in the "Cluster Labeling" entry.

refining or expanding it might be necessary in the future. For example, even though a feature selection component was not needed to describe current FCM data analysis studies, addition of this component might be necessary in future. Feature selection is specifically important as it can discard the uninformative and also redundant features, facilitate data visualization and data understanding, reduce the measurement and storage requirements, reduce training and utilization times, and defy the curse of dimensionality to improve prediction performance [88].

Figure 3 shows the percentages of the studies that have addressed each of the data analysis components according to the proposed framework.

As shown in Figure 3, most of the studies (more than 70%) focus on automated gating of FCM data from which 65% use unsupervised techniques and 35% use supervised techniques. However, only few studies focus on quality control and normalization of FCM data, suggesting that more work might still be needed in the future.

In the rest of this section we specifically discuss the FCM data analysis methods that have been used in the context of the framework introduced in Section 3.1.

5.1. Quality Assessment. The basis of the quality assessment method proposed in [42, 43] is that, given a cell line, or a single sample, divided in several aliquots, the distribution of the same physical or chemical characteristics (e.g., side light scatter (SSC) or forward light scatter (FSC)) should be similar between aliquots. To test this hypothesis, five distinct visualization methods were implemented to explore the distributions and densities of ungated FCM data: Empirical Cumulative Distribution Function (ECDF) plots, histograms, boxplots, and two types of bivariate plots. Hahne et al. [44] also propose a set of visualization tools to inspect box plots of fluorescent values, number of cells, and a measure defined as "odds ratio" for similar samples within a plate. These different graphical methods provide investigators with different views of the data and can quickly flag the samples that are different from the rest. As the flagged samples may be anomalous for biological reasons, these samples are worth studying further, and some determination as to whether the sample presents data quality issues or rather presents real biological significance should be made [42].

Problems with the cell suspension, clogging of the needle, or similar issues can cause unusual patterns in the data. flowQ R package [89] addresses such problems by developing several approaches that detect disturbances in the flow of cells and also detect unusual patterns in the acquisition of fluorescence and light scatter measurements over time. These are detected dynamically by identifying trends in the signal intensity over time or local changes in the measurement intensities. The underlying hypothesis is that measurement values are acquired randomly; hence there should not be any correlation to time. Other quality assessment strategies may include investigating the number of events or the number of live cells within a sample. Furthermore, specific statistical tests addressing quality assurance requirements of an experiment can be developed. For example, in the FCM experiments to monitor clonal repopulation of engrafted single cell hematopoietic stem cells in mice [90, 91], blood samples are taken and divided into three aliquots. Each aliquot is stained with cocktail specific for detecting granulocytes/monocytes, B cells, and T cells. The percentages of each cell type from the donor population should add to roughly 100%; otherwise possible problems with the staining or the gating have occurred. Using such criterion, automated quality assurance tools can be developed to identify possible problems in the experiments.

5.2. Normalization. The only study that touches on the normalization issue of FCM data proposes a method of normalizing all channels, using a model based on the size (FSC channel) of the events [44]. The authors show in their experiment that the increase in autofluorescence associated with cell size needed to be adjusted for and developed a specific linear model for this adjustment. Nonbiological variations can cause a shift or rotation in absolute position of cell populations. Figure 4 shows an example in which the voltage of the flow cytometer has changed in the channel that measures CD3 expression between the two experiments causing the population marked within the ellipsoid gate to move substantially (more than 10-fold change in median fluorescent intensity). Such variations should be accounted for during data analysis as they can cause misinterpretation of the results. For example, an ellipsoidal gate defined based on the data shown in Figure 4(a) would not capture the population of interest shown in Figure 4(b) even though the two populations represent the same cell types. While significant further developments to normalize FCM data are needed, care should be taken, as biologically motivated variations should be conserved while removing nonbiological variations.

5.3. Outlier Removal. Outliers can have a significant effect on automated gating results. For example, in unsupervised techniques, they can lead to overestimating the number of cell populations (i.e., clusters present in the data) needed to provide a good representation of the data. Moreover, data contaminated with outliers, when used as example data to train a supervised technique, can affect decision boundaries of the algorithm leading to poor gating results.

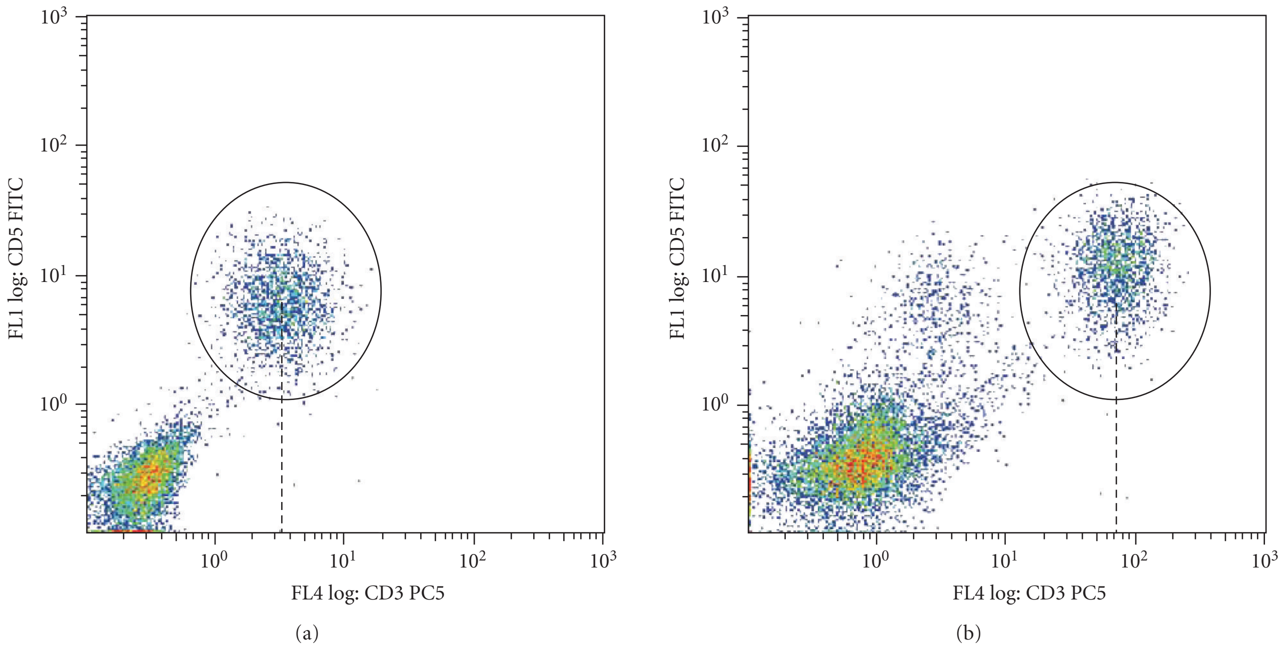


Figure 4: (a) and (b) Example of cases where flow cytometer voltage changes have caused in a shift in the absolute position of the populations within the ellipsoid gates.

Outliers can be handled in a number of ways depending on the learning technique being used. For example, in the model-based clustering framework [92, 93], they can be handled by either replacing the Gaussian distribution with a more robust one (e.g., *t* [94]) or adding an extra component to model the outliers (e.g., uniform [92]). Lo et al. [46] used a *t*-distribution in the context of model-based clustering to deal with outliers in FCM data. Jeffries et al. [45] represent two-dimensional FCM data as an image and apply a set of morphological operators on the corresponding image to remove outliers. Although Jeffries' study concentrates on two-dimensional data, the operators are applicable to multidimensional data as well. Cluster membership weights calculated during automated gating may also be used for outlier identification [30, 46]. When using supervised learning techniques, suspected examples can be removed from the learning phase to improve the generalization performance of the learning algorithm [95]. Furthermore, assigning decision confidence together with the labels of each event can be utilized to exclude the events that are less likely to belong to a specific class (e.g., [96–98]).

5.4. Automated Gating. More than 70% of the studies covered in this review have implemented approaches for automated gating of the FCM data. In the following subsections, we focus on these approaches in more detail. Although the approaches covered in these sections are implemented for automated gating purposes, most of them are applicable to interpretation stage of data analysis as well.

5.4.1. Supervised Techniques for Gating. Supervised techniques require training data and a training phase to learn the relationship between the events and output classes but unsupervised ones do not need this. Selection of training data that is representative of all cell populations of interest is important in training supervised techniques. Supervised techniques usually classify the input events to one of the predefined cell populations introduced to the algorithm in the training stage. Therefore, if a novel cell population exists in the data, the algorithm classifies that population as belonging to one of the predefined cell populations and not as a novel population. Two strategies can overcome this problem to some extent.

(i) The first one is assigning an "unknown" class for the input patterns that are unlikely to belong to known event categories [79, 96, 98]. A disadvantage of this solution is that if two novel categories exist in the test data, both will be classified as unknown even though the unknown class is comprised of multiple novel classes. It is, however, possible to add another stage of processing to further investigate the unknown events to see if they consist of multiple populations. Another similar solution would be to assume that each event can belong to several classes with different membership (e.g., event one belonging to "Class 1" with 70% chance and to "Class 2" with 30% chance) or to assign decision confidence for each classified event and reject less confident classifications as outliers or unknowns [96–98]. Using such a strategy, Wilkins et al. [75] show that more than 70% of novel species were successfully identified as "unknown" while the proportion of correctly classified species decreased moderately (from 93.8% to 86.8%) compared to the case when no novel species were identified.

(ii) The second approach used by Beckman et al. [79] suggests adding fictitious events that reside in some of the empty spaces. Input events that are close to these fictitious events are classified as unknown events rather than being classified as belonging to the populations of interest [79]. This approach, however, needs extensive intervention in the data space in order to generate populations that represent unwanted event types. Moreover, this task is impractical when the dimension of the data is high, as one needs to generate fictitious data points that represent different unknown categories throughout the whole data space [99].

Overall, supervised techniques are suitable for tasks where we know how many classes exist in the data and a choice of unknown class would exclude the events that do not belong to the classes of interest. On the other hand, unsupervised techniques are more suitable for novel class discovery tasks.

In supervised learning techniques, the training set should be a good representative of the future unseen data. Therefore, reproducible FCM data is necessary. For example, if there is excessive drift in the centroids of the cell populations, many of the cells could be misclassified. Some minor amount of drift can be usually accommodated by the algorithm itself and also having training sets composed of samples measured at different times for different individuals [40]. One approach to overcome this problem is to normalize the data before gating.

Care should be taken when using supervised techniques, as usually unequal numbers of training patterns of each class are available, and this can bias the training of the classifier towards the classes with higher number of training events. One solution that has been suggested and applied to FCM data is to take into account a posteriori probabilities and class probabilities (i.e., the proportion of each of the cell categories in the training data) [86, 99, 100].

During training, a supervised learning algorithm reaches a state where, given sufficient and informative data, it should be capable of predicting the correct label for unseen data. However, the algorithm may adjust itself to very specific features of the training data that have little relation to unseen data. In this process referred to as overfitting, the performance on the training examples is high while the performance on unseen data becomes worse. Roughly speaking, an algorithm that is overfit is like a botanist with a photographic memory who, when presented with a new tree, concludes that it is not a tree because it has a different number of leaves from anything he/she has seen before [101]. Overfitting can be avoided by employing techniques such as regularization and early stopping [102–104].

Regularization involves introducing a form of penalty for complexity of the classification model. An example of regularization in neural networks is *weight decay* algorithm used in MLP neural networks. As large weights can decrease the performance of an MLP classifier on unseen data, *weight decay* penalizes the large weights causing the weights to converge to smaller absolute values than they otherwise would [102]. This strategy has been used in the context of gating FCM data [77].

In early stopping, the available training data is divided into two sets, that is, a *new training* set and a validation set. In each iteration of learning, the data of the *new training* set is used to train the learning algorithm and the validation set is used to evaluate its performance. The learning phase is forced to stop once the performance on the validation set does not improve or degrades. This method can be used either interactively (based on human intervention) or automatically (based on some stopping criteria usually chosen in an adhoc fashion). As mentioned in [105], early stopping is widely used as it is easy to implement and has been reported to be superior to regularization methods in many cases (e.g., [106]).

A number of algorithms in the category of supervised techniques such as multilayer perceptron (MLP) networks (e.g., [48, 54]), radial basis function (RBF) networks (e.g., [54, 75]), and support vector machines (SVM) [80] have been used in the context of cell population identification in FCM data.

A typical MLP network consists of a set of nodes forming the input layer, one or more hidden layers, and an output layer. The MLP network has a highly connected topology since every input node is connected to all nodes in the first hidden layer, every node in the hidden layers is connected to all nodes in the next layer, and so on. The value of each node is determined by a weighted combination of input nodes, possibly including some nonlinear activation function.

An MLP network is trained by repeated presentation of input patterns to the network. During the training process, small iterative weight changes in the structure of the network are performed until the predicted outputs are considered close enough to desired outputs. Designing an MLP classifier is not a trivial task as one needs to determine optimal parameters of the MLP structure (e.g., number of hidden layers, number of hidden layer nodes, etc.) for each specific classification task. For most problems, one hidden layer is sufficient. Using two hidden layers rarely improves the model, and it may introduce a greater risk of converging to a local minima. The network may not be able to model complex data if inadequate number of hidden layer nodes is used. On the other hand, if too many nodes are used, the training time may become excessively long, and the network may overfit the data. In general, training an MLP is relatively slow and sometimes the algorithm gets stuck in local minima and therefore the training process has to be restarted [104]. It has been shown that if an accuracy of $(1-e)$ on a test set is desirable, the number of events in the training set, p, should satisfy $p \geq w/e$, where w is the total number of weights in the network [107]. Hence, to obtain 90% accuracy ($e = 0.1$) on test set, the desirable number of events required in training set is at least ten times the total number of weights. While having $p \geq w/e$ is definitely desirable, it is sometimes difficult in practice to build such a large database of clinical cases. An option is to use a perturbation method to generate a large number of cases by introducing small variations in actual cases [77]. The importance of having sufficiently large training sets to cover biological variation is highlighted by the increase in overall identification success of different marine microalgae in an FCM study [86].

An RBF neural network typically is comprised of three layers of nodes (i.e., input, hidden and output layers). The neurons in the hidden layer contain basis functions, *usually* Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the basis function. Normally the Euclidean distance is used as the distance measure, although other distance functions are also possible. An RBF network output is formed by a weighted sum of the hidden layer neuron outputs and the unity bias.

The parameters of an RBF network which are determined in the training stage consist of the positions of the basis function centers, the radius (spread) of the basis functions in each dimension, the weights in output sum applied to the hidden layer nodes outputs as they are passed to the summation layer, the parameters of the linear part, and so forth.

Various methods have been used to train RBF networks. One approach first uses k-means clustering to find cluster centers which are then used as the centers for the RBF functions. However, k-means clustering is a computationally intensive procedure, and it often does not generate the optimal number of centers. Another approach is to use a random subset of the training points as the centers.

Assuming that the data is linearly separable, among the infinite number of hyperplanes that separate the data, an SVM classifier picks the one that has the smallest generalization error. Intuitively, a good choice is the hyperplane that leaves the maximum margin between the two classes, where the margin is defined as the sum of the distances of the hyperplane from the support vectors. Support vectors are the examples closest to the separating hyperplane and the aim of an SVM classifier is to orientate this hyperplane in such a way that it is as far as possible from the closest members of both classes. If the two classes are nonseparable we can still look for the hyperplane that maximizes the margin and that minimizes a quantity proportional to the number of misclassification errors. The trade-off between margin and misclassification error is controlled by a positive constant C (referred to as error penalty) that has to be chosen beforehand [101, 108].

SVMs are very universal learners. In their basic form, SVMs learn linear threshold function. Nevertheless, by a simple “plug-in” of an appropriate kernel function, they can be extended to nonlinear classifiers such as polynomial classifiers, radial basis function (RBF) networks, and three-layer sigmoid neural networks.

Perhaps the biggest limitation of the SVM approach lies in the choice of the kernel. Once the kernel is fixed, SVM classifiers have only one user-chosen parameter (the error penalty) [101].

RBF networks can be trained significantly faster than MLPs. In addition to the number of hidden layers, a difference between RBF and MLP classifiers lies in the nodes of the hidden layer, which use different kernels (basis functions) to represent the data. RBF networks have the advantage of not suffering from local minima in the same way as MLPs. While for an RBF there is no restriction on decision boundaries formed, an MLP forms convex decision boundaries. Moreover, RBF’s hidden layer performs a nonlinear mapping from the input space into a (usually) higher-dimensional space in which the input patterns become linearly separable [109]. Although RBF networks are quick to train, when training is finished and it is being used, it is slower than an MLP. Therefore, where speed is a factor an MLP may be more appropriate.

SVM can be seen as a new way to train polynomial, neural network, or RBF classifiers. While most of the techniques used to train the above mentioned classifiers are based on the idea of minimizing the training error, which is usually called *empirical risk,* SVMs operate on another induction principle, called *structural risk minimization,* which minimizes an upper bound on the generalization error [108].

In the context of FCM data analysis, Boddy et al. [81] compares the performances of RBF networks using different basis functions. Specifically, radially symmetric and a more general arbitrarily oriented ellipsoidal basis functions were employed, with the latter proving to be significantly superior in performance. The distance between input patterns and the basis function centers are defined by a distance metric, which determines the shape of the basis function. The Euclidean distance metric produces hyperspherical (radially symmetric) basis functions around the basis functions centers. Mahalanobis distance metric, on the other hand, allows the hyperellipsoid (nonradially symmetric) to adopt any orientation that best fits the data distributions.

Wilkins et al. [54] compare several classification algorithms such as MLP, RBF, and LVQ (learning vector quantization) to identify phytoplankton species from FCM data. The authors show that identification success was more or less similar using the above-mentioned techniques. Therefore, they suggest using the criteria mentioned earlier and characteristics of the data at hand to decide which method is the best to use. In another study on phytoplankton species, Morris et al. [80] demonstrate that an SVM classifier outperforms RBF classification. These studies focus on specific data sets and their generalization on other data sets is unknown. Therefore, picking an algorithm based on the type of data at hand and above-mentioned characteristics of learning algorithms is recommended. One approach that might be worth considering in FCM studies is the multiple classifier systems (MCSs) [110]. MCSs are based on combining the outputs of ensembles of different classifiers (supervised learning techniques). Classification accuracy improvements are possible provided that a suitable combination function is designed and that the individual classifiers make different errors. Ideally, a combination function should take advantage of the strengths of individual classifiers, avoid their weaknesses, and improve classification accuracy [110].

5.4.2. Unsupervised Techniques for Gating. Algorithms for unsupervised analysis of FCM data should be

(i) computationally efficient as the amount of data generated for each FCM experiment is large (an FCM experiment contains measurements for up to millions of cells for up to 20 parameters),

(ii) able to detect clusters with different shapes as clusters (cell populations) in FCM data can have different shapes ranging from spherical shapes to irregular shapes such as being highly elongated or even being curved,

(iii) able to detect populations with different densities and percentages as FCM samples can contain a wide range of cell populations in terms of the density of cells (very sparse vs. very dense cell populations) and also percentages of cells in each population (populations of interest as low as 0.1% of total events),

(iv) able to determine the number of cell populations as the number of cell populations present in the data is usually not known apriori,

(v) able to handle outliers as data can contain significant number of outliers.

The above-mentioned characteristics of FCM data make unsupervised analysis challenging as existing clustering algorithms either do not address or have limitations in addressing these requirements.

Clustering algorithms require the number of clusters that they should identify to be specified apriori. There are several approaches for choosing the number of clusters, including resampling, cross-validation, and various information criteria [111]. Zeng et al. [53] use the peaks of density distribution of each channel of FCM data and estimate the numbers of clusters to be identified by *k*-Means algorithm. Lo et al. [46] propose to use Bayesian information criteria (BIC) in the context of a model-based clustering approach to estimate the optimal number of clusters. BIC is computationally cheap to compute once maximum likelihood estimation for the model parameters has been completed, an advantage over other approaches, especially in the context of FCM where datasets tend to be very large. While computationally cheap, BIC relies heavily on an approximation of marginal likelihoods, which might not be very accurate for some data. Alternative approaches such as the integrated completed likelihood [112] may improve the estimation of the number of clusters. Nevertheless, combined with expert knowledge, such approaches can provide guidance on choosing a reasonable starting number of clusters.

Sometimes it is possible that even if the actual number of clusters is known, the clustering algorithm may not identify the correct clusters at the level of separation that is desired. This can happen when there is a rare cell population within the FCM data. In this case, the clustering algorithm may consider the rare population as an outlier or as part of a larger cell population and instead divide larger cell populations into smaller populations. One approach to overcome this problem might be clustering the data with higher number of clusters with the hope that the rare populations are represented by separate clusters and use some merging algorithm to combine the clusters that are similar according to a criterion.

k-means clustering algorithm is one of the methods that have been used in literature. While this approach performs well when the clusters are spherical in shape, clusters in FCM data usually are not spherical. Demers et al. [82] have proposed an extension of *k*-means allowing for nonspherical clusters, but this algorithm has been shown to lead to inferior performance compared to fuzzy *k*-means clustering [50]. In fuzzy *k*-means [113], each cell can belong to several clusters with different association degrees, rather than belonging to only one cluster. Even though fuzzy *k*-means takes into consideration some form of classification uncertainty, it is a heuristic-based algorithm and lacks a formal statistical foundation. Other choices include hierarchical clustering algorithms (e.g., linkage or Pearson coefficients method). However, these algorithms are not appropriate for FCM data, since the size of the pairwise distance matrix increases in the order of n^2 with the number of cells, unless they are applied to some preliminary partition of the data [72], or they are used to cluster across samples, each of which is represented by a few statistics aggregating measurements of individual cells [87, 114]. Since the required processing time for some clustering algorithms increases significantly by the increase in the number of events and parameters of FCM data, subsampling the data might be a suitable approach to reduce the processing time. Care should be taken when performing subsampling to make sure that the properties of the original data are preserved after this process. For example, a random uniform sampling of data may not be a suitable approach as it can discard the small populations present in the data. One alternative might be using a guided sampling approach in which representative events are selected from low-density populations as well. This might be achieved by different strategies such as looking at density distributions of the data or performing a coarse clustering before subsampling procedure.

An alternative approach for FCM data gating is to model the FCM data with mixtures of distributions. The most commonly used model-based clustering approach is based on finite Gaussian mixture models [93, 115]. However, Gaussian mixture models rely on the assumption that each component follows a Gaussian distribution, which is often not the case when modeling FCM data. A common approach is to look for transformations of the data that make the normality assumption more realistic. Lo et al. [46] proposed the use of the Box-Cox [116] transformation prior to using a model-based clustering. In addition to nonnormality, there is also the problem of outlier identification in mixture modeling. As mentioned earlier, replacing the Gaussian distribution with a more robust one (e.g., *t* [94, 115]) or adding an extra component to model the outliers (e.g., uniform [92]) is suggested to deal with outliers. The *t*-distribution is similar in shape to the Gaussian distribution with heavier tails and thus provides a robust alternative [117]. The Box-Cox transformation is a type of power transformation, which can bring skewed data back to symmetry, a property of both the Gaussian and *t*-distributions. In particular, the Box-Cox transformation is effective for data where the dispersion increases with the magnitude, a scenario not uncommon to FCM data [46].

One of the benefits of model-based clustering approach is that it provides mechanism for both “hard” clustering (i.e., the partitioning of the whole data into separate clusters) and fuzzy clustering (i.e., a "soft" clustering approach in which each event may be associated with more than one cluster) [46]. The latter approach is in line with the rationale that there exists uncertainty about to which cluster an event should be assigned.

5.5. Cluster Labelling. Cluster labelling (or cluster matching) between samples is usually performed manually. Approaches that can label the clusters based on their location such as mean or median fluorescent intensity (MFI) of known cell populations or their location relative to other clusters have been used in literature [45]. Cluster labelling approaches that take into account the shape and rotation of cell populations in addition to their locations might provide more robust results. In case of using the absolute location of cell populations for cluster labelling, data normalization prior to labelling is necessary as significant changes in the location of cell populations (as shown in Figure 4) can result in mismatching cell populations. Note that in case of using supervised techniques for automated gating, labelling is not needed as the gating algorithm determines the labels of events (e.g., whether the events are of cell type 1 or cell type 2). Therefore, this information can be used for labelling (matching) cell populations between samples as well.

5.6. Feature Extraction. Prior to interpretation of gating results, features representing the identified cell populations need to be defined. In literature, usually the percentages and locations of cell populations are used for interpretation purposes. However, other characteristics of cell populations such as their shapes (e.g., whether they are spherical or ellipsoidal), dispersion, orientation, and proportion of a specific cell population relative to another cell population may also be useful to achieve better interpretation results. Since the features that may carry information are not always known apriori, one option is to generate as many features as possible and then use feature selection techniques to discard the uninformative and also redundant features.

Furthermore, approaches such as the one introduced in [41] that uses other representations of the characteristics of the FCM data (characteristics based on kernel density estimation in the case of [41]) might be interesting to investigate further. Since the final aim in some studies such as the one presented in [41] is to perform a classification task (e.g., healthy versus patient), gating FCM data may not be necessary (except to find basic cell populations such as live cells and lymphocytes) which can potentially eliminate the errors that can be introduced in the system by poor gating strategies.

5.7. Interpretation. Although mostly done manually, interpretation of results can utilize many methods that have been developed in computer science for finding associations between FCM samples with their labels (e.g., disease diagnosis) or identifying cluster of patients with similar FCM data. Depending on the purpose of the study, supervised or unsupervised learning techniques can be used. For example, if the aim is to classify a sample as disease or healthy, supervised learning techniques can be used. For the purpose of finding patients who have similar data, standard unsupervised learning techniques can be utilized.

6. Conclusions

The need for completely automated analysis of FCM data is becoming more evident with the advances in high-throughput FCM technology. To date, most research has been focused on developing approaches for automated gating of FCM data. Manual gating is recognized as labor intensive, subjective, and prone to error when processing large numbers of samples. Therefore, automated gating methods will allow for a faster and more robust data analysis pipeline. Although significant effort is still needed to develop automated gating algorithms that address challenging aspects of FCM data, we believe that the research community needs to look beyond automated gating and develop bioinformatics tools that facilitate building *completely* automated FCM data analysis pipelines. It should be noted that the development of robust, automated methods for high-throughput FCM data analysis also requires high-quality data to feed into the analysis framework. Generating this high-quality data requires well-designed experiments with the appropriate positive and negative controls.

A rigorous quantitative assessment is important before using automated approaches in practice, as a replacement for expert manual analysis. Moreover, it is likely that one data analysis solution may not be suitable to address specific questions of a study or address the challenges of analyzing a specific FCM dataset. For example, if somebody is interested in identifying a previously known type of cell, supervised techniques might be better suited. Overall, in order to use automated data analysis approaches in biomedical research and clinical setting, we need to develop more generic solutions or design smart algorithms that can tune themselves with little intervention, as the users may not have enough knowledge of bioinformatics techniques. The availability of a wide variety of example data is crucial, as it would aid in the development, evaluation, and comparison of different automated analysis methodologies.

The development of automated FCM data analysis approaches will greatly facilitate both basic research and clinical applications in medical/agricultural areas that depend upon this technique. Since FCM generates data sets as complex and informative as gene arrays using markers for different cell populations defined by phenotypic, activation, or cytokine expression features, optimizing FCM-based data analysis will also help develop FCM as a proteomics and diagnostic tool with widespread applications in both basic and clinical laboratories.

Acknowledgments

The authors would like to thanks Aaron Barsky, Alireza Hadj-Khodabakhshi, and Adrian Cortes for their comments on the manuscript. This work was supported by the National Institute of Health (R01 EB005034), Natural Science and

Engineering Research Council of Canada, and the Michael Smith Foundation for Health Research.

References

[1] R. C. Braylan, "Impact of flow cytometry on the diagnosis and characterization of lymphomas, chronic lymphoproliferative disorders and plasma cell neoplasias," *Cytometry A*, vol. 58, no. 1, pp. 57–61, 2004.

[2] R. L. Hengel and J. K. A. Nicholson, "An update on the use of flow cytometry in HIV infection and AIDS," *Clinics in Laboratory Medicine*, vol. 21, no. 4, pp. 841–856, 2001.

[3] O. C. Illoh, "Current applications of flow cytometry in the diagnosis of primary immunodeficiency diseases," *Archives of Pathology and Laboratory Medicine*, vol. 128, no. 1, pp. 23–31, 2004.

[4] F. L. Kiechle and C. A. Holland-Staley, "Genomics, transcriptomics, proteomics, and numbers," *Archives of Pathology and Laboratory Medicine*, vol. 127, no. 9, pp. 1089–1097, 2003.

[5] F. F. Mandy, "Twenty-five years of clinical flow cytometry: AIDS accelerated global instrument distribution," *Cytometry A*, vol. 58, no. 1, pp. 55–56, 2004.

[6] A. Orfao, F. Ortuño, M. de Santiago, A. Lopez, and J. San Miguel, "Immunophenotyping of acute leukemias and myelodysplastic syndromes," *Cytometry A*, vol. 58, no. 1, pp. 62–71, 2004.

[7] C. B. Bagwell, "DNA histogram analysis for node-negative breast cancer," *Cytometry A*, vol. 58, no. 1, pp. 76–78, 2004.

[8] M. Keeney, J. W. Gratama, and D. R. Sutherland, "Critical role of flow cytometry in evaluating peripheral blood hematopoetic stem cell grafts," *Cytometry A*, vol. 58, no. 1, pp. 72–75, 2004.

[9] P. O. Krutzik, J. M. Irish, G. P. Nolan, and O. D. Perez, "Analysis of protein phosphorylation and cellular signaling events by flow cytometry: techniques and clinical applications," *Clinical Immunology*, vol. 110, no. 3, pp. 206–221, 2004.

[10] H. Maecker and V. Maino, "Flow cytometric analysis of cytokines," in *Manual of Clinical Laboratory Immunology*, ASM Press, Washington, DC, USA, 6th edition, 2002.

[11] P. Pozarowski and Z. Darzynkiewicz, "Analysis of cell cycle by flow cytometry," *Methods in Molecular Biology*, vol. 281, pp. 301–311, 2004.

[12] P. Pala, T. Hussell, and P. J. M. Openshaw, "Flow cytometric measurement of intracellular cytokines," *Journal of Immunological Methods*, vol. 243, no. 1-2, pp. 107–124, 2000.

[13] I. Vermes, C. Haanen, and C. Reutelingsperger, "Flow cytometry of apoptotic cell death," *Journal of Immunological Methods*, vol. 243, no. 1-2, pp. 167–190, 2000.

[14] A. K. Lehmann, S. Sørnes, and A. Halstensen, "Phagocytosis: measurement by flow cytometry," *Journal of Immunological Methods*, vol. 243, no. 1-2, pp. 229–242, 2000.

[15] Y. D. Mahnke and M. Roederer, "Optimizing a multicolor immunophenotyping assay," *Clinics in Laboratory Medicine*, vol. 27, no. 3, pp. 469–485, 2007.

[16] D. Redelman, "CytometryML," *Cytometry A*, vol. 62, no. 1, pp. 70–73, 2004.

[17] M. Roederer and R. R. Hardy, "Frequency difference gating: a multivariate method for identifying subsets that differ between samples," *Cytometry*, vol. 45, no. 1, pp. 56–64, 2001.

[18] M. A. Suni, H. S. Dunn, P. L. Orr, et al., "Performance of plate-based cytokine flow cytometry with automated data analysis," *BMC Immunology*, vol. 4, article 9, 2003.

[19] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*, Springer, Berlin, Germany, 1996.

[20] L. A. Herzenberg, D. Parks, B. Sahaf, O. Perez, M. Roederer, and L. A. Herzenberg, "The history and future of the fluorescence activated cell sorter and flow cytometry: a view from Stanford," *Clinical Chemistry*, vol. 48, no. 10, pp. 1819–1827, 2002.

[21] W. R. Overton, "Modified histogram subtraction technique for analysis of flow cytometry data," *Cytometry*, vol. 9, no. 6, pp. 619–626, 1988.

[22] M. Roederer, "Spectral compensation for flow cytometry: visualization artifacts, limitations, and caveats," *Cytometry*, vol. 45, no. 3, pp. 194–205, 2001.

[23] S. C. De Rosa, J. M. Brenchley, and M. Roederer, "Beyond six colors: a new era in flow cytometry," *Nature Medicine*, vol. 9, no. 1, pp. 112–117, 2003.

[24] S. P. Perfetto, P. K. Chattopadhyay, and M. Roederer, "Seventeen-colour flow cytometry: unravelling the immune system," *Nature Reviews Immunology*, vol. 4, no. 8, pp. 648–655, 2004.

[25] D. R. Parks, "Data processing and analysis: data management," in *Current Protocols in Cytometry*, J. P. Robinson, Z. Darkzynkiewicz, P. N. Dean, et al., Eds., pp. 10.1.1–10.1.6, John Wiley & Sons, New York, NY, USA, 1997.

[26] H. M. Shapiro, "The evolution of cytometers," *Cytometry A*, vol. 58, no. 1, pp. 13–20, 2004.

[27] Z. Darzynkiewickz, H. Crissman, and J. W. Jacobberger, "Cytometry of the cell cycle: cycling through history," *Cytometry A*, vol. 58, no. 1, pp. 21–32, 2004.

[28] L. S. Cram, J. W. Gray, and N. P. Carter, "Cytometry and genetics," *Cytometry A*, vol. 58, no. 1, pp. 33–36, 2004.

[29] G. Tzircotis, R. F. Thorne, and C. M. Isacke, "A new spreadsheet method for the analysis of bivariate flow cytometric data," *BMC Cell Biology*, vol. 5, article 10, 2004.

[30] C. Chan, F. Feng, J. Ottinger, D. Foster, M. West, and T. B. Kepler, "Statistical mixture modeling for cell subtype identification in flow cytometry," *Cytometry A*, vol. 73, no. 8, pp. 693–701, 2008.

[31] G. Lizard, "Flow cytometry analyses and bioinformatics: interest in new softwares to optimize novel technologies and to favor the emergence of innovative concepts in cell research," *Cytometry A*, vol. 71, no. 9, pp. 646–647, 2007.

[32] H. M. Shapiro, *Practical Flow Cytometry*, Oxford University Press, New York, NY, USA, 2003.

[33] D. W. Galbraith, "Cytometry and plant sciences: a personal retrospective," *Cytometry A*, vol. 58, no. 1, pp. 37–44, 2004.

[34] R. C. Leif, J. H. Stein, and R. M. Zucker, "A short history of the initial application of anti-5-BrdU to the detection and measurement of S phase," *Cytometry A*, vol. 58, no. 1, pp. 45–52, 2004.

[35] J. W. Gratama and F. Kern, "Flow cytometric enumeration of antigen-specific T lymphocytes," *Cytometry A*, vol. 58, no. 1, pp. 79–86, 2004.

[36] H. T. Maecker, A. Rinfret, P. D'Souza, et al., "Standardization of cytokine flow cytometry assays," *BMC Immunology*, vol. 6, article 13, 2005.

[37] M. Chicurel, "Bioinformatics: bringing it all together," *Nature*, vol. 419, no. 6908, pp. 751–757, 2002.

[38] M. S. Boguski and M. W. McIntosh, "Biomedical informatics for proteomics," *Nature*, vol. 422, no. 6928, pp. 233–237, 2003.

[39] M. Keeney, D. Barnett, and J. W. Gratama, "Impact of standardization on clinical cell analysis by flow cytometry," *Journal of Biological Regulators and Homeostatic Agents*, vol. 18, no. 3-4, pp. 305–312, 2004.

[40] P. M. Ravdin, G. M. Clark, J. J. Hough, M. A. Owens, and W. L. McGuire, "Neural network analysis of DNA flow cytometry histograms," *Cytometry*, vol. 14, no. 1, pp. 74–80, 1993.

[41] W. G. Finn, K. M. Carter, R. Raich, L. M. Stoolman, and A. O. Hero, "Analysis of clinical flow cytometric immunophenotyping data by clustering on statistical manifolds: treating flow cytometry data as high-dimensional objects," *Cytometry B*, vol. 76, no. 1, pp. 1–7, 2009.

[42] N. Le Meur, A. Rossini, M. Gasparetto, C. Smith, R. R. Brinkman, and R. Gentleman, "Data quality assessment of ungated flow cytometry data in high throughput experiments," *Cytometry A*, vol. 71, no. 6, pp. 393–403, 2007.

[43] R. R. Brinkman, M. Gasparetto, S. J. J. Lee, et al., "High-content flow cytometry and temporal data analysis for defining a cellular signature of graft-versus-host disease," *Biology of Blood and Marrow Transplantation*, vol. 13, no. 6, pp. 691–700, 2007.

[44] F. Hahne, D. Arlt, M. Sauermann, et al., "Statistical methods and software for the analysis of highthroughput reverse genetic assays using flow cytometry readouts," *Genome Biology*, vol. 7, no. 8, p. R77, 2006.

[45] D. Jeffries, I. Zaidi, B. de Jong, M. J. Holland, and D. J. C. Miles, "Analysis of flow cytometry data using an automatic processing tool," *Cytometry A*, vol. 73, no. 9, pp. 857–867, 2008.

[46] K. Lo, R. R. Brinkman, and R. Gottardo, "Automated gating of flow cytometry data via robust model-based clustering," *Cytometry A*, vol. 73, no. 4, pp. 321–332, 2008.

[47] E. S. Costa, M. E. Arroyo, C. E. Pedreira, et al., "A new automated flow cytometry data analysis approach for the diagnostic screening of neoplastic B-cell disorders in peripheral blood samples with absolute lymphocytosis," *Leukemia*, vol. 20, no. 7, pp. 1221–1230, 2006.

[48] D. S. Frankel, S. L. Frankel, B. J. Binder, and R. F. Vogt, "Application of neural networks to flow cytometry data analysis and real-time cell classification," *Cytometry*, vol. 23, no. 4, pp. 290–302, 1996.

[49] M. A. Suni, H. S. Dunn, P. L. Orr, et al., "Performance of plate-based cytokine flow cytometry with automated data analysis," *BMC Immunology*, vol. 4, 2003.

[50] M. F. Wilkins, S. A. Hardy, L. Boddy, and C. W. Morris, "Comparison of five clustering algorithms to classify phytoplankton from flow cytometry data," *Cytometry*, vol. 44, no. 3, pp. 210–217, 2001.

[51] L. K. Habib and W. G. Finn, "Unsupervised immunophenotypic profiling of chronic lymphocytic leukemia," *Cytometry B*, vol. 70, no. 3, pp. 124–135, 2006.

[52] M. Roederer, W. Moore, A. Treister, R. R. Hardy, and L. A. Herzenberg, "Probability binning comparison: a metric for quantitating multivariate distribution differences," *Cytometry*, vol. 45, no. 1, pp. 47–55, 2001.

[53] Q. T. Zeng, J. P. Pratt, J. Pak, D. Ravnic, H. Huss, and S. J. Mentzer, "Feature-guided clustering of multi-dimensional flow cytometry datasets," *The Journal of Biomedical Informatics*, vol. 40, no. 3, pp. 325–331, 2007.

[54] M. F. Wilkins, L. Boddy, C. W. Morris, and R. Jonker, "A comparison of some neural and non-neural methods for identification of phytoplankton from flow cytometry data," *Computer Applications in the Biosciences*, vol. 12, no. 1, pp. 9–18, 1996.

[55] G. K. Valet and H. G. Höffkes, "Automated classification of patients with chronic lymphocytic leukemia and immunocytoma from flow cytometric three-color immunophenotypes," *Cytometry*, vol. 30, no. 6, pp. 275–288, 1997.

[56] G. Valet, M. Valet, D. Tschope, et al., "White cell and thrombocyte disorders. Standardized, self-learning flow cytometric list mode data classification with the CLASSIF1 program system," *Annals of the New York Academy of Sciences*, vol. 677, pp. 233–251, 1993.

[57] G. Valet, "Data pattern classification by the CLASSIF1 data sieving algorithm," http://www.classimed.de/classif1.html.

[58] V. C. Maino and H. T. Maecker, "Cytokine flow cytometry: a multiparametric approach for assessing cellular immune responses to viral antigens," *Clinical Immunology*, vol. 110, no. 3, pp. 222–231, 2004.

[59] M. J. Boedigheimer and J. Ferbas, "Mixture modeling approach to flow cytometry data," *Cytometry A*, vol. 73, no. 5, pp. 421–429, 2008.

[60] C. M. Kitsos, P. Bhamidipati, I. Melnikova, et al., "Combination of automated high throughput platforms, flow cytometry, and hierarchical clustering to detect cell state," *Cytometry A*, vol. 71, no. 1, pp. 16–27, 2007.

[61] Y. W. Qian, D. Mital, and S. Lee, "An online decision support system for diagnosing hematopoietic malignancies by flow cytometry immunophenotyping," in *Proceedings of the AMIA Annual Symposium*, p. 1084, 2007.

[62] J. Frelinger, T. B. Kepler, and C. Chan, "Flow: statistics, visualization and informatics for flow cytometry," *Source Code for Biology and Medicine*, vol. 3, p. 10, 2008.

[63] H. T. Maeker and V. C. Maino, "Analyzing T-cell responses to cytomegalovirus by cytokine flow cytometry," *Human Immunology*, vol. 65, no. 5, pp. 493–499, 2004.

[64] M. Dostál, Y. Giguère, T. Fait, J. Živný, and R. J. Šrám, "The distribution of major lymphocyte subsets in cord blood is associated with its pH," *Clinical Biochemistry*, vol. 34, no. 2, pp. 119–124, 2001.

[65] S. Andreatta, M. M. Wallinger, T. Posch, and R. Psenner, "Detection of subgroups from flow cytometry measurements of heterotrophic bacterioplankton by image analysis," *Cytometry*, vol. 44, no. 3, pp. 218–225, 2001.

[66] G. Grégori, A. Colosimo, and M. Denis, "Phytoplankton group dynamics in the Bay of Marseilles during a 2-year survey based on analytical flow cytometry," *Cytometry*, vol. 44, no. 3, pp. 247–256, 2001.

[67] G. Valet, H. Kahle, F. Otto, E. Brautigam, and L. Kestens, "Prediction and precise diagnosis of diseases by data pattern analysis in multiparameter flow cytometry: melanoma, juvenile asthma, and human immunodeficiency virus infection," *Methods in Cell Biology*, vol. 64, pp. 487–508, 2001.

[68] U. Petrausch, D. Haley, W. Miller, K. Floyd, W. J. Urba, and E. Walker, "Polychromatic flow cytometry: a rapid method for the reduction and analysis of complex multiparameter data," *Cytometry A*, vol. 69, no. 12, pp. 1162–1173, 2006.

[69] G. Valet, G. Roth, and W. Kellermann, "Risk assessment for intensive care patients by automated classification of flow cytometric data," in *Cytometric Cellular Analysis: Phagocyte Function*, pp. 289–306, Wiley-Liss, New York, NY, USA, 1998.

[70] L. W. Diamond, D. T. Nguyen, M. Andreeff, R. L. Maiese, and R. C. Braylan, "A knowledge-based system for the interpretation of flow cytometry data in leukemias and lymphomas," *Cytometry*, vol. 17, no. 3, pp. 266–273, 1994.

[71] W. J. Clancey, "Heuristic classification," *Artificial Intelligence*, vol. 27, no. 3, pp. 289–350, 1985.

[72] T. C. Bakker Schut, B. G. De Grooth, and J. Greve, "Cluster analysis of flow cytometric list mode data on a personal computer," *Cytometry*, vol. 14, no. 6, pp. 649–659, 1993.

[73] R. F. Murphy, "Automated identification of subpopulations in flow cytometric list mode data using cluster analysis," *Cytometry*, vol. 6, no. 4, pp. 302–309, 1985.

[74] H. Balfoort, J. Snoek, J. Smiths, L. Breedveld, J. Hofstraat, and J. Ringelberg, "Automatic identification of algae: neural network analysis of flow cytometric data," *Journal of Plankton Research*, vol. 14, pp. 575–589, 1992.

[75] M. F. Wilkins, L. Boddy, C. W. Morris, and R. R. Jonker, "Identification of phytoplankton from flow cytometry data by using radial basis function neural networks," *Applied and Environmental Microbiology*, vol. 65, no. 10, pp. 4404–4410, 1999.

[76] M. Godavarti, J. J. Rodriguez, T. A. Yopp, G. M. Lambert, and D. W. Galbraith, "Automated particle classification based on digital acquisition and analysis of flow cytometric pulse waveforms," *Cytometry*, vol. 24, no. 4, pp. 330–339, 1996.

[77] R. Kothari, H. Cualing, and T. Balachander, "Neural network analysis of flow cytometry immunophenotype data," *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 8, pp. 803–810, 1996.

[78] R. Jonker, R. Groben, G. Tarran, et al., "Automated identification and characterisation of microbial populations using flow cytometry: the AIMS project," *Scientia Marina*, vol. 64, no. 2, pp. 225–234, 2000.

[79] R. J. Beckman, G. C. Salzman, and C. C. Stewart, "Classification and regression trees for bone marrow immunophenotyping," *Cytometry*, vol. 20, no. 3, pp. 210–217, 1995.

[80] C. W. Morris, A. Autret, and L. Boddy, "Support vector machines for identifying organisms—a comparison with strongly partitioned radial basis function networks," *Ecological Modelling*, vol. 146, no. 1–3, pp. 57–67, 2001.

[81] L. Boddy, C. W. Morris, M. F. Wilkins, et al., "Identification of 72 phytoplankton species by radial basis function neural network analysis of flow cytometric data," *Marine Ecology Progress Series*, vol. 195, pp. 47–59, 2000.

[82] S. Demers, J. Kim, P. Legendre, and L. Legendre, "Analyzing multivariate flow cytometric data in aquatic sciences," *Cytometry*, vol. 13, no. 3, pp. 291–298, 1992.

[83] C. E. Pedreira, E. S. Costa, M. E. Arroyo, J. Almeida, and A. Orfao, "A multidimensional classification approach for the automated analysis of flow cytometry data," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 3, pp. 1155–1162, 2008.

[84] M. Steinbrich-Zöllner, J. R. Grün, T. Kaiser, et al., "From transcriptome to cytome: integrating cytometric profiling, multivariate cluster, and prediction analyses for a phenotypical classification of inflammatory diseases," *Cytometry A*, vol. 73, no. 4, pp. 333–340, 2008.

[85] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 10, pp. 6567–6572, 2002.

[86] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation*, vol. 3, pp. 461–483, 1991.

[87] E. Lugli, M. Pinti, M. Nasi, et al., "Subject classification obtained by cluster analysis and principal component analysis applied to flow cytometric data," *Cytometry A*, vol. 71, no. 5, pp. 334–344, 2007.

[88] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[89] Anonymous FlowQ, "Qualitiy control for flow cytometry," http://www.bioconductor.org/packages/2.2/bioc/html/flowQ.html.

[90] B. Dykstra, D. Kent, M. Bowie, et al., "Long-term propagation of distinct hematopoietic differentiation programs in vivo," *Cell Stem Cell*, vol. 1, no. 2, pp. 218–229, 2007.

[91] D. Kent, B. Dykstra, and C. Eaves, "Isolation and assessment of long-term reconstituting hematopoietic stem cells from adult mouse bone marrow," in *Current Protocols in Stem Cell Biology*, vol. 2, Unit 2A.4, 2007.

[92] J. D. Banfield and A. E. Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, vol. 49, no. 3, pp. 803–821, 1993.

[93] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.

[94] D. Peel and G. J. McLachlan, "Robust mixture modelling using the t distribution," *Statistics and Computing*, vol. 10, no. 4, pp. 339–348, 2000.

[95] S. Lallich, F. Muhlenbach, and D. A. Zighed, "Improving classification by removing or relabeling mislabeled instances," in *Proceedings of the 13th International Symposium on Foundations of Intelligent Systems (ISMIS '02)*, Lecture Notes in Computer Science, pp. 5–15, 2002.

[96] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, NY, USA, 1990.

[97] W. J. Krzanowski, T. C. Bailey, D. Partridge, J. E. Fieldsend, R. M. Everson, and V. Schetinin, "Confidence in classification: a bayesian approach," *Journal of Classification*, vol. 23, no. 2, pp. 199–220, 2006.

[98] R. Davis, "Expert Systems: where are we? And where do we go from here?" Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1982, http://hdl.handle.net/1721.1/5677.

[99] L. Boddy, M. F. Wilkins, and C. W. Morris, "Pattern recognition in flow cytometry," *Cytometry*, vol. 44, no. 3, pp. 195–209, 2001.

[100] L. Al-Haddad, C. W. Morris, and L. Boddy, "Training radial basis function neural networks: effects of training set size and imbalanced training sets," *Journal of Microbiological Methods*, vol. 43, no. 1, pp. 33–44, 2000.

[101] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[102] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems*, 1992.

[103] N. Morgan and H. Bourlard, *Generalization and Parameter Estimation in Feedforward Nets: Some Experiments*, Morgan Kaufmann, San Fransisco, Calif, USA, 1990.

[104] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, USA, 2008.

[105] L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria," *Neural Networks*, vol. 11, no. 4, pp. 761–767, 1998.

[106] W. Finnoff, F. Hergert, and H. G. Zimmermann, "Improving model selection by nonconvergent methods," *Neural Networks*, vol. 6, pp. 771–783, 1993.

[107] E. B. Baum and D. Haussler, "What size net gives valid generalization?" *Neural Computation*, vol. 1, pp. 151–160, 1989.

[108] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[109] M. Buhmann, "Radial basis functions," *Acta Numerica*, vol. 9, pp. 1–38, 2001.

[110] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, 1994.

[111] C. Biernacki and G. Govaert, "Choosing models in model-based clustering and discriminant analysis," *Journal of Statistical Computation and Simulation*, vol. 64, no. 1, pp. 49–71, 1999.

[112] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 719–725, 2000.

[113] P. J. Rousseeuw, L. Kaufman, and E. Trauwaert, "Fuzzy clustering using scatter matrices," *Computational Statistics and Data Analysis*, vol. 23, no. 1, pp. 135–151, 1996.

[114] M. Maynadié, F. Picard, B. Husson, et al., "Immunophenotypic clustering of myelodysplastic syndromes," *Blood*, vol. 100, no. 7, pp. 2349–2356, 2002.

[115] G. McLachlan and D. Peel, *Finite Mixture Models*, Wiley-Interscience, New York, NY, USA, 2004.

[116] G. Box and D. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society B*, pp. 211–252, 1964.

[117] K. L. Lange, "Robust statistical modeling using the t distribution," *Journal of the American Statistical Association*, pp. 881–896, 1989.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2009, Article ID 476106, 6 pages
doi:10.1155/2009/476106

Research Article

Fluorescence Intensity Normalisation: Correcting for Time Effects in Large-Scale Flow Cytometric Analysis

Calliope A. Dendrou, Erik Fung, Laura Esposito, John A. Todd, Linda S. Wicker, and Vincent Plagnol

Juvenile Diabetes Research Foundation/Wellcome Trust Diabetes and Inflammation Laboratory, Department of Medical Genetics, Cambridge Institute for Medical Research, University of Cambridge, Cambridge CB2 0XY, UK

Correspondence should be addressed to Vincent Plagnol, vincent.plagnol@cimr.cam.ac.uk

Received 28 April 2009; Revised 31 July 2009; Accepted 23 August 2009

Recommended by Ryan R. Brinkman

A next step to interpret the findings generated by genome-wide association studies is to associate molecular quantitative traits with disease-associated alleles. To this end, researchers are linking disease risk alleles with gene expression quantitative trait loci (eQTL). However, gene expression at the mRNA level is only an intermediate trait and flow cytometry analysis can provide more downstream and biologically valuable protein level information in multiple cell subsets simultaneously using freshly obtained samples. Because the throughput of flow cytometry is currently limited, experiments may need to span over several weeks or months to obtain a sufficient sample size to demonstrate genetic association. Therefore, normalisation methods are needed to control for technical variability and compare flow cytometry data over an extended period of time. We show how the use of normalising fluorospheres improves the repeatability of a cell surface CD25-APC mean fluorescence intensity phenotype on $CD4^+$ memory T cells. We investigate two types of normalising beads: broad spectrum and spectrum matched. Lastly, we propose two alternative normalisation procedures that are usable in the absence of normalising beads.

1. Introduction

Genome-wide association (GWA) studies have revolutionised the mapping of common genetic variants, mostly single nucleotide polymorphisms (SNPs), with susceptibility to a wide range of common, multifactorial disorders [1], in particular autoimmune diseases [2]. The next step to followup on these findings is the identification of the molecular effects of these genetic risk variants. A potential approach to achieve this goal is to associate these risk alleles, in sufficiently large cohorts, with quantitative molecular traits. This approach has been widely used in the context of gene expression mRNA analysis [3–6] but RNA is only an intermediate step and downstream protein level traits provide more valuable biological information.

Multicolour flow cytometry analysis can provide rich protein level data simultaneously on different subsets of cells; this is of particular importance for post-GWA investigations as genetic heterogeneity identified in disease-associated regions can differentially affect various cell subsets. However, the throughput of current flow cytometry approaches, including data analysis and sample collection, is limited to a small number of samples per day or week, especially when fresh blood is required. As the identification of subtle molecular effects directed by common genetic variants may require the analysis of a relatively large number of samples, flow cytometry experiments may need to span over several months. Owing to the complexity of flow cytometry technology, various technical artifacts, including variability in reagents or measuring instruments, can create time-related biases. Consequently, normalisation procedures are necessary to enable the comparison of samples analysed at different dates.

Similar issues have been identified in the context of gene expression microarray analysis. For these analyses researchers typically take advantage of the large number of independent measurements (one per gene or probe), implicitly using the rank of a gene of interest as a summary statistic. Such techniques are not available for flow cytometry data, and therefore specific approaches are required.

With the motivation of understanding the molecular effects of type 1 diabetes (T1D) risk variants located in the IL2 receptor α-chain (IL-2RA/CD25) gene region [7], we quantified cell surface expression of CD25 on CD4$^+$ T cells using flow cytometry [8]. We analysed 192 samples over a seven-month period, including 15 pairs of repeated individuals (with blood donations separated by at least three months) in order to assess measurement repeatability. We show how time-related biases affect the repeatability of a phenotype of interest, computed as a mean fluorescence intensity (MFI) in a population of CD4$^+$ memory T cells. We used the repeatability level of this genetically controlled and stable phenotype as a proxy for technical variability of the flow cytometry measurements. We show how using fluorescent calibration beads to normalise the MFIs can control for day-to-day technical variability, generated by the flow cytometer, that could not be controlled for otherwise.

2. Results

2.1. Repeatability of CD25-APC Normalised Mean Fluorescent Intensity (MFI) Phenotype. Using multicolour flow cytometry analysis, we previously identified CD25 cell surface expression on CD4$^+$ memory T cells to be associated with genetic variants in the CD25 gene region [8]. This phenotype is a MFI of anti-CD25 conjugated to APC in this cell population. To analyse this cell population the 192 samples were gated manually (using the software FlowJo, Tree Star, Inc.) to correct for interindividual and technical variability (see Figure S1 in supplementary material available online at doi: 10.1155/2009/476106 for a description of gating procedure). Constant flow cytometer settings, pooling of different antibody batches prior to the start of the study, and strict protocol adherence were used to control for technical variability. Nevertheless, when analysing the distribution of this MFI phenotype across time, we observed significant time effects. Because this phenotype is correlated to *CD25* genotype, we restricted this analysis to 149 samples with an identical T1D susceptible *CD25* genotype at the main CD25 expression associated SNP [8]. However, time-associated trends remained significant even in this subgroup ($p = 5 \times 10^{-4}$ when regressing the MFI against a quadratic model for time, coded in number of days, see Figure 1(a)). These time effects are probably due to fluctuations in the flow cytometer that cannot be measured.

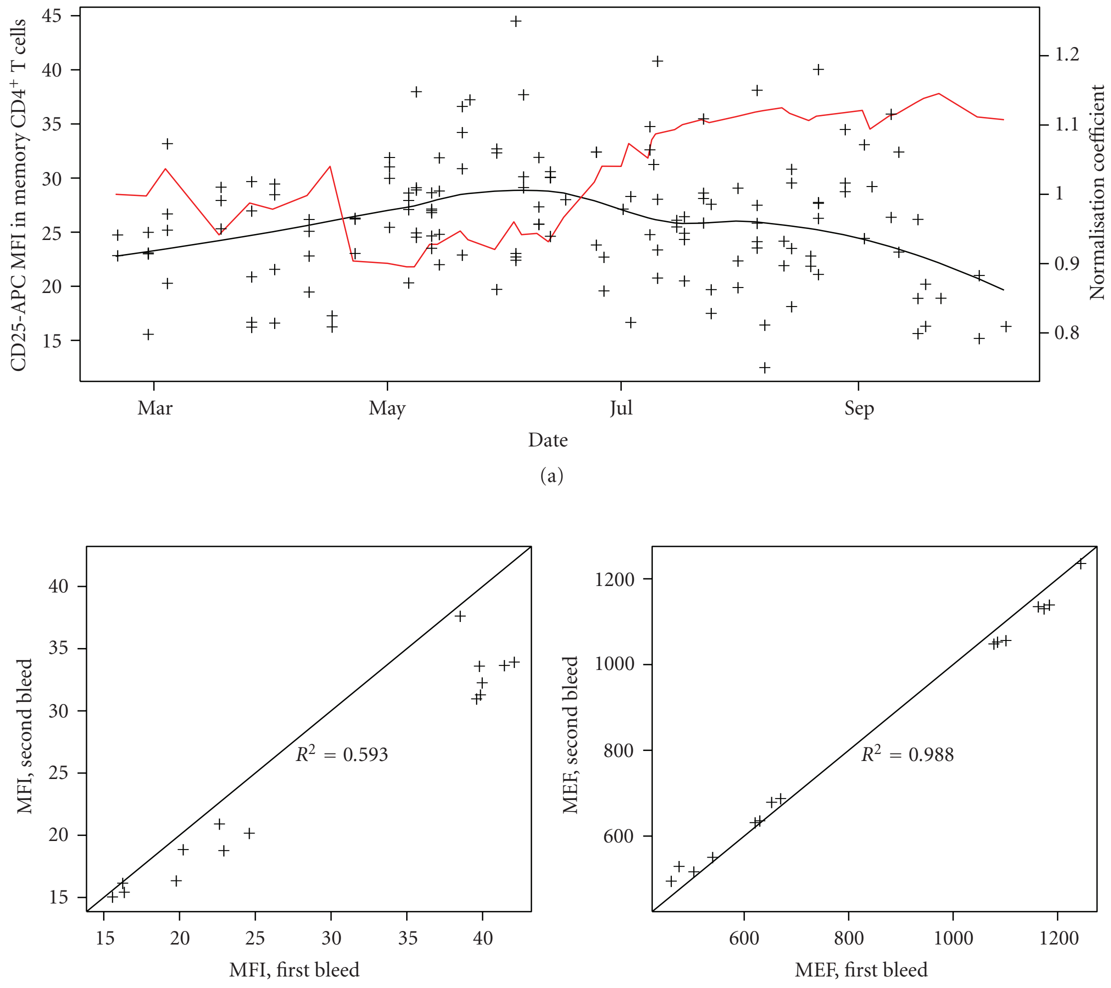
To better control for technical day-to-day variability of the flow cytometry measures, MFIs were converted to molecules of equivalent fluorochrome (MEF) using six peak calibration beads (Dakocytomation, see Methods). For each experimental day, the MFIs of the six peak calibration beads were measured using flow cytometer settings identical to the ones used for the analysed samples. Using the MFI to MEF correspondence provided by the manufacturer we fitted a linear model MEF $= \alpha \times$ MFI and used this linear transformation for MFI normalisation. The efficiency of this procedure is illustrated by the improved repeatability of the MEF in contrast with the nonnormalised MFI (Figures 1(b) and 1(c)), thus demonstrating an improved control for day-to-day technical variability.

2.2. Background Subtraction Using Isotype Control. Typical flow cytometry procedures to control for day-to-day technical variability use a fluorochrome-conjugated isotype control antibody to quantify the background, nonspecific, fluorescence intensity. Subtraction procedures are then applied to compare the background level with the observed intensity in order to estimate the fraction of positive cells, as defined by cells with a fluorescence level exceeding background [9]. In the example described here, measures obtained using background subtraction (either two-percent of background or maximum positive difference, see [9]) are less replicable ($R^2 = 0.443$) and correlations with the MEF phenotype are limited ($R^2 = 0.59$, see Figure 2).

These differences are consistent with the fact that the MFI and the fraction of CD25+positive cells provide different types of information. Therefore, these summary statistics require different normalisation approaches: one using normalisation beads, the other using an isotype control.

2.3. Broad Spectrum versus Spectrum Matched Beads. The calibration beads used in this study are broad spectrum beads, which means that the same set of beads can be used to normalise fluorochromes at different wavelengths (e.g., PE and APC using the same set of beads). Alternative normalisation tools use spectrum matched beads, that is, fluorescent beads whose light spectrum matches exactly the fluorochrome of interest, for example, APC. Such spectrum matched beads are required to standardise flow cytometry measurements across different laboratories or flow cytometers [10]. The fact that the data presented in this study were generated using a single flow cytometer (BD Biosciences LSRII) limits the complexity of MFI normalisation, thus justifying the use of broad spectrum beads.

To better understand the impact of broad versus spectrum matched normalising beads, we analysed normalising beads from another dataset generated during the same time period using the same flow cytometer. For this additional dataset broad spectrum (Dakocytomation) and APC spectrum matched (BD Biosciences) were tested. For technical reasons, and also to better understand the effect of variability in photomultiplier tube voltage (controlling the light detection sensitivity), flow cytometer settings were not kept constant through time for these additional beads data. Indeed we observed that, as expected, the normalisation coefficient is strongly negatively correlated with the APC photomultiplier tube voltage (Figure 3). Note that, in contrast with the data in Figure 3, the APC voltage remained constant for all other data (Figures 1, 2, 4, and 5), and, therefore, differences in voltage settings explain the differences in MFI trends between Figures 1(a) and 3. We found very close agreement between broad spectrum and APC spectrum matched beads (Figure 3), hence justifying the use of broad spectrum beads if the analysis involves a single flow cytometer operated under a strictly adhered-to protocol.



(a)

(b)

(c)

Figure 1: (a) Black crosses show nonnormalised MFIs in the CD4$^+$ memory T cell population as a function of time. The back line was fitted line to these MFI values using a loess procedure. The red line shows the normalisation coefficient estimated from the beads. (b) Repeatability plots ($n = 15$ pairs) for MFIs of CD25-APC cell surface expression in the CD4$^+$ memory T cell population. (c) Repeatability plots ($n = 15$ pairs) for CD25-APC MEF (normalised MFI) in the same cell population. For (b) and (c), each individual's blood donations were separated by at least 3 months.

2.4. Isotype Control Is Not Usable for MFI Normalisation. We then investigated whether MFIs obtained by measuring the isotype control fluorescence are usable for MFI normalisation, in contrast with the traditional use for background subtraction. Isotype controls are primarily used to provide information on nonspecific binding via Fc receptors present on the cells of interest. In our analyses, we attempted to block such Fc binding using mouse IgG immunoglobulin (Sigma-Aldrich Company), thereby making the isotype control primarily a measurement of the autofluorescence [11] of the cell population examined. We hypothesized that, owing to this nonspecificity, the biological donor-to-donor variability would have a more limited effect on isotype fluorescence, thus providing some information of technical variability.

In Figure 4, we show a comparison of the variability across time of the normalising beads and isotype control MFIs. We found that the variability of the isotype control MFIs greatly exceeds the variability obtained from normalising beads. A regression analysis using a quadratic model for time (coded as number of days) regressed against the average isotype MFI for each day explains only 18.4% of the variance of the isotype MFI values. The same regression for the normalising bead MFIs explains 64.8% of the measurement variance. The isotype MFIs also showed large variation across different donors analysed on the same day, suggesting that donor-to-donor differences in autofluorescence levels contribute to the isotype MFI variability. Moreover, for low MFI values in the range of the isotype control MFIs, the signal-to-noise ratio is low.

Overall, the isotype MFIs are highly variable and affected by donor-to-donor variability. In addition, the biological variability captured by the isotype control MFIs (average MFI less than 2) is not significant when analysing higher CD25 cell surface MFIs in the CD4$^+$ memory T cell

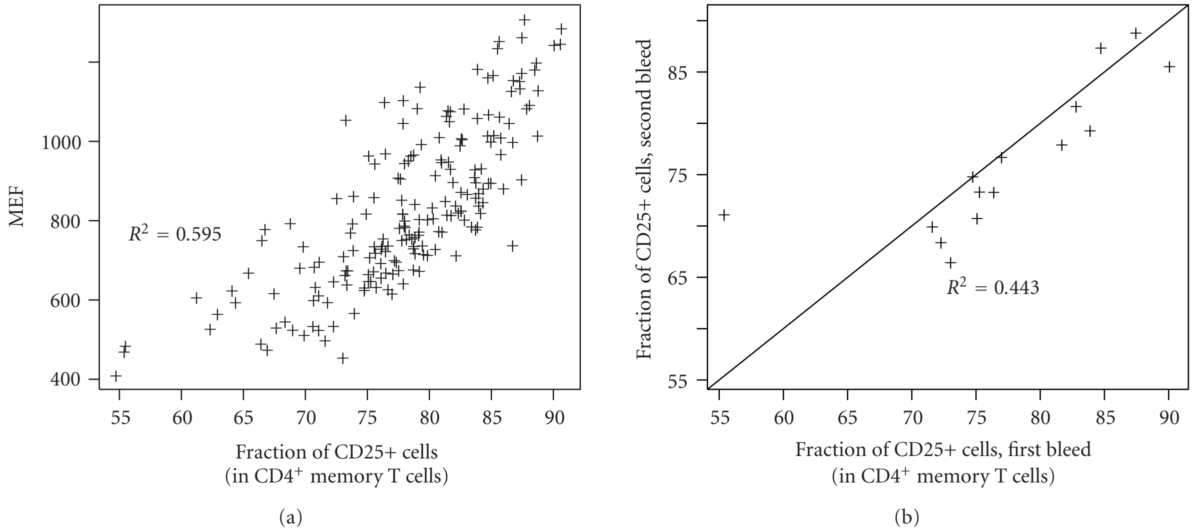


Figure 2: (a) Correlation between the fraction of CD25-positive cells in the CD4$^+$ memory T cell population and the CD25-APC MEF in this population. (b) Repeatability ($n = 15$) of the estimated fraction of CD25-positive cells in the CD4$^+$ memory T cell population obtained by background subtraction of the isotype control distribution.

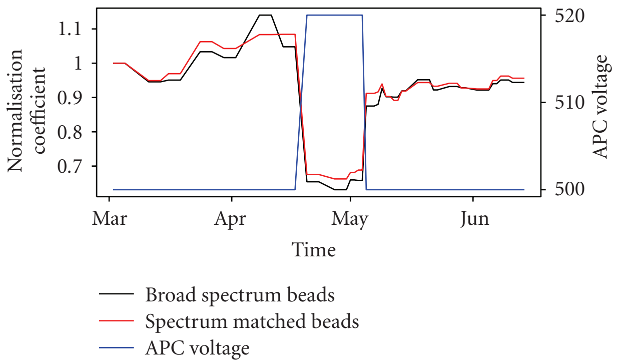


Figure 3: Variability across time of the normalisation coefficient for broad spectrum beads (black) and APC spectrum matched beads (red). The blue line shows the APC photomultiplier tube voltage setting used to measure the beads MFI.

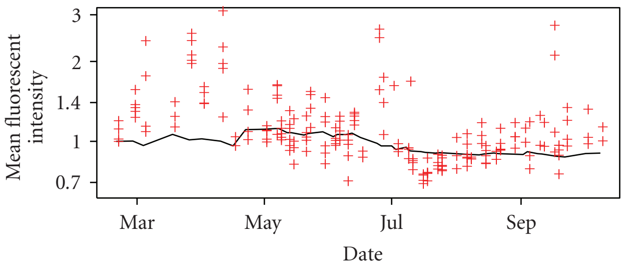
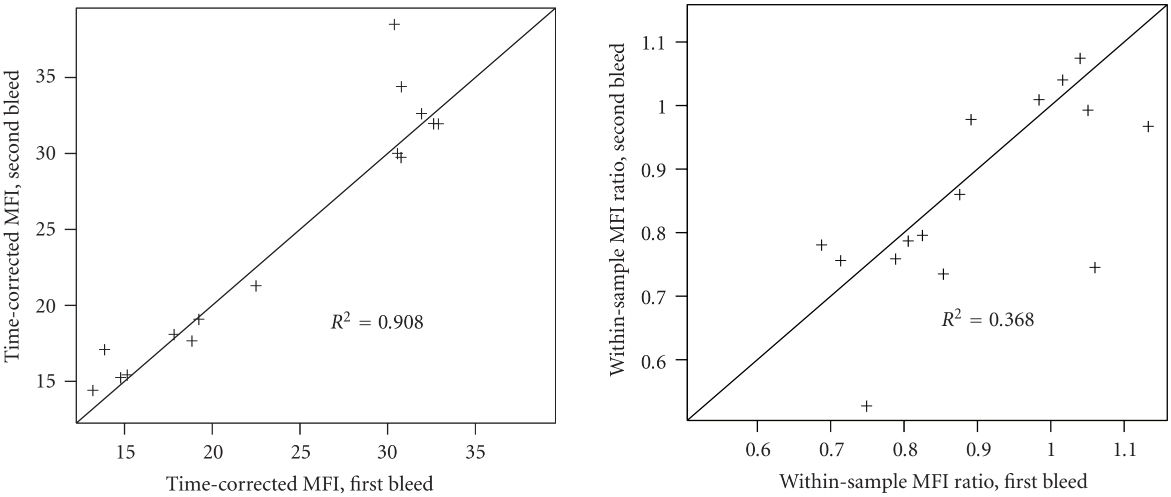


Figure 4: Variability across time of the isotype control MFIs (red crosses, one point per sample) and the normalising beads MFIs (black line, one point per experimental day). MFIs are scaled such that the value is equal to one for the first day, and a logarithmic scale is used for the y-axis.

population (average MFI: 25). Thus, the biological donor-to-donor information captured by the isotype control is not relevant for normalising the MFIs of interest. Taken together, these results indicate that the isotype control is not usable for MFI normalisation.

2.5. Across-Sample Normalisation in the Absence of Calibration Beads. We then investigated alternative procedures allowing for the control of flow cytometry day-to-day technical variability in MFI measurements in the absence of calibration beads. First, we investigated whether we could use the 192 samples analysed to estimate the trend associated with technical variability, and use this estimate to correct for time-related biases. Because of CD25 genotype-phenotype correlations [8], we only included 149 individuals with identical T1D susceptible genotypes at the main CD25 expression associated SNP. We coded time as the number of days since the first bleed and regressed a quadratic model for time against the CD25-APC MFI estimated in the total CD4$^+$ T cell population to generated predicted values p_t. The multiplicative normalising factor was estimated as $\alpha_t = p_t/p_{t=0}$. Applying this correcting factor to our main phenotype of interest (CD25-APC MFI in the CD4$^+$ memory T cell population, Figure 5(a)) significantly improved the phenotype repeatability ($R^2 = 0.91$) and helped control for time-related biases.

2.6. Within-Sample Normalisation in the Absence of Calibration Beads. We then investigated a second procedure for MFI normalisation, a flow cytometry approach analogous to quantile normalisation for gene expression microarray data. In the context of microarray data, quantile normalisation takes advantage of a large number of independent data points (one point per gene or probe) to rank a gene of interest within the overall distribution of gene intensities. This procedure corrects at least partially for variability



(a)

(b)

Figure 5: (a) Repeatability for the CD4$^+$ memory T cell population CD25-APC MFI normalised using a multiplicative correction factor estimated by a regression analysis on the set of 149 samples with identical T1D susceptible genotype. (b) Repeatability for the CD4$^+$ memory T cell population CD25-APC MFI divided by the CD25-APC MFI in the full CD4$^+$ T cell gate for the same sample/analysed tube.

across independent microarray experiments. Flow cytometry analysis, on the other hand, does not provide large numbers of independent data points. However, some partially uncorrelated MFI measures are available when analysing independent cell subsets. Therefore, we recoded our MFI phenotype of interest (computed in CD4$^+$ memory T cells) by computing, for each sample, the ratio of MFIs between CD4$^+$ memory T cells and total CD4$^+$ T cells. The advantage of this approach is the use of an internal control within the same sample, therefore providing control for technical variability. The drawback is the reliance on this additional phenotype to be biologically stable. This situation is similar to a gene expression analysis where a single gene is used for normalising the expression intensities; the underlying assumption is that the expression of this normalising gene is stable. In the example provided here the repeatability of the resulting phenotype was poor ($R^2 = 0.37$, Figure 5(b)), indicating that the repeatability of the MFI in the total CD4$^+$ T cells is lower than what we observed in the CD4$^+$ memory T cells.

3. Discussion

We have identified CD25 cell surface expression on CD4$^+$ memory T cells to be a biologically stable phenotype, quantifiable by flow cytometry analysis. We have shown that the use of broad-spectrum fluorescent normalising beads significantly reduces the day-to-day variability of flow cytometry measurements. This normalisation could not have been achieved with the sole use of an isotype control, thus motivating the development of efficient tools for flow cytometry data normalisation.

We also investigated two alternative normalisation methods, less effective than normalising beads in this example but useful in situations where fluorescent beads are absent. A potentially useful approach consists of using the MFIs obtained from a different population of cells within the same sample, thus providing an internal normalisation. However, this procedure will only be useful in a situation where a different population with repeatable MFI values exists.

In spite of these results, normalisation of fluorescence intensity data from flow cytometry remains challenging. Indeed, controlling the technical variability of such a complex experimental procedure over extended periods of time is difficult. The development of methods for higher throughput flow cytometry, enabling the analysis of dozens of samples on the same day, may address some of these issues by shortening the duration of the experiment. However, when the phenotype of interest requires the analysis of fresh blood, which is the case in this study, the limiting factor becomes the number of blood samples collected per day, which is unlikely to become much higher. We have shown recently that CD25 cell surface expression on memory cells is decreased and more variable if frozen peripheral blood mononuclear cells are analysed [8], thereby ruling out storage of frozen cells as a way to increase throughput. Therefore, for such experiments the requirement for proper normalisation of flow cytometry data across several months remains a necessity.

An elegant approach to circumvent normalisation issues is the use of a nested design comparing, on each experimental day, both categories of samples (e.g., individuals with different genotypes, or cases/controls). When using this design, only phenotypes of individuals analysed on the same day are compared with each other, thus avoiding biases associated with day-to-day technical variability. When the study is balanced (i.e., the same number of samples from each category is analysed on each day) the loss of statistical power to detect phenotype differences is minimal, while the design becomes much more robust to technical variability.

4. Methods

4.1. Antibodies and Whole Blood Immunostaining. The anti-human monoclonal antibodies used for cell surface immunostaining were APC-conjugated anti-CD25 (BD Biosciences, clones M-A251 and 2A3), Alexa-Fluor 700-conjugated anti-CD4, Alexa-Fluor 488-conjugated anti-CD127, and Pacific Blue-conjugated anti-CD45RA (BioLegend). The isotype control antibodies used were APC-conjugated mouse IgG1 (BD Biosciences) and Alexa-Fluor 488-conjugated mouse IgG1 (BioLegend). To minimize potential variation due to antibody batch differences, all antibodies were obtained prior to the start of the experiment and all vials of antibody derived from the same clone and labelled with the same fluorochrome were pooled prior to usage. To better visualize lower-level CD25 expression, we increased CD25 detection sensitivity by simultaneously using two anti-CD25 monoclonal antibodies, (labelled with the same fluorochrome (clones 2A3 and M-A251), that recognize distinct epitopes on the CD25 molecule and therefore do not cross-compete. Prior to staining, whole blood samples were blocked with mouse IgG immunoglobulin (Sigma-Aldrich Company) at a concentration of 2 μg per 100 μL blood. All samples were stained within 5 hours postvenesection. After blocking, samples were stained for 40 minutes and then lysed for 10 minutes with freshly prepared 1X BD FACS Lysing Solution (BD Biosciences). Following erythrocyte lysis, samples were incubated at 4°C and were washed with BD CellWASH (BD Biosciences). The samples were fixed with freshly prepared 1X BD CellFIX (BD Biosciences). The samples were stored at 4°C until analysis by flow cytometry.

4.2. Flow Cytometry Analysis. All immunostained samples were analyzed using a BD LSRII Flow Cytometer with BD FACSDiVa Software (BD Biosciences). Each day donor samples were evaluated, we also analysed six peak normalising fluorospheres (Blank Beads and Calibration Beads, Dakocytomation) for MFI normalisation purposes. For our second dataset, where voltage settings were allowed to vary, six peak normalising fluorospheres (Blank Beads and Calibration Beads, Dakocytomation) and BD Calibrite APC Beads (BD Biosciences) were tested on each experimental day.

4.3. Data Processing and Statistical Analysis. The flow cytometry data were analyzed using FlowJo (Tree Star, Inc.). The remaining data processing/statistical analysis was performed using the R programing language. Gates were automatically extracted from the FlowJo output using an in-house XML parsing script based on the R XML 2.3.0 library. These gates were applied to the raw FCS files using the R flowCore 1.8.3 library. Repeatability R^2 values are estimated using $[\mathrm{var}(X) - \sum_i (X_i^1 - X_i^2)^2]/\mathrm{var}(X)$ where $(X_i^1)_{i=1}^n$ and $(X_i^2)_{i=1}^n$ designate the first and second sets of replicates ($n = 15$ in this study).

Acknowledgments

The authors thank the Wellcome Trust (WT), the Juvenile Diabetes Research Foundation (JDRF), and the National Institute for Health Research (NIHR) for funding the JDRF/WT Diabetes and Inflammation laboratory. The Cambridge Institute for Medical Research (CIMR) is in receipt of a Wellcome Trust Strategic Award (079895). This work was also supported by the National Institute of Health (USA) Grant P01 AI39671 and a grant from the JDRF UK Centre for Diabetes Genes, Autoimmunity and Prevention (D-GAP; 4-2007-1003). Linda S. Wicker is a WT Principal Research Fellow. Vincent Plagnol is a JDRF advanced postdoctoral fellow. They acknowledge the use of the NIHR Biomedical Research Centre Cambridge BioResource and thank Sarah Nutland, Jennifer Sambrook, and Willem Ouwehand for contributing to this study.

References

[1] P. R. Burton, D. G. Clayton, L. R. Cardon, et al., "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls," *Nature*, vol. 447, no. 7145, pp. 661–678, 2007.

[2] A. Zhernakova, C. C. van Diemen, and C. Wijmenga, "Detecting shared pathogenesis from the shared genetics of immune-related diseases," *Nature Reviews Genetics*, vol. 10, no. 1, pp. 43–55, 2009.

[3] M. Morloy, C. M. Molony, T. M. Weber, et al., "Genetic analysis of genome-wide variation in human gene expression," *Nature*, vol. 430, no. 7001, pp. 743–747, 2004.

[4] A. L. Dixon, L. Liang, M. F. Moffatt, et al., "A genome-wide association study of global gene expression," *Nature Genetics*, vol. 39, no. 10, pp. 1202–1207, 2007.

[5] B. E. Stranger, M. S. Forrest, M. Dunning, et al., "Relative impact of nucleotide and copy number variation on gene phenotypes," *Science*, vol. 315, no. 5813, pp. 848–853, 2007.

[6] H. H. H. Göring, J. E. Curran, M. P. Johnson, et al., "Discovery of expression QTLs using large-scale transcriptional profiling in human lymphocytes," *Nature Genetics*, vol. 39, no. 10, pp. 1208–1216, 2007.

[7] C. E. Lowe, J. D. Cooper, T. Brusko, et al., "Large-scale genetic fine mapping and genotype-phenotype associations implicate polymorphism in the IL2RA region in type 1 diabetes," *Nature Genetics*, vol. 39, no. 9, pp. 1074–1082, 2007.

[8] C. A. Dendrou, V. Plagnol, E. Fung, et al., "Cell-specific protein phenotypes for the autoimmune locus *IL2RA* using a genotype-selectable human bioresource," *Nature Genetics*, vol. 41, no. 9, pp. 1011–1015, 2009.

[9] W. R. Overton, "Modified histogram subtraction technique for analysis of flow cytometry data," *Cytometry*, vol. 9, no. 6, pp. 619–626, 1988.

[10] A. Schwartz, E. Fernández Repollet, R. Vogt, and J. W. Gratama, "Standardizing flow cytometry: construction of a standardized fluorescence calibration plot using matching spectral calibrators," *Communications in Clinical Cytometry*, vol. 26, no. 1, pp. 22–31, 1996.

[11] J. E. Aubin, "Autofluorescence of viable cultured mammalian cells," *Journal of Histochemistry & Cytochemistry*, vol. 27, no. 1, pp. 36–43, 1979.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2009, Article ID 686759, 7 pages
doi:10.1155/2009/686759

Research Article

Automatic Clustering of Flow Cytometry Data with Density-Based Merging

Guenther Walther,[1] Noah Zimmerman,[2] Wayne Moore,[2] David Parks,[2] Stephen Meehan,[2] Ilana Belitskaya,[2] Jinhui Pan,[2] and Leonore Herzenberg[2]

[1] *Department of Statistics, Stanford University, Stanford, CA 94305, USA*
[2] *Department of Genetics, Stanford University, Stanford, CA 94305, USA*

Correspondence should be addressed to Guenther Walther, gwalther@stanford.edu

Received 1 May 2009; Revised 27 July 2009; Accepted 25 August 2009

Recommended by Raphael Gottardo

The ability of flow cytometry to allow fast single cell interrogation of a large number of cells has made this technology ubiquitous and indispensable in the clinical and laboratory setting. A current limit to the potential of this technology is the lack of automated tools for analyzing the resulting data. We describe methodology and software to automatically identify cell populations in flow cytometry data. Our approach advances the paradigm of manually gating sequential two-dimensional projections of the data to a procedure that automatically produces gates based on statistical theory. Our approach is nonparametric and can reproduce nonconvex subpopulations that are known to occur in flow cytometry samples, but which cannot be produced with current parametric model-based approaches. We illustrate the methodology with a sample of mouse spleen and peritoneal cavity cells.

1. Introduction

Flow cytometry allows to measure simultaneously multiple characteristics of thousands of cells. This ability has made flow cytometry a prevalent instrument in both the research and clinical settings. A major road block to tapping the full potential of this technology is the lack of data analysis methodology and software that allows for an automated and objective analysis of the data generated by this high-throughput instrument. One important part of the analysis of flow cytometry data is gating, that is, the identification of homogeneous subpopulations of cells. The current standard technique for this type of analysis is to draw 2D gates manually with a mouse on a computer screen, based on the user's interpretation of density contour lines that are provided by software tools such as FlowJo (http://www.treestar.com/) or BioConductor [1, 2]. The cells falling in this gate are extracted and the process is repeated for different 2D projections of the gated cells, thus resulting in a sequence of two-dimensional gates that describe subpopulations of the multivariate flow cytometry data.

There are several obvious problems with this kind of analysis. It is subjective as it is based on the user's interpretation and experience, it is error-prone, difficult to reproduce, time consuming, and does not scale to a high-throughput setting. For these reasons manual gating has become a major limiting aspect of flow cytometry [3–5], and there is a widely recognized need for more advanced analysis techniques [6, 7].

There have been several recent attempts to produce automatic and objective gates. Those employ the k-means algorithm [8–10] or mixture models with Gaussian components [11] or with t components and a Box-Cox transformation [12]. A drawback of all of these methods is that they produce necessarily convex subpopulations; whereas occasionally subpopulations occur that are not convex and are, for example, kidney shaped. Such subpopulations can arise, for example, when two markers are added sequentially, so that there is a developmental progression over time that moves the subpopulation first in one direction and then in another direction. The methodology introduced in this paper is grounded in nonparametric statistical theory which allows for such subpopulations.

We follow the paradigm that clusters of the data can be delineated by the contours of high-density regions [13], which is also the rationale that underlies manual gating. We implement this paradigm algorithmically by constructing a grid with associated weights that are derived by binning the data. The purpose of this grid is twofold. It allows for a fast computation of the density estimate via the Fast Fourier Transform, and it provides for an economical but flexible representation of clusters. We model each high-density region by a collection of grid points. This collection is determined algorithmically as follows. We establish links between certain neighboring grid points based on statistical decisions regarding the gradient of the density estimate. The goal is to connect neighboring grid points by a chain of links that follow the density surface "uphill." The result of this first processing stage is a number of chains that link certain grid points and which either terminate at the mode of a cluster or represent background that will not be assigned to a cluster. In a second stage the algorithm will combine some of these chains if statistical procedures indicate that they represent the same cluster. The idea of following the gradient uphill to determine clusters is motivated by manual gating and is similar to a proposal by [14], which albeit does not provide the statistical methodology required to make decisions about nonzero gradients and combining certain chains. Reference [15] gives a visual display of gradients but no algorithm for finding clusters by linking the gradients.

The end result of our algorithm is clusters that are represented by chains that link certain grid points. This representation has the advantage that it provides an efficient data structure for visualizing and extracting the cells that belong to a cluster. The chains that link grid points in a cluster represent a tree structure which can be traversed backwards to efficiently enumerate all grid points in the cluster and hence to retrieve all cells in the cluster via their nearest neighbor grid point.

2. Methods

2.1. Representing the Distribution on a Grid. Binning data on a grid allows fast processing with little loss of accuracy [16]. The current software implementation of our methodology works with successive 2D projections and we describe the methodology in this setting, although the algorithm can be generalized to work in higher dimensions from the start.

Thus we have n data points $x_i = (x_{i1}, x_{i2})$, $i = 1, \ldots, n$. To construct a grid we choose a positive integer M, typically $M = 128$ or 256, and construct the grid consisting of M^2 points as follows. Set $\Delta_j = (\max_i x_{i,j} - \min_i x_{i,j})/(M-1)$, $j = 1, 2$, and define the jth coordinate of $y_{(m_1,m_2)}$ to be $y_{m_j} = \min_i x_{i,j} + (m_j - 1)\Delta_j$, $m_j = 1, \ldots, M$. Then the grid is defined as $\{y_{(m_1,m_2)} : (m_1, m_2) \in \{1, \ldots, M\}^2\}$.

Next, each grid point $y_{\mathbf{m}}$, where $\mathbf{m} = (m_1, m_2) \in \{1, \ldots, M\}^2$, is assigned a weight $w_{\mathbf{m}}$ by linearly binning [16] the observations x_i, that is,

$$w_{\mathbf{m}} = \sum_{i=1}^{n}\prod_{j=1}^{2}\max\left(0, 1 - \frac{\left|x_{i,j} - y_{m_j}\right|}{\Delta_j}\right). \qquad (1)$$

The grid $\{y_{\mathbf{m}}, \mathbf{m} \in \{1, \ldots, M\}^2\}$ and the associated weights $\{w_{\mathbf{m}}, \mathbf{m} \in \{1, \ldots, M\}^2\}$ represent an approximation to the cell distribution. Our software implementation allows the user to choose various values of M. A larger choice of M results in a finer grid and hence a more precise approximation of the cell distribution at the expense of more computing time. However, in accordance with the results in [16], we found that a relatively small number of bins already give an excellent approximation. Within a precision of 0.01% of the total cell population we could not detect a change in the outcome of gating small subpopulations when increasing M from our default value of 256 to 512.

Our clustering algorithm described below uses only the grid and the associated weights to derive the clustering assignment. This assignment is then applied to cluster observations x_i as follows. Each observation x_i is assigned to the grid point $y_{\mathbf{m}}$ that is the closest to x_i in Euclidean norm. Then x_i is assigned to the same cluster to which its associated grid point $y_{\mathbf{m}}$ is assigned. Likewise, all observations assigned to a certain cluster can be retrieved as follows. Find all grid points $y_{\mathbf{m}}$ assigned to the given cluster, then find all observations x_i that are assigned to these grid points.

2.2. Computing the Estimate of the Cell Density. At each grid point $y_{\mathbf{m}}, \mathbf{m} \in \{1, \ldots, M\}^2$, an estimate of the density surface $\hat{f}(y_{\mathbf{m}})$ is computed as follows.

Denote by $\phi(x) = 1/\sqrt{2\pi}\exp(-x^2/2)$ the Gaussian kernel. Then the estimated density at $y_{\mathbf{m}}$ is given by (see, e.g., [16])

$$\hat{f}(y_{\mathbf{m}}) = \frac{1}{n}\sum_{l_1=-Z_1}^{Z_1}\sum_{l_2=-Z_2}^{Z_2} w_{\mathbf{m}-\mathbf{l}} \times \prod_{j=1}^{2}\frac{\phi\left(l_j\Delta_j/h_j\right)}{h_j}, \qquad (2)$$

where $\mathbf{l} = (l_1, l_2)$, $Z_j = \min(\lfloor 4h_j/\Delta_j \rfloor, M-1)$, and $h_j = \mathrm{SD}(\{x_{i,j}, i = 1, \ldots, n\})n^{-1/6}$, where SD denotes standard deviation. The above sum can be computed quickly with the Fast Fourier Transform (FFT) in a well-known way [16], but it can also be computed directly using the above formula without the FFT.

2.3. Association Pointers between the Grid Points. First, for each grid point we compute the standard error of the corresponding density estimate and then label those grid points as background whose density does not pass a certain statistical threshold. The interpretation of this criterion is that it tests whether the density is significantly different from zero; see Step 1 for details.

Next we want to construct links between grid points that follow the density gradient, that is, point "uphill." To this end, we visit each grid point in turn and compare the density estimate on this grid point with those of its neighboring grid points, of which there are at most eight. We establish a link to that neighboring grid point that has the highest value of the density estimate, provided that the difference in density estimates is statistically significant (Step 2). Testing whether the latter difference is nonzero is necessary as otherwise the variability of the density estimate may lead to links that may accidentally connect different clusters. Computationally we implement links by way of the programming language data type of a pointer.

Next we follow each chain to its end and determine whether it represents a cluster or background (Step 3). Then we determine whether two clusters need to be merged because they are connected by a path that possesses no statistically significant trough (Step 4). This is done by iteratively building a set of grid points which are neighbors to a local maximum of the density surface, are not maxima or background, and do not exhibit a statistically significant change in density when compared to the local maximum. If this set in turn possesses a neighboring grid point that is a local maximum, then we found a path (via this set) between two local maxima that does not exhibit a statistically significant trough. Consequently the last part of Step 4 merges the corresponding clusters by establishing pointers to the grid point with the highest density. We iterate Step 4 until there are no more changes in the clusters (Step 5). It can be shown that there will be only finitely many iterations. Step 6 takes care of remaining points that are assigned to the background. Thus the resulting number of clusters is determined by the data via the statistical methodology described previously.

Here is a more formal description of the various steps.

Step 1. Consider all grid points $y_{\mathbf{m}}, \mathbf{m} \in \{1, \ldots, M\}^2$, in turn. For each grid point $y_{\mathbf{m}}$ compute

$$\hat{\sigma}_{\mathbf{m}}^2 = \frac{1}{n(n-1)} \sum_{l_1=-Z_1}^{Z_1} \sum_{l_2=-Z_2}^{Z_2} w_{\mathbf{m}-\mathbf{l}} \times \prod_{j=1}^{2} \frac{\phi^2\left(l_j \Delta_j / h_j\right)}{h_j^2} - \frac{1}{n-1} \hat{f}(y_{\mathbf{m}})^2. \tag{3}$$

$\hat{\sigma}_{\mathbf{m}}$ is an estimate of the standard error of the estimated density at $y_{\mathbf{m}}$. $\hat{\sigma}_{\mathbf{m}}^2$ can be computed with the FFT as above. Define the index set $\mathcal{S} = \{\mathbf{m} \in \{1, \ldots, M\}^2 : \hat{f}(y_{\mathbf{m}}) > 4.3 * \sqrt{\hat{\sigma}_{\mathbf{m}}^2}\}$. The factor 4.3 is an adjustment for multiple testing over the grid and is obtained by calculations as in [15]. Thus $\mathcal{S}$ is the set of grid points, where the density is significantly different from zero. Grid points outside this set are marked as background. From each grid point $y_{\mathbf{m}}, \mathbf{m} \notin \mathcal{S}$, a pointer is established that points to a dummy state that represents background noise.

Step 2. For all grid points $y_{\mathbf{m}}, \mathbf{m} \in \mathcal{S}$, in turn.

Consider all the neighboring grid points $p_1, \ldots, p_{n_m}$, which are defined as the set of all grid points contained in the box $\bigcap_{j=1}^{2} \{x : y_{m_j} - \Delta_j \leq x_j \leq y_{m_j} + \Delta_j\}$. Let $p \in \{p_1, \ldots, p_{n_m}\}$ such that $\hat{f}(p) = \max_{k=1,\ldots,n_m} \hat{f}(p_k)$, splitting ties in an arbitrary manner. Then establish an association pointer from $y_{\mathbf{m}}$ to p provided the following two conditions hold:

$\hat{f}(p) > \hat{f}(y_{\mathbf{m}})$ and $(\partial/\partial e)\hat{f}(y_{\mathbf{m}}) > \lambda_{\mathbf{m}}$, where $e = (p - y_{\mathbf{m}})/\|p - y_{\mathbf{m}}\|$, $\|\cdot\|$ denotes Euclidean norm, and $(\partial/\partial e)\hat{f}(y_{\mathbf{m}})$ and $\lambda_{\mathbf{m}}$ are defined as follows:

$$\begin{gathered}
\frac{\partial}{\partial e} \hat{f}(y_{\mathbf{m}}) = \sum_{a=1}^{2} e_a \frac{\partial}{\partial y_{m_a}} \hat{f}(y_{\mathbf{m}}), \\
\frac{\partial}{\partial y_{m_a}} \hat{f}(y_{\mathbf{m}}) = \frac{1}{n} \sum_{l_1=-Z_1}^{Z_1} \sum_{l_2=-Z_2}^{Z_2} w_{\mathbf{m}-\mathbf{l}} \times \frac{-l_a \Delta_a}{h_a^2} \prod_{j=1}^{2} \frac{\phi\left(l_j \Delta_j / h_j\right)}{h_j}, \\
\lambda_{\mathbf{m}} = q\left(0.95^{1/\kappa}\right) \sqrt{\hat{\Sigma}_m^2}, \\
\kappa = \frac{\#\mathcal{S} \sum_{\mathbf{m} \in \mathcal{S}} w_{\mathbf{m}}}{n 2\pi \prod_{j=1}^{2} h_j \sum_{\mathbf{m} \in \mathcal{S}} \hat{f}(y_{\mathbf{m}})}, \\
\hat{\Sigma}_m^2 = \frac{1}{n-1} \left(\sum_{a,b=1}^{2} e_a e_b \left[A - \frac{\partial}{\partial y_{m_a}} \hat{f}(y_{\mathbf{m}}) \frac{\partial}{\partial y_{m_b}} \hat{f}(y_{\mathbf{m}}) \right] \right), \\
A = \frac{1}{n} \sum_{l_1=-Z_1}^{Z_1} \sum_{l_2=-Z_2}^{Z_2} w_{\mathbf{m}-\mathbf{l}} \times \frac{l_a l_b \Delta_a \Delta_b}{h_a^2 h_b^2} \prod_{j=1}^{2} \frac{\phi^2\left(l_j \Delta_j / h_j\right)}{h_j^2}.
\end{gathered} \tag{4}$$

Here e_1, e_2 denote the standard Euclidean basis vectors. $\hat{\Sigma}_m^2$ is an estimate of the variance of $(\partial/\partial e)\hat{f}(y_{\mathbf{m}})$ and $q(0.95^{1/\kappa})$ is the normal distribution critical value adjusted for multiple testing via κ; see, for example, [15]. A is an estimate of $(\partial/\partial y_{m_a}) f(y_{\mathbf{m}}) (\partial/\partial y_{m_b}) f(y_{\mathbf{m}})$. $q(x)$ denotes the $100 \cdot x$th percentile of the standard normal distribution. All the sums can be computed with the FFT as above. Checking that the derivative at $y_{\mathbf{m}}$ in the direction of p is significant, rather than just linking $y_{\mathbf{m}}$ to p, prevents an accidental linking of different clusters. However, this approach may result in not being able to establish links near the maximum, where the density surface is flat. This is addressed by Step 4, which merges such grid points.

Step 3. For all grid points $y_{\mathbf{m}}, \mathbf{m} \in \mathcal{S}$, in turn.

If a pointer originates at $y_{\mathbf{m}}$, then it will point to a different grid point, which itself may have a pointer originating from it. This succession of pointers is followed until one arrives at a grid point $y_{\mathbf{z}}$ that either

(a) $y_{\mathbf{z}}$ does not have any pointer originating from it, or

(b) $y_{\mathbf{z}}$ has a pointer originating from it which points to a dummy state that represents a cluster or background noise.

In case (a) all the pointers visited in succession will be removed and new pointers originating from each grid point visited in succession will be established to the dummy state that represents the background noise, provided the following condition holds:

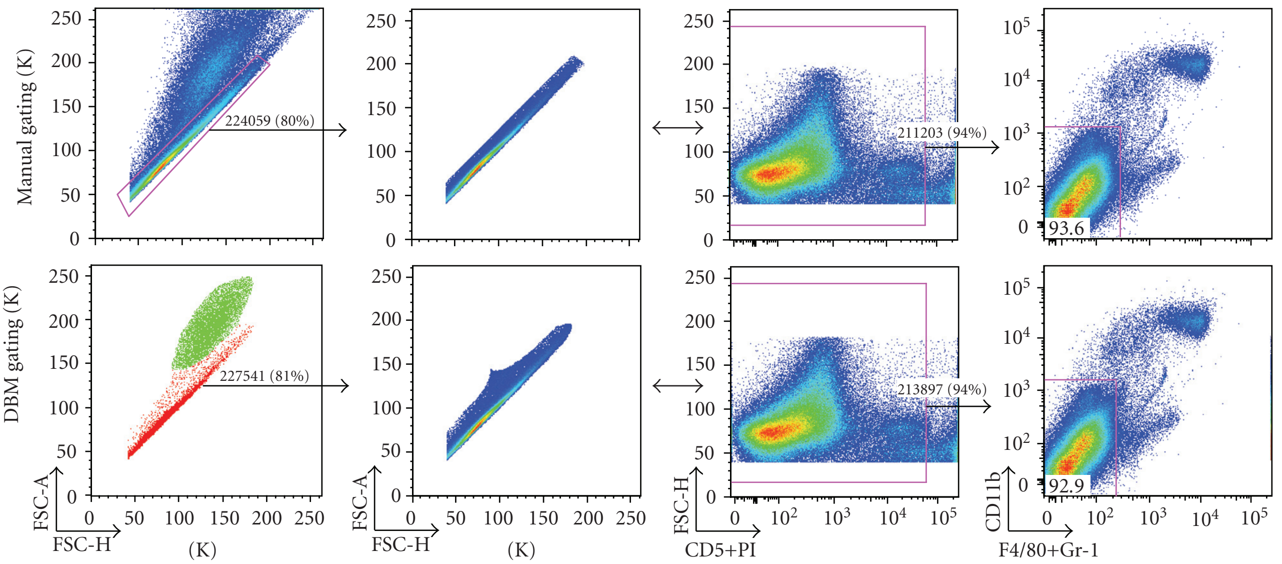


Figure 1: Comparison of manual and DBM gating in the scatter dimensions—singlet gates are shown as determined by the researcher (top) and DBM (bottom, colored plot frames) for neonatal mouse spleen cells. The subset is further gated using the researcher's live/dead gate and displayed in context of the next gating decision by the researcher. Note that the results of the DBM clustering are displayed with same software that was used for the manual gating (FlowJo). This was done to facilitate the comparison and because a suitable display system for publication has not yet been developed. Thus in the bottom left plot, color is used to code the clusters found by DBM.

$$\hat{f}(y_{\mathbf{z}}) < q\left(0.95^{1/\kappa}\right)\sqrt{\hat{\sigma}_{\mathbf{z}}^2}. \qquad (5)$$

Otherwise, provided there is a pointer into $y_{\mathbf{z}}$, then a new pointer will be established that originates from $y_{\mathbf{z}}$ and points to a newly established dummy state that represents a new cluster.

In case (b) no pointers are removed or established.

Step 4. Let $\{y_{\mathbf{m}(1)},\ldots y_{\mathbf{m}(k)}\}$ be the set of all grid points which have a pointer originating from them to a dummy state representing a cluster, enumerated such that $\hat{f}(y_{\mathbf{m}(1)}) \geq \cdots \geq \hat{f}(y_{\mathbf{m}(k)})$.

For $i = 1,\ldots,k$ do the following.

Set $\mathcal{A} = \{\mathbf{m}(i)\}$. Iterate the following loop until no more indices are added to $\mathcal{A}$:

(Begin loop)

For each index $\mathbf{a} \in \mathcal{A}$ in turn, add all the indices $\mathbf{p}$ to $\mathcal{A}$ that satisfy

(i) $y_{\mathbf{p}}$ is a neighbor of $y_{\mathbf{a}}$ as defined in Step 2,

(ii) no pointer originates from $y_{\mathbf{p}}$,

(iii) $\hat{f}(y_{\mathbf{p}}) + \hat{\sigma}_{\mathbf{p}} \geq \hat{f}(y_{\mathbf{m(i)}}) - \hat{\sigma}_{\mathbf{m(i)}}$

(End loop)

Denote by $\mathcal{B}$ the set of indices of grid points which satisfy the following two conditions. The grid point possesses a pointer originating to a dummy state representing a cluster, and the grid point has some $y_{\mathbf{p}}, \mathbf{p} \in \mathcal{A}$ as neighbor. If $\mathcal{B}$ is not empty, then do the following.

Define $\mathbf{q}$ by $\hat{f}(y_{\mathbf{q}}) = \max_{\mathbf{r}\in\mathcal{B}}\hat{f}(y_{\mathbf{r}})$, breaking ties arbitrarily.

Establish a pointer from each $y_{\mathbf{p}}, \mathbf{p} \in \mathcal{A} \setminus \{\mathbf{m}(\mathbf{i})\}$, to $y_{\mathbf{q}}$.

For each $\mathbf{r} \in \mathcal{B}, \mathbf{r} \neq \mathbf{q}$, remove the pointer from $y_{\mathbf{r}}$ to the dummy state representing a cluster and establish a new pointer from $y_{\mathbf{r}}$ to $y_{\mathbf{q}}$.

(End loop over i)

Step 5. Repeat Step 4 until there are no more additions or deletions of pointers to dummy states representing clusters.

Step 6. From each grid point that does not have a pointer originating from it, establish a pointer pointing to the dummy state that represents the background noise.

After Step 6 every grid point has a pointer originating from it. Following the succession of pointers leads to a dummy state which represents either background noise or a cluster. All grid points which are thus linked to the same dummy state pertain to the same cluster (or background noise). Cluster memberships of observations x_i derive from the cluster memberships of the grid points as explained in Section 2.1.

3. Results

We implemented the density-based merging (DBM) algorithm in a Java application with a graphical user interface that allows cluster visualization and sequential selection of clusters to support progressive gating. To enable comparison of DBM gating with data gated manually with a commercial analysis package (FlowJo, http://www.treestar.com/), we record cluster assignments for each event in association with the original data. These values are used as synthetic gating

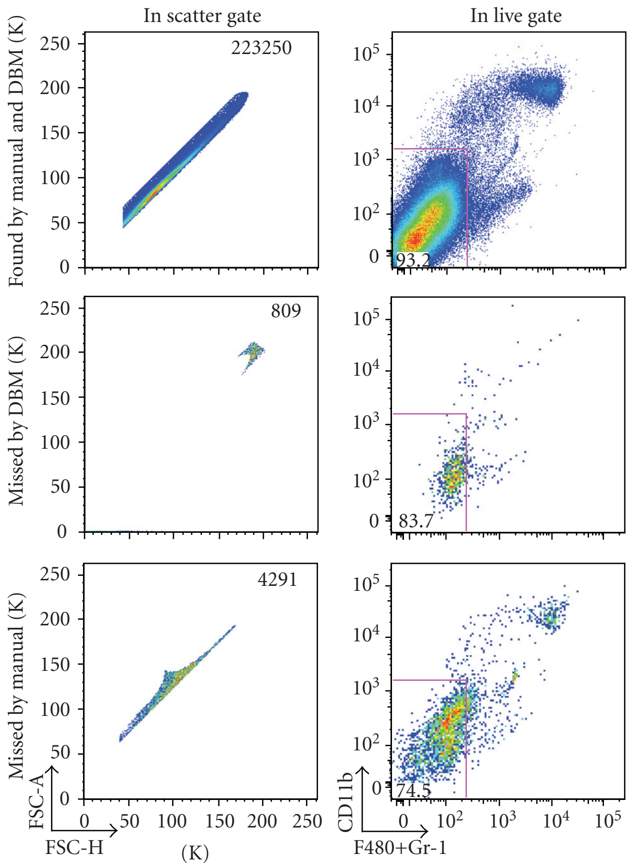


Figure 2: Differences in manual versus DBM gating in scatter dimensions—cells included by both gates (top), cells included in the manual gate and excluded by the DBM gate (middle), and cells included in the DBM gate and excluded by the manual gate (bottom) are displayed (column 1). Cells are live/dead gated as described in the text, and shown in the context of the next manual gating decision (column 2).

parameters in the commercial package, where we can directly compare results.

Mouse spleen and peritoneal cavity cells harvested in serum-containing medium were incubated on ice for 15 minutes with a 10-color staining combination. Data were collected on an LSR II (Becton Dickinson).

In the data shown in Figures 1–3, we replicate manual gating decisions from a dataset previously analyzed by a senior researcher using FlowJo. The researcher has sequentially selected gates that progressively restrict the inclusion of cells to ultimately encompass a known functionally distinct subset. For each of these sequential manual gating decisions, we select the corresponding cluster(s) defined by the DBM algorithm. In our analysis, we thus reproduce the existing workflow of the researcher, with the notable exception that we use gating boundaries that are defined algorithmically.

Figure 1 (first column) compares the initial gating in the forward-scatter area/height dimensions performed manually (top) or with DBM (bottom). The research intention here is to separate single cells from doublets and other debris. Drawing the manual gate requires a great deal of experience for a researcher to draw, owing to the lack of visual differentiation between the overlapping populations. DBM identifies two clusters that agree surprisingly well with the manual gate: the red cluster contains 81% of the total events; the corresponding expert gate contains 80% of the total events; the overlap between the two gates is 98%.

Two views of the events encompassed by the clusters are shown in columns 2 and 3 of Figure 1. Column 4 shows further gating of the samples with the same manual gate applied to the manually gated (top) and DBM gated (bottom) data shown in columns 2 and 3. The similarity of the yield from the manually gated and DBM gated sample underscores the strong overlap between the two samples.

In each case, a small percentage of the events captured by one of the gating methods are excluded from the other (Figure 2). Importantly we find that the DBM gate tends to better capture the desired events then does the researcher's gate. We define desirable events as those included in the subsequent gates that the expert set. The gate set by the expert included fewer cells in the desired subset than the DBM gate, resulting in a loss of desired cells (3474 cells). The expert gate also included fewer cells outside the desired subset. However, the additional "nondesired" cells included in the DBM gate are not relevant since the expert has gated these out of the subsequent analysis. Thus, in this situation, the DBM gate is more successful than the expert gate.

In Figures 1 and 2, we analyzed the results of a single DBM gate generated to match the first gate that the expert applied in the gating series. Figure 3, which is based on a different dataset, compares results from three sequential gates applied by the researcher with the comparable sequential DBM gates. The researcher has chosen three sequential gates (Figure 3, top): the first gate excludes doublets and debris; the second gate excludes dead cells (bright PI); the third, which yields a subset that is enriched for B cells (the target of interest to the expert), excludes monocytes and macrophages (CD11bbr, F4/80+GR-1br).

Applying the corresponding sequence of DBM clusters results in a distribution (Figure 3, bottom) that is almost indistinguishable from the distribution obtained with the expert's gates. The principal differences is a small increase in the number of cells in the B cell subset desired by the expert, and the inclusion of a small percentage of cells that lie near, but not within, the B cell subset.

We view these results as extremely promising. We are pleased that the DBM algorithm performed at least as well than the expert in terms of identifying the subset of interest in this study. We plan to perform future studies with more diverse datasets to provide a more detailed investigation of the performance of the DBM algorithm.

4. Discussion

Flow cytometry allows to separate cells into subsets for further analysis. The potential of this technology is currently limited by a lack of automatic and objective data analysis and gating techniques. We introduced methodology and

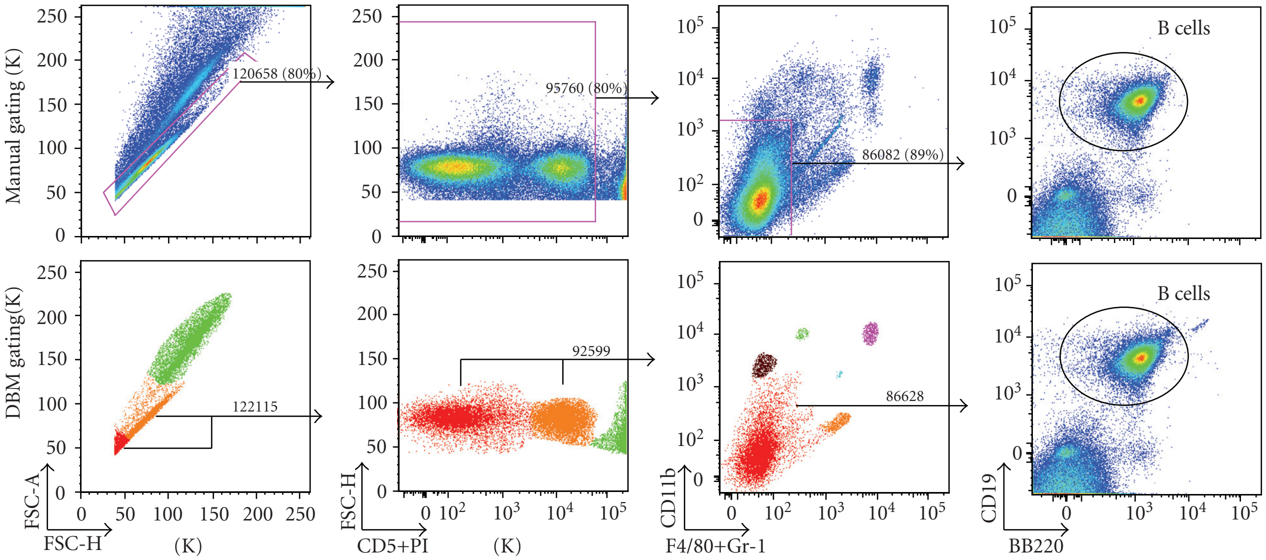


Figure 3: Comparison of manual and DBM gating for 3-step gating sequence—adult mouse spleen cells are analyzed using the researcher's manual gates (top plots) and the corresponding clusters identified by DBM (bottom plots with colored plot frames). Color is used to code the clusters found by DBM in the first three plots on bottom. Each of the manual/DBM gate pairs has < 4% difference in total number of cells. In this study, the researcher is interested in B cells (column 4).

demonstrated a software implementation that allows automatic 2D gating that is based on statistical theory and hence objective, reproducible, and fast. Typically, the automatic gating takes only a fraction of a second. An important feature of this methodology is that it is nonparametric and allows for nonconvex gates, which current parametric methodology with mixture models does not provide. Likewise, the nonparametric statistical theory provides the information necessary to decide on the number of populations in the sample, which is known to be a difficult problem in the context of parametric mixture models with no satisfactory solution currently available.

We implemented our methodology in a sequential 2D setting to automate the traditional manual gating. While the methodology can in principle be implemented in a higher-dimensional setting, there are also advantages to stick with the traditional sequential procedure. First, many users are familiar with the sequential gating procedure and may be hesitant to work with the high-dimensional output of a "black box," which may be difficult to interpret. Second, it is common practice to first project the data on the forward light scatter (FSC) and sideward light scatter (SSC) to distinguish basic cell types (e.g., monocytes and lymphocytes) and to remove dead cells and cell debris. Also, the user may have prior knowledge that leads her to consider certain 2D projections or gating paths. These aspects are readily incorporated in our implementation. Third, sequential 2D gating allows for an informative and straightforward visualization of the gating and the results.

We implemented our methodology in software called ClusterGenie which we plan to be open source but distributed commercially. We demonstrated it on a sample of mouse spleen and peritoneal cavity cells. Our results compared favorably with expert gating of the data in FlowJo. We plan a rigorous quantitative assessment of our methodology in the near future.

Acknowledgments

The authors acknowledge Eliver Ghosn and Yang Yang for data and interpretation of results. This work was supported by NIH Grant R01AI077395 and National Library of Medicine training Grant T15LM007033.

References

[1] R. Gentleman, V. J. Carey, and D. M. Bates, "Bioconductor: open software development for computational biology and bioinformatics," *Genome Biology*, vol. 5, p. R80, 2004.

[2] B. Ellis, P. Haaland, F. Hahne, N. Le Meur, and N. Gopalakrishnan, "flowCore: basic structures for flow cytometry data," R package version 1.10.0.

[3] S. C. De Rosa, J. M. Brenchley, and M. Roederer, "Beyond six colors: a new era in flow cytometry," *Nature Medicine*, vol. 9, no. 1, pp. 112–117, 2003.

[4] D. Redelman, "CytometryML," *Cytometry A*, vol. 62, no. 1, pp. 70–73, 2004.

[5] J. Spidlen, R. C. Gentleman, P. D. Haaland, et al., "Data standards for flow cytometry," *OMICS*, vol. 10, no. 2, pp. 209–214, 2006.

[6] S. C. De Rosa, L. A. Herzenberg, L. A. Herzenberg, and M. Roederer, "11-color, 13-parameter flow cytometry: identification of human naive T-cells by phenotype, function, and T-cell receptor diversity," *Nature Medicine*, vol. 7, no. 2, pp. 245–248, 2001.

[7] G. Lizard, "Flow cytometry analyses and bioinformatics: interest in new softwares to optimize novel technologies and

to favor the emergence of innovative concepts in cell research," *Cytometry A*, vol. 71, no. 9, pp. 646–647, 2007.

[8] R. F. Murphy, "Automated identification of subpopulations in flow cytometric list mode data using cluster analysis," *Cytometry*, vol. 6, no. 4, pp. 302–309, 1985.

[9] T. C. Bakker Schut, B. G. D Grooth, and J. Greve, "Cluster analysis of flow cytometric list mode data on a personal computer," *Cytometry*, vol. 14, no. 6, pp. 649–659, 1993.

[10] S. Demers, J. Kim, P. Legendre, and L. Legendre, "Analyzing multivariate flow cytometric data in aquatic sciences," *Cytometry*, vol. 13, no. 3, pp. 291–298, 1992.

[11] M. J. Boedigheimer and J. Ferbas, "Mixture modeling approach to flow cytometry data," *Cytometry A*, vol. 73, no. 5, pp. 421–429, 2008.

[12] K. Lo, R. R. Brinkman, and R. Gottardo, "Automated gating of flow cytometry data via robust model-based clustering," *Cytometry A*, vol. 73, no. 4, pp. 321–332, 2008.

[13] J. A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, New York, NY, USA, 1975.

[14] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with application in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[15] F. Godtliebsen, J. S. Marron, and P. Chaudhuri, "Significance in scale space for bivariate density estimation," *Journal of Computational and Graphical Statistics*, vol. 11, no. 1, pp. 1–21, 2002.

[16] M. P. Wand, "Fast computation of multivariate kernel estimators," *Journal of Computational and Graphical Statistics*, vol. 3, pp. 433–445, 1994.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2009, Article ID 247646, 12 pages
doi:10.1155/2009/247646

Research Article

Merging Mixture Components for Cell Population Identification in Flow Cytometry

Greg Finak,[1] Ali Bashashati,[2] Ryan Brinkman,[2] and Raphaël Gottardo[1, 3]

[1] *Computational Biology Unit, Clinical Research Institute of Montreal, 110 Pine Avenue West, Montreal, QC, Canada H2W1R7*
[2] *Terry Fox Laboratory, BC Cancer Research Center, Vancouver, BC, Canada V5Z 1L3*
[3] *Département de Biochimie, Université de Montreal, Montreal, QC, Canada*

Correspondence should be addressed to Raphaël Gottardo, raphael.gottardo@ircm.qc.ca

Received 7 April 2009; Revised 6 July 2009; Accepted 22 August 2009

Recommended by George Luta

We present a framework for the identification of cell subpopulations in flow cytometry data based on merging mixture components using the flowClust methodology. We show that the cluster merging algorithm under our framework improves model fit and provides a better estimate of the number of distinct cell subpopulations than either Gaussian mixture models or flowClust, especially for complicated flow cytometry data distributions. Our framework allows the automated selection of the number of distinct cell subpopulations and we are able to identify cases where the algorithm fails, thus making it suitable for application in a high throughput FCM analysis pipeline. Furthermore, we demonstrate a method for summarizing complex merged cell subpopulations in a simple manner that integrates with the existing flowClust framework and enables downstream data analysis. We demonstrate the performance of our framework on simulated and real FCM data. The software is available in the flowMerge package through the Bioconductor project.

1. Introduction

Flow cytometry (FCM) can be applied in a high-throughput fashion to process thousands of samples per day. However, data analysis can be a significant challenge because each data set is a multiparametric description of millions of individual cells. Consequently, despite widespread use, FCM has not reached its full potential due to the lack of an automated analysis platform to assist high-throughput data generation.

A critical bottleneck in data analysis is gating, the identification of groups of similar cells for further study. The process involves identification of regions in multivariate space containing homogeneous cell populations of interest. Generally, gating has been performed manually by expert users, but manual gating is subject to user variability, which can potentially impact results [1–3].

A number of methods have been developed to automate the gating process [4–7]. These include model-based methods such as multivariate mixture models that describe the joint density of the flow cytometry data as a mixture of simpler distributions [5, 6]. The simplest of these methods utilizes a mixture of multivariate gaussian distributions [5]. However it is not sufficiently flexible to model the outliers or asymmetrical cell populations frequently found in flow cytometry data [6].

A more recent approach compensates for these effects by applying a data transformation during the model fitting process [6, 8]. This transformation makes data more symmetric, while the use of a multivariate *t* distribution allows the model to handle outliers [6, 8, 9].

These model-based gating methods effectively amount to clustering of the data and generally employ likelihood-based measures such as the Bayesian information criterion (BIC) or Akaike information criterion (AIC) to select an appropriate model (number of clusters) from a range of possibilities [10]. While these measures are effective for choosing a model that provides a good fit to the underlying data distribution, they are problematic for clustering flow cytometry data, where the goal is to determine the correct number of distinct cell populations. BIC favors models with more mixture components in order to provide a better fit to the data distribution [11]. However, this comes at the

Table 1: Distributional assumptions, data transformation, and model selection criteria for the five clustering models compared in this study.

Distribution	Transformation	Model selection criteria	Model name
Multivariate-t	Box-Cox	BIC	flowClust$_{\mathrm{BIC}}$
	Box-Cox	ICL	flowClust$_{\mathrm{ICL}}$
	Box-Cox	Fixed K	flowClust$_{\mathrm{K}}$
	Box-Cox	BIC, entropy	flowMerge
	Box-Cox	BIC, entropy, fixed K	flowMerge$_{\mathrm{K}}$
Gaussian	None	BIC	GMM$_{\mathrm{BIC}}$
	None	ICL	GMM$_{\mathrm{ICL}}$
	None	fixed K	GMM$_{\mathrm{K}}$

cost of overestimating the number of well-separated clusters, particularly when clusters are asymmetric and/or nonconvex.

An alternative measure recently proposed for model selection is the Integrated Complete Likelihood (ICL)[11]. The ICL is an entropy-penalized BIC criterion, wherein the BIC is penalized by an entropy term, which increases as a function of the overlap between model components. Consequently, ICL favors models with fewer components and provides a better estimate of the number of well-separated populations; however this generally comes at the cost of a poor fit to the empirical data distribution, especially if clusters are asymmetric, nonconvex, or otherwise not readily fit by a simple parametric distribution [12].

In flow cytometry, where the shapes of cell populations can be asymmetric and nonconvex, neither of the above model fitting criteria are well suited to the clustering problem. An ideal model would allow multiple mixture components to represent an individual cluster or cell population, thus providing a good fit to the data and a good estimate of the number of distinct clusters. Such an algorithm has recently been proposed for Gaussian mixture models (GMMs) [12]. The algorithm starts with the best model selected by the BIC criterion and iteratively merges pairs of overlapping clusters in order to minimize the entropy of the model [12]. Because it is based on the best fitting BIC model, this approach retains the good distributional fitting properties of the best BIC model, while simultaneously allowing multiple mixture components to represent a single cluster. Like the ICL measure, it also provides a reasonable estimate of the number of well separated clusters in the data [12]. Merging clusters to improve fitting of nonconvex cell population has also recently been suggested by Pyne et al. [13].

Here we extend the work of Baudry et al. to subpopulation identification in flow cytometry data [12]. We combine the cluster merging algorithm with the more flexible model classes provided by a multivariate t-mixture with Box-Cox transformed data and develop a method for summarizing merged clusters that is compatible with the flowClust framework [6]. Additionally, we automate the choice of the number of clusters in the cluster merging algorithm, making it suitable for application in a high throughput FCM analysis pipeline. We propose a method for the identification of borderline cases where the merging algorithm fails, which can be flagged for manual analysis. In Table 1 we list the distributional assumptions, model selection criteria, and the abbreviations used to refer to the five models compared throughout this paper.

Employing the cluster merging algorithm under the flowClust framework provides a better fit and a better estimate of the number of distinct cell populations for complicated flow cytometry data distributions, than either the flowClust$_{\mathrm{BIC}}$, flowClust$_{\mathrm{ICL}}$, GMM$_{\mathrm{BIC}}$, or GMM$_{\mathrm{ICL}}$ models. The cluster merging algorithm provides a simpler visual representation of the data that is more amenable to interpretation. We demonstrate the performance of our algorithm on simulated and real FCM data. The software is available through the Bioconductor project.

2. Materials and Methods

2.1. The flowClust Framework. We embed the cluster merging algorithm within the flowClust framework available in BioConductor [6, 14]. The flowClust package is used to fit mixture models of multivariate t distributions to flow cytometry data. Additionally, the model allows the data to be Box-Cox transformed during model fitting, with the goal of making the data distribution more symmetric and bringing it closer to "normality". The model allows a number of parameters to be estimated from the data, including the degrees of freedom ν of the multivariate t distributions being fitted and the Box-Cox transformation parameters λ (Table 1). While flowClust does allow independent degrees of freedom and independent Box-Cox transformation parameters to be estimated for each mixture component, we chose to use a common degrees of freedom and common Box-Cox transformation parameter, estimated from the data, across all mixture components in a model. This was done in order to have closed form estimates of summary statistics for the merged components. Note also that this additional flexibility is not necessary in our framework as subpopulations can be represented as mixtures of multiple components. In the rest of this paper, we refer to this as the *flowClust* model.

2.2. The Cluster Merging Algorithm. We have implemented the cluster merging algorithm described in [12], with several modifications allowing its use with flow cytometry data within the flowClust framework. Briefly, we begin with the optimal flowClust$_{\mathrm{BIC}}$ solution of K clusters. At the first iteration of the algorithm, two clusters are chosen for merging in order to minimize the entropy of the data under the new cluster assignments, as described in [12]. The entropy of clustering for a K cluster mixture model is defined as

$$\mathrm{ENT}(K) = -2\sum_{k=1}^{K}\sum_{i=1}^{N} p_{ik}\log_2\left(p_{ik}\right), \tag{1}$$

where p_{ik} is the probability of data point i belonging to cluster k. Thus for two overlapping clusters k, $k+1$, the probability of a data point i in the overlapping region belonging to either cluster is nonzero, and the entropy is high. If the clusters overlap very little or not at all, then the entropy is zero or near zero. Consequently, by iteratively merging overlapping components, the entropy of clustering is reduced. At each successive iteration, two more clusters are merged until, at the Kth iteration, the data is defined by a single cluster.

Baudry et al. suggest two data-driven approaches for choosing the optimal k-cluster solution [12]. The first involves identifying an "elbow" in a plot of the entropy of clustering versus the number of clusters in a solution. The second involves identifying peaks in a plot of the number of clusters versus the change in entropy obtained by merging two clusters in the $k+1$ cluster solution into a single cluster to form the k cluster solution (see [12] for details). Here, we propose an automated approach for choosing the optimal k-cluster solution based on changepoint analysis of the entropy versus number of clusters plot, making the cluster merging algorithm suitable for inclusion in an automated workflow for flow cytometry data analysis [8].

2.3. Parameter Representation of Merged Mixture Components. It is important to be able to have a parametric representation of merged clusters in order to summarize characteristics of the population. To this end, we model a merged cluster as a multivariate t distribution with degrees of freedom, ν, equal to the degrees of freedom of its component clusters. We let $\mathbf{X}_i$ and $\mathbf{X}_j$ be random variables that represent the p dimensional measurements of cells in clusters i and j. We let $\mathbf{X}_*$ be the random variable that represents the p dimensional measurements of cells in the cluster created by merging clusters i and j (i.e., any two clusters). We let f_*, f_i, and f_j be the distributions of $\mathbf{X}_*$, $\mathbf{X}_i$, and $\mathbf{X}_j$, respectively, and n_i, n_j the number of events in clusters i and j, respectively. Thus f_* can be written as a mixture of f_i and f_j (see [12] for details) as follows:

$$p_* f_* = p_i f_i + p_j f_j. \tag{2}$$

Thus, by definition, the proportion of cells p_* in the merged cluster is equal to the sum of the proportions of the components p_i and p_j, given by

$$p_* = p_i + p_j. \tag{3}$$

Because we model the merged cluster as a single multivariate t distribution we can summarize merged components with individual sets of parameters describing their locations and scales. To estimate the mean and covariance matrix of the merged component, we match the first two moments of the distributions in (2) (see [15]), giving

$$\begin{aligned}
\boldsymbol{\mu}_* &= \frac{\left(p_i\boldsymbol{\mu}_i + p_j\boldsymbol{\mu}_j\right)}{p_*},\\
\boldsymbol{\Sigma}_* &= \frac{(\nu_* - 2)p_i\left[(\nu_i/(\nu_i-2))\boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i\boldsymbol{\mu}_i'\right]}{p_*\nu_*}\\
&\quad + \frac{(\nu_* - 2)p_j\left[\left(\nu_j/\left(\nu_j-2\right)\right)\boldsymbol{\Sigma}_j + \boldsymbol{\mu}_j\boldsymbol{\mu}_j'\right]}{p_*\nu_*}\\
&\quad - \frac{(\nu_* - 2)p_*\boldsymbol{\mu}_*\boldsymbol{\mu}_*'}{p_*\nu_*}.
\end{aligned} \tag{4}$$

The expressions in (4) are the mean vector and covariance matrix of the merged distribution, which is approximated by a multivariate t model with $\nu_* = \nu_i$ and $\nu_i = \nu_j$ degrees of freedom. As previously mentioned, a common Box-Cox transformation parameter allows us to estimate the parameters of the merged clusters on the transformed scale.

2.4. Estimating the Number of Clusters/Cell Subpopulations. Our stopping criteria for merging are based on analysis of the number of clusters in a solution versus the clustering entropy of that solution. Intuitively, when mixture components overlap significantly, the entropy of clustering will be a large value. As components are combined in subsequent iterations of the merging algorithm, the entropy will decrease. When only well separated components are left in the clustering solution, further merging will have little impact on the total entropy of clustering. This is reflected in a change of slope in the plot of the clustering entropy versus the number of components at the point, where the remaining clusters are well separated. We refer to this as the optimal flowMerge solution.

We formalize this idea by fitting piecewise linear regression to the entropy versus the number of clusters in the series of flowMerge model and allow the regression to have either one or two segments (i.e., one or no changepoint). Furthermore, we force the location of the changepoint to be an integer, thus reflecting the discrete nature of the clustering. Formally, if we have K models with an increasing number $(1\cdots K)$ clusters, we fit a series of two-segment piecewise linear regressions to the entropy versus the number of clusters in the mixture models. The first segment is fit to the data points for mixture models $1\cdots k$ and the second segment to the data points for models $k\cdots K$, where $k \in \{2\cdots K-1\}$, assuming $K > 3$. The position of the change point, k, is chosen to minimize the residual sum of squares between the observed data and the piecewise regression line. Once we have selected the location of the changepoint, we choose between the presence and absence of a changepoint (i.e., two-segment piecewise regression versus simple linear regression) using the BIC criterion.

When $K = 3$, there are not enough data points to fit a changepoint model, therefore we determine the presence or absence of a changepoint by computing the angle θ between the two component regression lines, given by $\theta = \arctan(|a-b|/(1+ab))(180/\pi)$ where a and b are the slopes of the two lines. We set an empirical cutoff of $\theta = 1$ degree for identification of a changepoint. Another borderline case is for $K = 2$ clusters, in which case we always return the two component solution. For these borderline cases, the sample is flagged with a warning. In practice, however, we have rarely found cases where the flowClust$_{\text{BIC}}$ fit has $K < 4$ components.

2.5. Identifying Borderline Cases. We flag potential cases where the merging algorithm fails to identify a good solution through several different criteria.

(1) If the number of clusters in the flowMerge solution is equal to the number of clusters in the flowClust$_{\text{BIC}}$ solution.

(2) If the number of clusters in the flowMerge solution is less than the number of clusters in the flowClust$_{\text{ICL}}$ solution.

(3) If no changepoint is detected (BIC chooses no change point model).

(4) If the entropy of the flowMerge solution is unusually high (an outlier) compared to the entropy of the flowMerge solution for comparable samples using the same markers.

In the above cases, samples are flagged for manual inspection of the automated gating. To facilitate the comparison in (4), we normalize the entropy by the number of events in the sample as well as the number of clusters in the merged solution:

$$\text{ENT}_N(K) = \frac{-2\sum_{k=1}^{K}\sum_{i=1}^{N} p_{ik}\log_2(p_{ik})}{NK}. \qquad (5)$$

2.6. The CLL Data Set. We applied the cluster merging algorithm to a real-world data set consisting of 137 samples from 18 individuals with CLL (chronic lymphocytic leukemia) provided by the BC Cancer Agency. The data set is composed of between six and seven samples per individual. Each sample is labeled with three fluorescent markers. The entire panel of markers is designed for immunophenotyping of lymphomas in a clinical setting (Table 2).

We performed automated gating using flowClust on the forward scatter and side scatter channels, followed by cluster merging of the optimal flowClust$_{\text{BIC}}$ solution. We compared the number of clusters obtained by the flowClust$_{\text{BIC}}$, flowClust$_{\text{ICL}}$, and flowMerge solutions. The lymphocyte subpopulation was selected from the merged solution and automated gating was applied to this subpopulation in the fluorescence dimensions. Again, the flowClust$_{\text{BIC}}$, flowClust$_{\text{ICL}}$, and flowMerge solutions were compared, as well as the GMM$_{\text{BIC}}$ solution.

2.7. Simulation. We simulated data from the empirical distribution of a real FCM data set. Based on the CD8 versus CD4 projection of a CLL sample, we estimated the empirical distribution using a two-dimensional kernel density estimator on a 100 by 100 point grid, and sampled 100 data sets of size $N = 9198$ equal to the original number of events. Events were simulated in a two-step process, first we sampled according to the CD8 marginal density derived from the two-dimensional kernel density estimate on a 100 × 100 point grid, then sampled in the CD4 dimension, conditional on the sampled CD8 value, defined by the 100 × 1 element bin of the kernel density estimate. The simulated data sets were gated using the manual gates established on the original data for CD8+/CD4−, CD8−/CD4+, and CD8−/CD4− cell populations (Figure 6(a)). These manual gates were used to calculate misclassification rates for automated gating using the flowClust$_{\text{BIC}}$, flowClust$_{\text{ICL}}$, flowMerge$_{\text{K}}$, and GMM$_{\text{BIC}}$ models with the number of clusters fixed at the true number ($K = 3$) and with the number of clusters chosen by the optimal model.

Table 2: Summary of the antibody markers used in the CLL data.

Antibody combination	Ab1	Ab2	Ab3	No. tubes
1	CD10	CD11	CD20	18
2	CD45	CD14	CD19	18
3	CD5	CD19	CD3	18
4	CD5	CD19	CD38	5
5	CD5	ZAP70	CD19	1
6	CD5	ZAP70	CD3	1
7	CD57	CD2	CD8	4
8	CD57	CD56	CD3	4
9	CD7	CD4	CD8	13
10	FMC7	CD23	CD19	18
11	IgG	IgG	IgG	1
12	IgG1	IgG1/IgG2a	IgG2	13
13	Kappa	Lambda	CD19	18

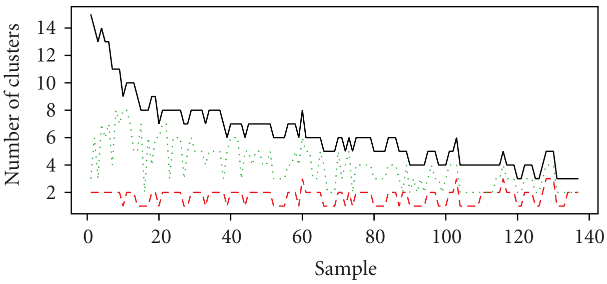


Figure 1: flowClust$_{\text{BIC}}$, flowClust$_{\text{ICL}}$, flowMerge solutions for automated gating of forward versus side scatter across 137 clinical samples of CLL. The flowClust$_{\text{BIC}}$ fit: black solid curve. The flowClust$_{\text{ICL}}$ fit: red dashed curve. The flowMerge fit: green dashed curve.

3. Results

3.1. CLL Data Set. We compared the number of clusters identified by the flowClust$_{\text{BIC}}$, flowClust$_{\text{ICL}}$, flowMerge models used for automated gating of 137 lymph node-derived CLL samples in the forward versus side scatter dimensions (Figure 1). The forward and side scatter data for

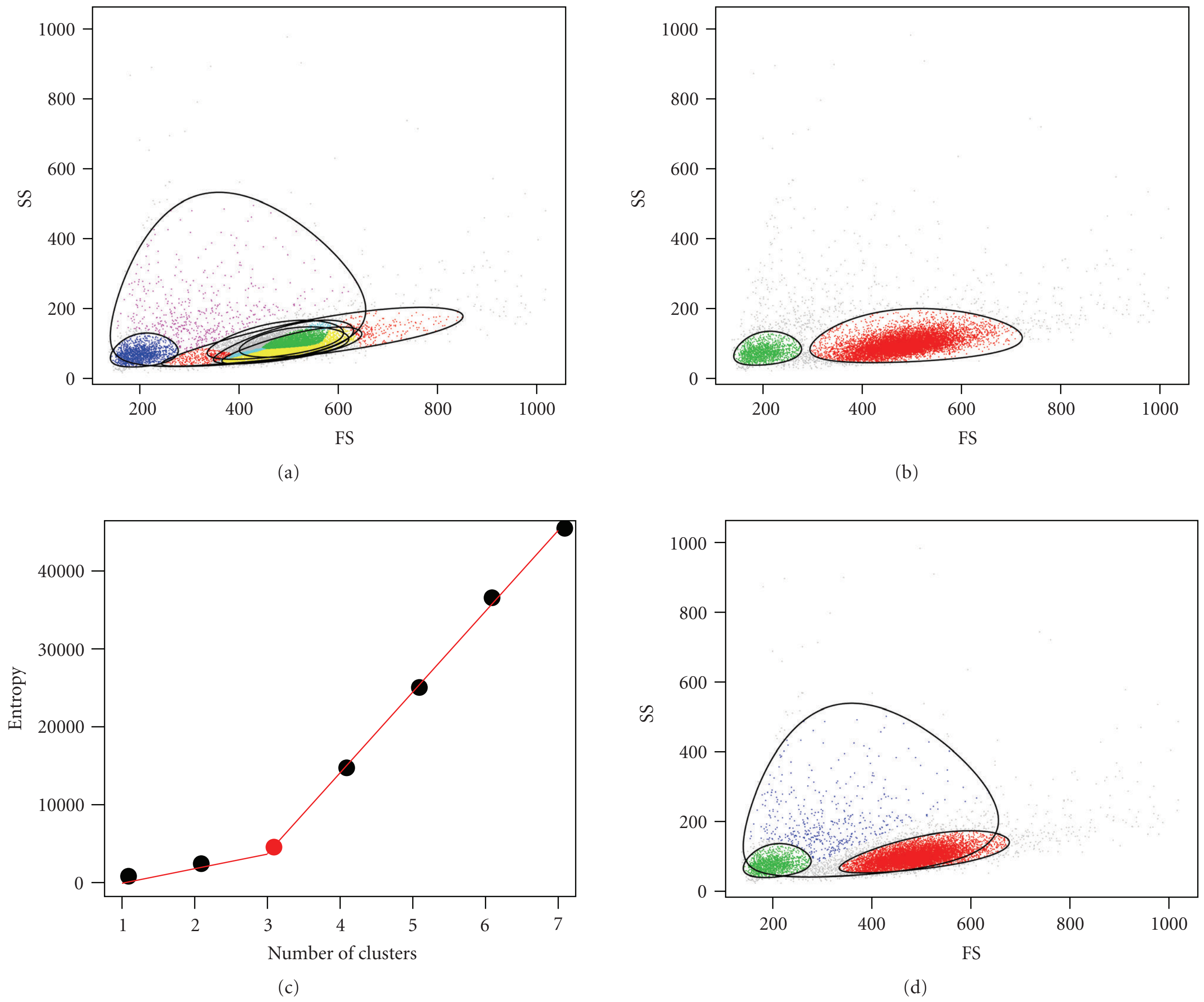


Figure 2: Examples of the flowClust$_{\text{BIC}}$, flowClust$_{\text{ICL}}$, flowMerge cluster solutions for forward versus side scatter in a sample of CLL flow cytometry data. (a) The flowClust$_{\text{BIC}}$ solution with seven clusters. (b) The flowClust$_{\text{ICL}}$ solution with two clusters. (c) The entropy versus number of clusters plot, fit to a two-component piecewise linear regression model. The best fitting model has a changepoint at three clusters. (d) The flowMerge solution corresponding to $K = 3$ clusters provides a better fit to the lymphocyte population than either the flowClust$_{\text{BIC}}$ or flowClust$_{\text{ICL}}$ solutions and provides a good estimate of the true number of cell populations.

these samples contain between two and three predominant cell populations that correspond to lymphocytes, debris, and outliers. The number of clusters identified by the flowClust$_{\text{BIC}}$ solution shows large variability across all samples. This solution generally required more mixture components than the true number of cell populations (median 6 clusters, range 3–15). Importantly, multiple components were often required to model the lymphocyte population (Figure 2(a)), which is the cell population of interest.

In contrast, the flowClust$_{\text{ICL}}$ fit is better but tends to underestimate the true number of cell populations. Across the 137 CLL samples, ICL identified a median of two populations per sample (range from 1 to 3). The ICL also provides a poor fit to the data, inadequately modeling the lymphocyte population (Figure 2(b)).

The flowMerge solution derived from the flowClust$_{\text{BIC}}$ solution provides both a good fit to the underlying data, including the lymphocyte cell population, as well as an improved estimate of the true number of cell populations (Figures 2(c) and 2(d)). The number of clusters estimated through merging is generally between the flowClust$_{\text{BIC}}$ and flowClust$_{\text{ICL}}$ solutions (median of 4 populations, range 2 to 8 clusters).

We performed automated gating in the fluorescence channels on the lymphocyte subpopulation derived from the previous autogating step. In 60/137 cases (43%), the GMM$_{\text{BIC}}$ solution returned more clusters than the flowClust$_{\text{BIC}}$ solution. In 95% of those cases the GMM$_{\text{BIC}}$ fit was within 5 components of the flowClust$_{\text{BIC}}$ fit. These two models returned an equal number of clusters in 29/137 cases (21%), and in 48/137 (35%) of cases, the GMM_{BIC} fit had fewer components. However, in the latter cases, 95% of the samples differed by only a single component (Figure 3, black curve). In general, for the fluorescence dimensions, the $\text{flowClust}_{\text{BIC}}$ model estimated fewer cell subpopulations than the GMM_{BIC} model, in accordance with what is expected, given that the former is a more robust and flexible model.

The $\text{flowClust}_{\text{ICL}}$ fit generally underestimated the number of cell subpopulations and provided a poor fit to the data distribution (Figure 3, red curve and Figure 4(a)). In the example shown, the $\text{flowClust}_{\text{ICL}}$ solution identifies two cell subpopulations in the CD8/CD4/CD7 dimensions and fails to discriminate between the CD4+/CD7+ and CD4+/CD7− cell subpopulations. Additionally, it entirely fails to capture the CD8+ cell subpopulation (Figure 4(a)).

In contrast, for the same sample, the $\text{flowClust}_{\text{BIC}}$ fit requires 13 components and clearly overestimates the number of cell subpopulations. Specifically, the CD4−/CD7−/CD8− cells require multiple mixture components to model a single subpopulation (Figure 4(b)).

The choice of the number of clusters for the flowMerge solution is automated by fitting a piecewise linear model to the entropy versus number of clusters (Figure 4(c)). This solution is derived from the $\text{flowClust}_{\text{BIC}}$ fit and provides a good compromise between model fit and subpopulation identification. It correctly discriminates between the different unique cell subpopulations that were missed by the $\text{flowClust}_{\text{ICL}}$ solution, while combining the overlapping mixture components required to model the CD8/CD4/CD7 negative cell subpopulation in the $\text{flowClust}_{\text{BIC}}$ solution (Figure 4(d)).

We identify cases where cluster merging fails by examining the distribution of the entropy of the flowMerge solution across multiple comparable samples (Figures 5(a)–5(d)). In the forward versus side scatter dimensions, cell populations tend to be complex and overlapping. This is reflected in the distribution of the normalized entropy (Figure 5(a), left). The normalized entropy of the merged solution has a broad distribution (90% of the samples below 0.4, median 0.2) and the solution itself may have many clusters. In contrast, for the fluorescence dimensions, the merged solution identifies well separated populations, reflected by a normalized entropy distribution that is tightly distributed around zero (90% of samples below 0.2, median 0.03) (Figure 5(a), right). We correct for the relationship between the entropy and the number of clusters in the merged solution as well as the number of events by normalizing the entropy (Figure 5(b)). Normalization reduces the correlation of the entropy with the number of clusters ($\rho = 0.38$ versus $\rho = 0.77$ for FS versus SS, and $\rho = 0.08$ versus $\rho = 0.49$ for fluorescence dimensions) (Figure 5(b)). This allows us to identify flowMerge solutions where the entropy is unusually large (in the right tail of the distribution), independent of the number of clusters or events. For forward versus side scatter and for fluorescence channels, we can identify samples where the merged solution contains highly overlapping components (Figure 5(c)). None the less, for forward versus side scatter, the lymphocyte population is sufficiently dense that it can be readily identified visually. Such cases are therefore flagged for manual analysis. Importantly, this criterion allows us to identify general classes of samples where merging fails. We note several sets of markers (notably CD10/CD11c/CD20 and Kappa/Lambda/CD19), where the normalized entropy of clustering is high for all, or a majority of samples (Figure 5(d)). This type of outlier detection is suitable for a high throughput setting to quickly assess flowMerge model fit across groups of parameters and identify those where the automated merging algorithm is problematic. In these cases, again, manual inspection may be required to find an appropriate merged solution. More careful analysis of these cases could suggest strategies to improve automated gating techniques for flow cytometry data.

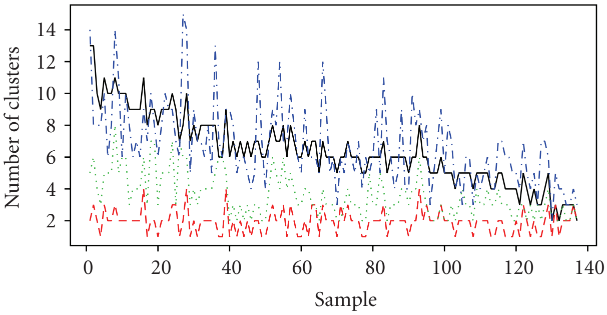


Figure 3: The number of clusters chosen by the $\text{flowClust}_{\text{BIC}}$, $\text{flowClust}_{\text{ICL}}$, flowMerge, and GMM_{BIC} solutions for automated gating of CD8, CD4, and CD7 across 137 samples of CLL. The $\text{flowClust}_{\text{BIC}}$ solution: solid black curve. The $\text{flowClust}_{\text{ICL}}$ solution: dashed red curve. The flowMerge solution derived from the $\text{flowClust}_{\text{BIC}}$ solution: dashed green curve. The GMM_{BIC} solution: dashed blue curve.

3.2. Simulation. We simulated 100 data sets of CD8 versus CD4 fluorescence based on the empirical distribution of real CD8 versus CD4 CLL data. This simulation approach ensured that the simulated data was not biased towards any of the models under investigation. This data had three cell subpopulations defined based on the contours in the CD4 versus CD8 dimensions. These included CD4+/CD8− cells, CD8+/CD4− cells, CD4−/CD8− cells, (outliers were defined by events outside these gates) (Figure 6(a)). No CD4+/CD8+ cell subpopulation could be discerned from the kernel density estimate of this particular sample. We simulated 9198 events per sample (equal to the number of events in the original data) and assigned them to populations based on the manually defined gates from the original data. Kernel density estimates based on simulated data are comparable to the original data (Figure 6(b)).

We compared the number of clusters selected under the optimal $\text{flowClust}_{\text{ICL}}$, $\text{flowClust}_{\text{BIC}}$, GMM_{BIC}, and flowMerge solutions (Figure 6(c)). The $\text{flowClust}_{\text{ICL}}$ solution systematically underestimated the true number of subpopulations (2 clusters estimated in all simulations). The GMM_{BIC} and $\text{flowClust}_{\text{BIC}}$ solutions both significantly overestimated the true number of cell subpopulations in all simulations (median 10 and 9, resp., Figure 6(c)). The median $\text{flowClust}_{\text{BIC}}$solution ($K = 9$ clusters, Figure 6(d)) required two components to model the CD4+/CD8− subpopulation,

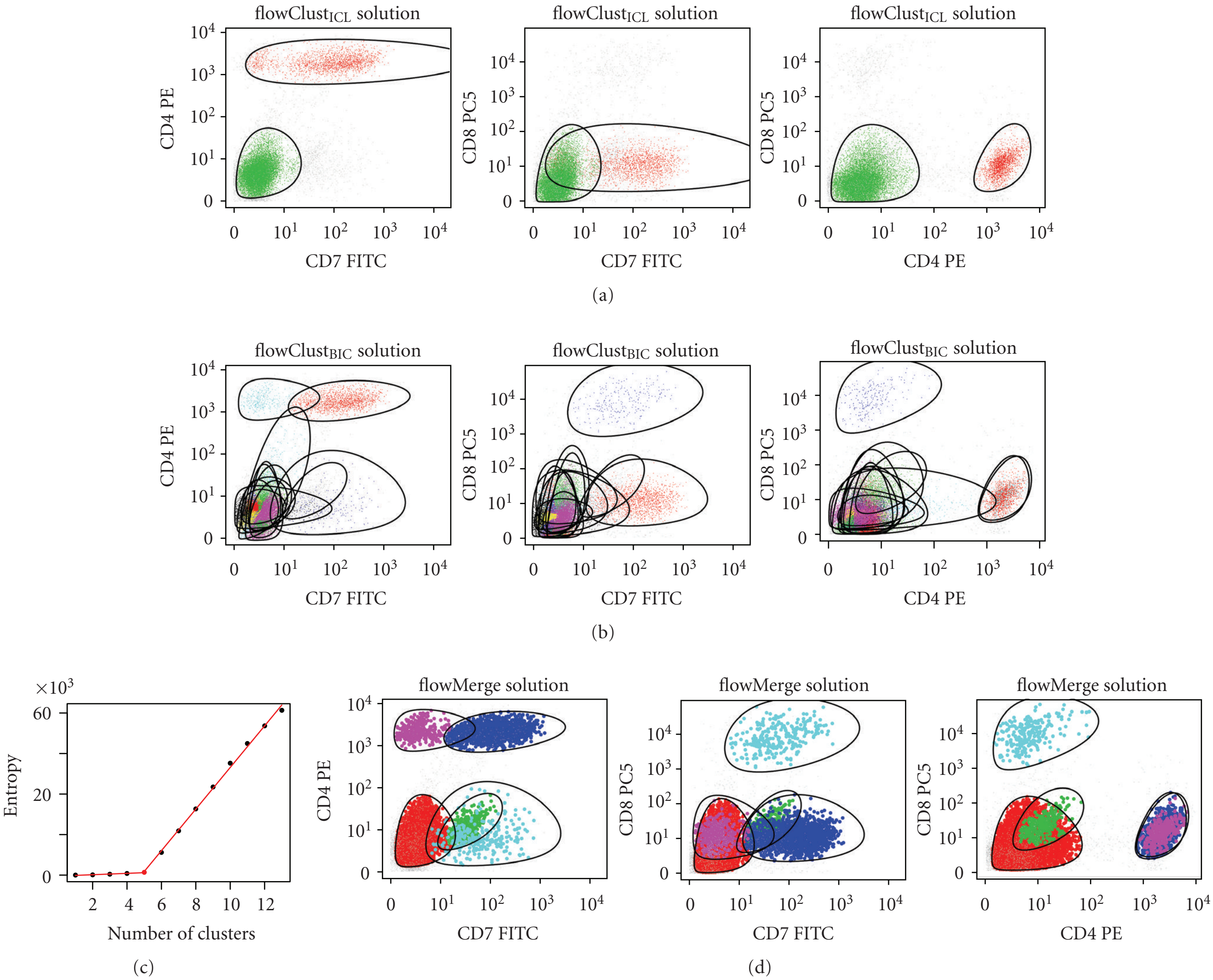
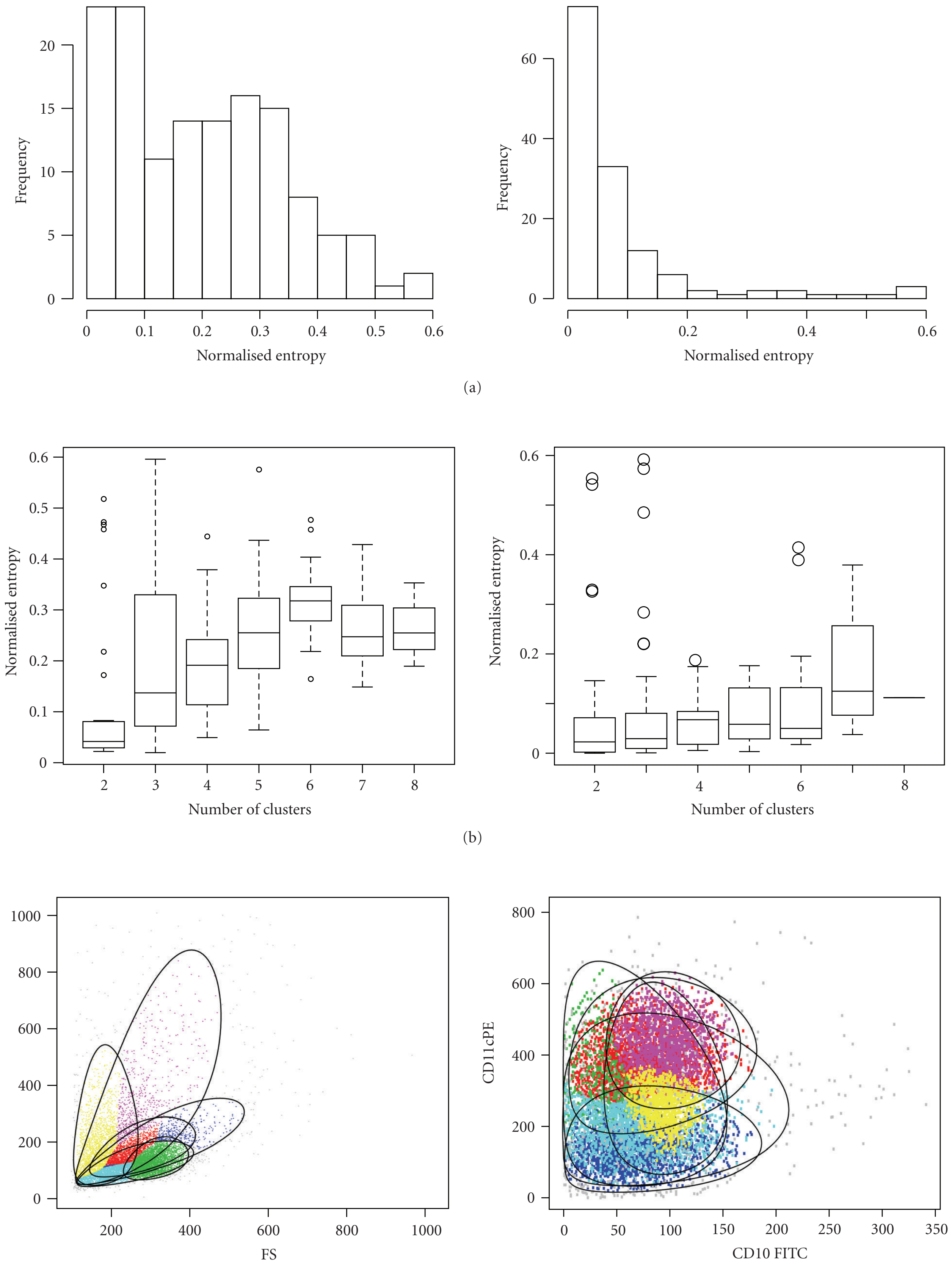


Figure 4: Example of flowClust$_{\mathrm{ICL}}$, flowClust$_{\mathrm{BIC}}$, and flowMerge solutions fitted to a CLL sample in the CD8, CD4, and CD7 dimensions. (a) Three projections of the flowClust$_{\mathrm{ICL}}$ solution. (b) Three projections of the flowClust$_{\mathrm{BIC}}$ solution. (c) Entropy versus number of clusters for a series of flowMerge model fits with a piecewise linear regression fitted to the data. The changepoint located at $K = 5$ clusters is selected automatically. (d) Three projections of flowMerge solution with $K = 5$ clusters derived from the flowClust$_{\mathrm{BIC}}$ solution.

one for the CD8+/CD4− subpopulation, three for the CD4−/CD8− subpopulation, and three components for modeling various outlier low-frequency subpopulations. Although the flowMerge solution overestimated the true number of clusters on average, it provided the closest estimate of the true number of cell subpopulations (median 5). In 16% of simulations, the flowMerge solution estimated the correct number of clusters. In 51% of simulations it overestimated the true number by only one cluster. Closer examination reveals that the extra clusters serve predominantly to model outlier populations (Figure 6(e)). These results are summarized in Table 3.

We also compared the misclassification rates for the different models, relative to class assignments from manual gating. This was done in two ways. First, we fixed the number of clusters to the true number ($K = 3$) for the flowClust$_{\mathrm{K}}$, GMM$_{\mathrm{K}}$, and flowMerge$_{\mathrm{K}}$ models (Figure 6(f)). Note that the former three sets of models are distinct from their "optimal" counterparts by virtue of fixing the number of clusters. Alternately, we compute the misclassification rate between the optimal flowClust$_{\mathrm{BIC}}$, flowMerge or GMM$_{\mathrm{BIC}}$ solutions, choosing the three components from each that minimize the misclassification rate (Figure 6(g)). When the number of components was fixed to the true number, the GMM$_{\mathrm{K}}$ model had the highest misclassification rate (12.3%) (Figure 6(h)), flowClust$_{\mathrm{K}}$ had the second highest misclassification rate (10.5%) (Figure 6(i)), while the flowMerge$_{\mathrm{K}}$ solution (with fixed K) derived from the optimal flowClust$_{\mathrm{BIC}}$ model, had the lowest misclassification rate (4.2%) (Figure 6(j) and Table 3). Both the GMM$_{\mathrm{K}}$ and the flowClust$_{\mathrm{K}}$ solutions with a fixed number of components failed to correctly identify the rare CD8+/CD4− cell subpopulation in the simulated

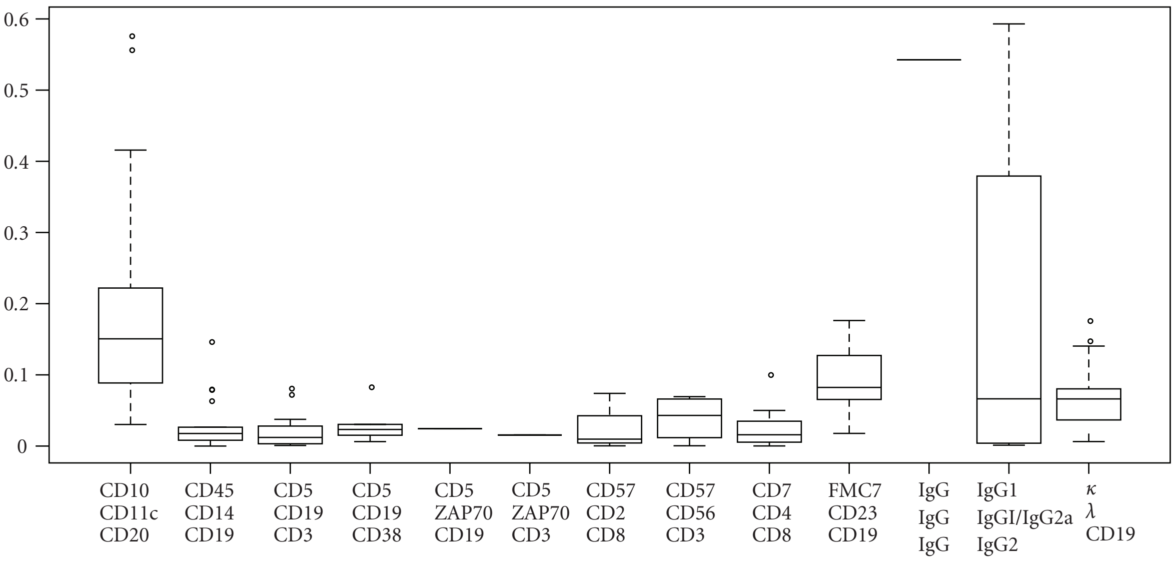


(a)

(b)

(c)

Figure 5: Continued.



(d)

Figure 5: Detecting failed cluster merging. (a) Distribution of the entropy (normalized for the number of events and clusters) of the flowMerge solution for forward versus side scatter (left) and fluorescence channels (right) across 137 samples. (b) The relationship between the normalized entropy and the number of clusters in the flowMerge solution for forward scatter versus side scatter (left) and fluorescence channels (right). (c) Example of flowMerge solutions with unusually high normalized entropy from the right tail of the distribution for forward versus side scatter (left) and fluorescence (right). (d) A plot of the normalized entropy versus samples grouped by antibody labels identifies antibody combinations that are problematic for automated gating with the automated merging algorithm.

Table 3: Mean, standard deviation, 95% coverage, and bias of the estimated number of clusters for each model, as well as the mean, standard deviation and 95% coverage for the misclassification rate of each model. CI: coverage interval.

Statistic	Model	Mean	SD	95% CI	Bias
Number of clusters	flowClust $_{\text{BIC}}$	9.03	1.59	6–12	6.03
	flowClust$_{\text{ICL}}$	2.00	—	2-2	−1.00
	GMM$_{\text{BIC}}$	10.41	1.31	8–12	7.14
	flowMerge	5.45	0.97	4–7	2.45
Misclassification rate ($K = 3$)	flowClust	0.103	0.00826	0.0937–0.112	—
	GMM	0.124	0.00537	0.114–0.134	—
	flowMerge$_{\text{K}}$	0.0445	0.0104	0.0312–0.0669	—
Misclassification rate (best model)	flowClust$_{\text{BIC}}$	0.398	0.101	0.230–0.613	—
	GMMBIC	0.499	0.0756	0.339–0.625	—
	flowMerge	0.0685	0.0223	0.0383–0.121	—

data (Figures 6(h) and 6(i)). In contrast, the flowMerge$_{\text{K}}$ solution correctly identified this subpopulation as a distinct entity.

The misclassification rates for the optimal flowClust$_{\text{BIC}}$, flowMerge, and GMM$_{\text{BIC}}$ solutions were calculated as described, relative to the manually derived gates (Figure 6(g)). These followed a pattern similar to the misclassification rates with a fixed number of components (GMM$_{\text{BIC}}$ was the highest, followed by flowClust$_{\text{BIC}}$, followed by flowMerge). However, in contrast to the fixed component solutions, the misclassification rates for the flowClust$_{\text{BIC}}$ and GMM$_{\text{BIC}}$ solutions were significantly higher than the flowMerge solution (Table 3). This is due to the fact that multiple model components are required to represent distinct cell populations, something only permitted within the cluster merging framework.

4. Discussion

Model-based automated gating of flow cytometry data is difficult when cell subpopulations are nonconvex, or have complicated multidimensional shapes that are not readily

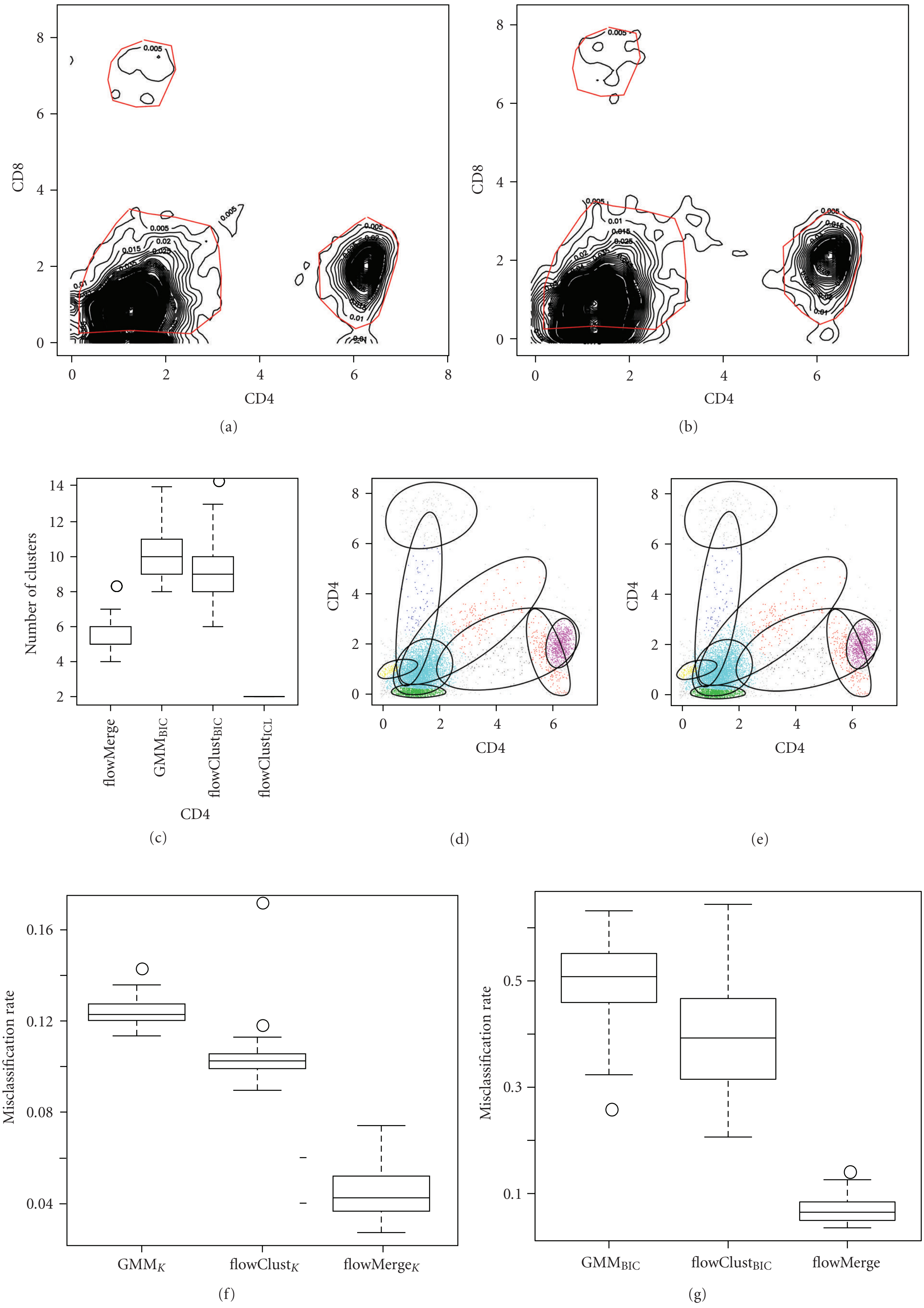


Figure 6: Continued.

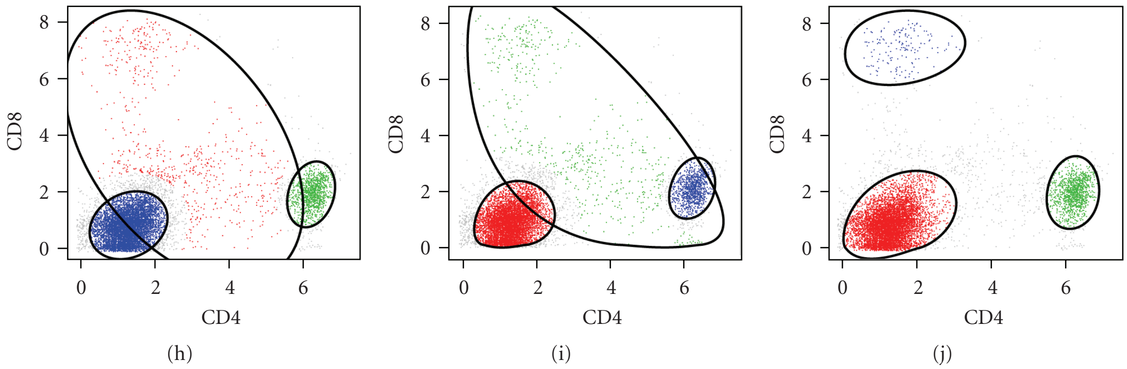


Figure 6: Simulation results for CD4 versus CD8 dimensions of a CLL sample. (a) The 2D kernel density estimate of the real CD4 versus CD8 data. Gates for the CD4+/CD8−, CD8+/CD4−, and CD4−/CD8− subpopulations are represented by light coloured lines. Events outside the gates are considered outliers. (b) An example of the kernel density estimate of simulated data drawn from the distribution defined by the real data. (c) The number of clusters selected by the flowMerge solution, the GMM$_{\mathrm{BIC}}$ solution, the flowClust$_{\mathrm{BIC}}$, and flowClust$_{\mathrm{ICL}}$ solutions over 100 realizations of simulated data. (d) The median flowClust$_{\mathrm{BIC}}$ flowClust solution with 9 components. (e) The median flowMerge solution with 5 components. (f) The misclassification rate (MCR) for the flowMerge$_{\mathrm{K}}$ solution, the GMM$_{\mathrm{K}}$ solution, and the flowClust$_{\mathrm{K}}$ solution with the number of clusters fixed to the true number of cell subpopulations ($K = 3$). (g) The misclassification rates for the three components from the optimal GMM$_{\mathrm{BIC}}$, flowClust$_{\mathrm{BIC}}$, and flowMerge solutions minimizing the MCR. (h) A GMM, (i) flowClust, (j) and flowMerge$_{\mathrm{K}}$ solution with a fixed number of clusters.

modeled by single components of simpler multivariate distributions. This issue is resolved, in part, by allowing multiple mixture components to represent the same cell subpopulation. However, for further analysis, cell subpopulations are generally summarized by a variety of statistics; this requires one to summarize an arbitrary number of mixture components for a single cell subpopulation. Consequently the cluster merging algorithm is not suitable for application to flow cytometry data without further modifications. By taking advantage of the fact that a merged cluster is itself a mixture (see (2)), and approximating the merged distribution as a density from the same family as its components, we use moment matching to summarize the merged cluster with a single set of parameters that provides a good approximation to the underlying data (see (3) and (4)). This simple representation of otherwise complicated distributions allows downstream data analysis to proceed in the usual manner and fits within the existing flowClust framework, allowing for easy visualization of automated gating results.

Comparison of the cluster merging algorithm with other automated gating models (Table 1) using both simulated and real data demonstrate that merging provides a better fit and better estimate of the true number of cell subpopulations than the other models. Estimates of the number of cell populations derived from standard model-selection measures such as BIC or ICL are not entirely suitable for flow cytometry data (Figures 2 and 4). BIC, while providing a good fit to the data, requires many more clusters than actual number of cell subpopulations, while ICL underestimates the number of cell subpopulations and provides a poor fit to the data, missing both rare cell subpopulations and poorly fitting those that have complicated structure (Figures 4(a), 4(b) and Table 3). The flowMerge solution provides a good compromise between these two extremes. It is based on the flowClust$_{\mathrm{BIC}}$ solution, thus retaining the property of good fit to the distribution, while simultaneously eliminating ambiguity associated with multiple overlapping components representing the same cell subpopulation. Merging decreases the entropy of clustering by making local changes to the model without compromising the global fit.

We use a changepoint model to estimate the optimal number of clusters in the merged solution. This allows the cluster merging algorithm to be implemented in a high-throughput pipeline for flow cytometry data analysis. In general, this approach provides satisfactory results, both for forward versus side scatter dimensions as well as for fluorescence dimensions (Figures 1 and 3). The number of clusters chosen by flowMerge is generally between the flowClust$_{\mathrm{BIC}}$ and flowClust$_{\mathrm{ICL}}$ solutions, and although it still tends to overestimate the number of cell subpopulations by several components, these generally model outlier cell subpopulations (Figure 2(d) and 6(e)). Interestingly, our simulation results also show that our framework for summarizing merged components allows some of these outlier subpopulations to be merged with clusters representing more dense cell subpopulations, of interest, without adversely affecting the fit of the model. This is due to the fact that the parameters of merged clusters are weighted linear combinations of the parameters of the component clusters. Therefore components of lower density contribute less to the mean and covariance parameters of merged clusters (Figures 6(e)–6(g)).

Our results on real flow data demonstrate that the cluster merging algorithm improves our ability to identify the lymphocyte cell subpopulation from the forward versus side scatter dimensions. This high density subpopulation is often represented by multiple mixture components in the flowClust$_{\mathrm{BIC}}$ and GMM$_{\mathrm{BIC}}$ solutions. Merging allows this subpopulation to be represented by a single model component (Figure 2). Even in cases where merging fails, the algorithm is sufficiently robust that prior information about the expected number of cell populations could be used to identify an appropriate merged solution manually, while retaining a good fit to the data distribution (Figures 6(d) and 6(j)). Others have suggested incorporating information from the repeated-measures design of some flow cytometry data sets to help make gating decisions [16]. The application of cluster merging for identification of cell populations in the fluorescence dimensions is also beneficial. It reduces the complexity of subpopulations represented by multiple components. A comparison of the flowClust$_{\text{BIC}}$ and flowClust$_{\text{ICL}}$ solutions shows that these two criteria tradeoff model fit against a simpler representation of cell subpopulations (Figures 4(a) and 4(b)). The flowClust$_{\text{ICL}}$ solution frequently fails to correctly identify all but the highest density regions; whereas the flowClust$_{\text{BIC}}$ solution often overestimates the number of clusters in high density regions.

Our cluster merging framework provides a robust modeling approach for automated gating of flow cytometry data. It provides a good compromise between the flowClust$_{\text{BIC}}$ and flowClust$_{\text{ICL}}$ solutions by combining the good model fitting characteristics of BIC-based model selection with a more modest estimate of the true number of clusters, a characteristic of the ICL-based model selection. It allows us to represent complicated cell populations using single mixture components for which we can readily obtain closed-form parameter estimates for use in further analysis. Additionally, these estimates are robust to outlier cell populations. The cluster merging approach to gating has a lower misclassification rate than other models considered here, irrespective of whether the number of clusters was fixed at the true number or chosen from amongst the components in the optimal fitting model. Together, these factors make cluster merging a powerful tool for automated gating of flow cytometry data.

Acknowledgments

The authors thank Andrew Weng and Randy Gascoyne for providing the flow cytometry data. This work was supported by the NIH Grant EB008400, by the Michael Smith Foundation for Health Research (for the second and third authors), and by the Natural Science and Engineering Research Council of Canada (for the second author).

References

[1] J. W. Gratama, J. Kraan, M. Keeney, V. Granger, and D. Barnett, "Reduction of variation in T-cell subset enumeration among 55 laboratories using single-platform, three or four-color flow cytometry based on CD45 and SSC-based gating of lymphocytes," *Clinical Cytometry*, vol. 50, no. 2, pp. 92–101, 2002.

[2] C. Satoh, K. Dan, T. Yamashita, R. Jo, H. Tamura, and K. Ogata, "Flow cytometric parameters with little interexaminer variability for diagnosing low-grade myelodysplastic syndromes," *Leukemia Research*, vol. 32, no. 5, pp. 699–707, 2008.

[3] M. Van Blerk, M. Bernier, X. Bossuyt, et al., "National external quality assessment scheme for lymphocyte immunophenotyping in Belgium," *Clinical Chemistry and Laboratory Medicine*, vol. 41, no. 3, pp. 323–330, 2003.

[4] R. Achuthanandam, J. Quinn, R. J. Capocasale, P. J. Bugelski, L. Hrebien, and M. Kam, "Sequential univariate gating approach to study the effects of erythropoietin in murine bone marrow," *Cytometry Part A*, vol. 73, no. 8, pp. 702–714, 2008.

[5] M. J. Boedigheimer and J. Ferbas, "Mixture modeling approach to flow cytometry data," *Cytometry Part A*, vol. 73, no. 5, pp. 421–429, 2008.

[6] K. Lo, R. R. Brinkman, and R. Gottardo, "Automated gating of flow cytometry data via robust model-based clustering," *Cytometry Part A*, vol. 73, no. 4, pp. 321–332, 2008.

[7] M. Roederer and R. R. Hardy, "Frequency difference gating: a multivariate method for identifying subsets that differ between samples," *Cytometry*, vol. 45, no. 1, pp. 56–64, 2001.

[8] C. A. Sugar and G. M. James, "Finding the number of clusters in a dataset: an information-theoretic approach," *Journal of the American Statistical Association*, vol. 98, no. 463, pp. 750–763, 2003.

[9] G. McLachlan and D. Peel, "Robust cluster analysis via mixtures of multivariate t-distributions," in *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, vol. 1451 of *Lecture Notes in Computer Science*, pp. 658–666, January 1998.

[10] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[11] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 719–725, 2000.

[12] J. Baudry, A. Raftery, G. Celeux, K. Lo, and R. Gottardo, "Combining mixture components for clustering," Tech. Rep., Universite Paris Sud, University of Washington, University of British Columbia, Vancouver, Canada, August 2008.

[13] S. Pyne, X. Hu, K. Wang, et al., "Automated high-dimensional flow cytometric data analysis," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 21, pp. 8519–8524, 2009.

[14] R. C. Gentleman, V. J. Carey, D. M. Bates, et al., "Bioconductor: open software development for computational biology and bioinformatics," *Genome Biology*, vol. 5, no. 10, article R80, 2004.

[15] S. Richardson and P. J. Green, "On bayesian analysis of mixtures with an unknown number of components," *Journal of the Royal Statistical Society. Series B*, vol. 59, no. 4, pp. 731–792, 1997.

[16] U. Naumann and M. P. Wand, "Automation in high-content flow cytometry screening," *Cytometry Part A*, vol. 75, no. 9, pp. 789–797, 2009.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2009, Article ID 809469, 8 pages
doi:10.1155/2009/809469

Research Article

Bridging the Divide between Manual Gating and Bioinformatics with the Bioconductor Package flowFlowJo

John J. Gosink,[1] Gary D. Means,[2] William A. Rees,[2] Cheng Su,[3] and Hugh A. Rand[1]

[1] *Department of Computational Biology, Amgen, 1201 Amgen Court West, Seattle, WA 98119, USA*
[2] *Department of Molecular Sciences, Amgen, 1201 Amgen Court West, Seattle, WA 98119, USA*
[3] *Department of Biostatistics—Medical Sciences, Amgen, 1201 Amgen Court West, Seattle, WA 98119, USA*

Correspondence should be addressed to Hugh A. Rand, rand@amgen.com

Received 31 March 2009; Revised 19 June 2009; Accepted 19 July 2009

Recommended by Raphael Gottardo

In flow cytometry, different cell types are usually selected or "gated" by a series of 1- or 2-dimensional geometric subsets of the measurements made on each cell. This is easily accomplished in commercial flow cytometry packages but it is difficult to work computationally with the results of this process. The ability to retrieve the results and work with both them and the raw data is critical; our experience points to the importance of bioinformatics tools that will allow us to examine gating robustness, combine manual and automated gating, and perform exploratory data analysis. To provide this capability, we have developed a Bioconductor package called flowFlowJo that can import gates defined by the commercial package FlowJo and work with them in a manner consistent with the other flow packages in Bioconductor. We present this package and illustrate some of the ways in which it can be used.

1. Introduction

Flow cytometry is a high-information content platform that is increasingly becoming a high-throughput platform as well [1]. Flow cytometers measure individual cells, and thus are capable of revealing subtleties of biology that other technologies cannot detect. Recent advances in instrumentation such as 4 and 5 color laser systems and the availability of reagents and protocols for assessing internal proteins and their phosphorylation state are serving to make flow cytometry a very important tool for understanding disease processes in human biology [2]. There is also a growing appreciation that it is important to assess cells not only in their quiescent state, but also in response to various stimuli [3]. This adds another layer of complexity to flow cytometry data sets. Powerful analysis tools are needed to properly explore and analyze data sets in which each sample has many stimuli, cell subpopulations, and phosphoprotein measurements.

There are a number of challenges associated with the analysis of these large, complex flow cytometry data sets. The challenges can be divided into. (1) acquisition of high-quality data, (2) tools for data organization, annotation, and query, (3) tools for data manipulation, and (4) techniques and statistical methods for data analysis. All of these components are related and, done well, serve to reinforce each other. The first two of these tasks tend to be application- and lab-specific, while the latter two lend themselves well to the development of shared tools for all those faced with complex flow cytometry analyses. Similar to tools developed for microarrays, a set of packages is evolving in the Bioconductor community that holds great promise for flow cytometry data analysis. These packages which include flowCore [4, 5], flowQ, flowViz, flowUtil, flowStats, flowClust [6] and others all operate on a common set of core methods and classes for reading, transforming, gating and otherwise manipulating flow cytometry data.

In the analysis of flow cytometry data it is important to be able to work with the gates that have been manually defined. Commonly these gates are defined in a commercial flow cytometry analysis package that is used, along with "cut-and-paste" and simple analysis packages such as Excel or Prism, to provide results. This becomes problematic when dealing with complex problems and large data sets.

To address this problem, we have built a package that provides a way to extract data from one such commercial package, FlowJo (http://www.flowjo.com/), into the publicly accessible analysis platform R/Bioconductor. We chose to use FlowJo because it is amongst the most commonly used flow cytometry programs and it stores its session information in an open format. The package flowFlowJo can produce R data structures with either summary statistics or fully flowCore compliant objects representing the various gates, compensation matrices, and other related information embedded in FlowJo sessions. The goal of flowFlowJo is to make it easy, in R, to use compensation and gating information that has been produced using FlowJo. The flowFlowJo package provides the ability to work with both the raw data and the gating information in a powerful analysis environment that makes full use of the existing open source community efforts.

2. Software

2.1. Overview of the flowFlowJo Package. FlowJo is a commercially available software package used for the gating, visualization, and analysis of data from flow cytometry experiments. FlowJo saves its session information in an eXtensible Markup Language (XML) text file called a *workspace*. A workspace file contains all the information necessary to describe the gating structures, compensation, transformations, locations of the Flow Cytometry Standard (FCS) [7] files, graphs, and figures created by the user. FlowJo workspace files do not contain raw cytometry data.

The R package flowFlowJo is a set of methods and classes designed to extract the file locations, gates, compensation matrices, and some of the other information contained in FlowJo workspace files and return the information in a manner consistent for use with the Bioconductor flowCore packages. The flowFlowJo package can execute the following actions when supplied with the location of one or more FlowJo workspaces:

(1) read and parse the workspace(s),

(2) extract the location of all of the FCS files referenced in the workspace(s),

(3) extract all of the intermediate and final gates as flowCore S4 class filters objects,

(4) extract the spillover matrices,

(5) extract the transformation settings,

(6) organize the extracted information into a set of data structures so that all of the compensation and gating strategies described in the workspace(s) can be executed in R. In effect, this captures and executes much of the analysis workflow stored in the FlowJo workspace,

(7) return a set of identically ordered lists containing all of the file locations, file names, filter objects, filter names, and compensation matrices.

These operations are typically done by an analyst using flowFlowJo in order to

(1) produce summary tables of the names and numbers of gates described in the workspace(s),

(2) execute the complete set of gatings described in the workspace, returning a comprehensive table of summary statistics for all of the populations for each of the channels,

(3) obtain a set of ordered lists of FCS file paths, spillover matrices, and flowCore S4 filter objects identical with that created by the researcher using FlowJo. These objects can then be used in a more detailed event-level analysis than would be possible from simple summary statistics alone.

Figure 1 illustrates how the major components of the flowFlowJo package are related in typical data analysis sessions. The following code examples demonstrate part of such an R session using flowFlowJo to analyze a set of cytometric data. In the first line of this example the analyst reads in a FlowJo workspace from a file on his system. In the second line the analyst obtains a list of all the files and gate names referenced in the workspace to ensure that correct number and types of gates have been obtained. For brevity, the contents of this call are not shown in the demo below. In the third line the analyst "executes" the workflow detailed in the workspace via the *collectSummaryFlowInfo* command to assemble a complete set of summary statistics on all of the FCS files and all of the gates described in the workspace. In this example, the analyst also instructs the code to recover the photomultiplier tube voltage setting as recorded in each FCS file via the `keywords` argument. In fact, the `keywords` argument allows the analyst to recover any of the metadata embedded within the header section of each FCS file. The list of possible keywords and their values can be found for any FCS file with the standard flowCore call, *keyword*. In the fourth line of code, the analyst converts the complex summary object to a standard R data structure while merging it with additional metadata describing experimental details:

```
   fjListObj <- readFlowJoList("C://Documents
and Settings/TestFlowJoFile.wsp")
   gateAndFileInfo <- getFlowJoSummary
(fjListObj)
   summaryStatsObj <- collectSummaryFlowInfo
(fjListObj, keywords=c("$P1V"))
   flowReport <- createFlowReport
(summaryStatsObj, extraMetaDataFrame)
```

The analyst then works with the resulting standard R data structure to produce reports and analyses as needed. The above code provides only summary statistical information on the populations delineated in the workspace. However in some cases the analyst may wish to examine the distribution of data within a population much more carefully or gain event by event access to the cells within a population. The *getFlowJoGates* command as invoked in the first line of the example session below returns an ordered list-of-lists containing all of the file locations, file names, compensation matrices, gate names and flowCore compliant filter objects corresponding to all of the FCS files with the regular expression "Specimen.*C01" in their full pathname. As discussed above, the *getFlowJoSummary* command will return the full set of file names referenced in the in a FlowJo workspace from which the analyst may wish to choose a subset via the `fileNamePatterns` argument. The default for the `fileNamePatterns` argument returns the information for all of the FCS files referenced in the workspace. In the second and third lines below, an FCS file is loaded into memory and compensated. The fourth and subsequent lines illustrate standard flowCore operations on the associated "CD3+:Lymphocyte" filter object. The *summary* command shows the number and percent of cells recorded in the FCS file that fall within the boundaries of the CD3+: Lymphocyte gate. Finally the gate is adjusted by moving each of its forward scatter polygon coordinates 10% higher:

```
   gateList <- getFlowJoGates(fjListObj,
fileNamePatterns=c ("Specimen.*C01"))
   aFlowFrame <- read.FCS
(gateList$FCSFilename[[1]])
   aFlowFrame <- flowJoCompensate
(aFlowFrame, gateList$compMats[[1]])
   aFilter <- gateList$filter[[1]]
   aFilter
   filter 'Specimen_001_C1_C01.fcs:
Lymphocytes: CD3+'
   the intersection between the 2 filters
   Polygonal gate 'Specimen_001_C1_C01.fcs:
Lymphocytes' with 6 vertices in dimensions
FSC-A and SSC-A
   Rectangular gate 'Specimen_001_C1_C01.fcs:
CD3+' with dimensions:
    Pacific Blue-A: (337.211599131617,
5996.56443562053)
    PE-A: (11.0542047560856, 37903.8875296341)
   summary(filter(aFlowFrame, aFilter))
   Specimen_001_C1_C01.fcs:Lymphocytes: CD3++:
14342 of 99286 events (14.45%)
   summary(filter(aFlowFrame, aFilter@filters
[[1]]@boundaries [,"FSC-A"] * 1.1))
   Specimen_001_C1_C01.fcs:Lymphocytes:CD3++:
13043 of 99286 events (13.14%)
```

As can be seen, the types of operations that can be conducted at this point are virtually limitless. The *getFlowJoGates* method simply provides the user with all of the relevant components found in the FlowJo workspace as R and flowCore compliant objects in a set of commonly ordered lists.

2.2. File Locations, Gates/Filters, Spillover Matrices Compensation Matrices and Transformations. Prior to using the flowFlowJo package, FlowJo will have been used to manually process (compensate and gate) one or more FCS files to produce one or more FlowJo workspaces. This is a routine process for those analyzing flow cytometry data. Worth noting is that the location of the FCS files is stored in the FlowJo workspace as absolute or relative paths. Moving the FCS files to another location will cause the location of these files as extracted from the workspace to be in error and further processing steps on these files will be impossible. In anticipation of this possibility, the *readFlowJoList* method allows the user to specify an alternate path for the referenced FCS files.

It is common practice that an assay is performed over many weeks or months, with the data from each day's run being accumulated into a single FlowJo workspace. Furthermore, it is not uncommon for the files containing the data from various runs to be given the same names. The package flowFlowJo allows for this by reading in any number of FlowJo workspaces at the same time and tracking the location of the FCS files by their full pathname.

Some inconsistencies appear in the use of terminology in flow cytometry software and literature with respect to compensation matrices. FlowJo workspaces include sections labeled "CompensationMatrix" which are more properly referred to as "spillover" matrices. The spillover matrix elements represent the proportion of the signal emitted by each fluorescent dye that falls within the band pass windows for each of the other fluorescent dyes. The compensation matrix is the inverse of this matrix. Currently, in order to obtain similar results (e.g., mean fluorescent intensities and cell counts) between FlowJo and flowCore, it is necessary to multiply the observed signal values by the spillover matrix to the data with the usual flowCore method call (*compensate*) and then to divide all of the observed fluorescent (nonscatter) data by the maximum of the values in the spillover matrix. The flowFlowJo package implements an internal method, *flowJoCompensate*, to automatically take care of this issue when generating summary statistics. It is also worth noting that FlowJo (and flowFlowJo) allow for a different spillover matrix for each FCS file referenced within each workspace.

Standardized interpretation of the gating coordinates can also be problematic. The information contained within the DivaSettings and TransformSettings sections of the FlowJo workspace records the user's preference for gating visualizations. This data is parsed and returned by the *readFlowJoList* method. However, all the fluorescence channel coordinates are encoded by FlowJo in their nontransformed gate coordinates. Hence there are no methods in flowFlowJo that currently utilize transformation and "DivaSettings" data, since they appear to have no impact on the obtained results. Additionally, due to code legacy, FlowJo reads the scatter gate data of FCS files in only 12 bit resolution (i.e., a maximum value of 4096). However modern flow cytometers typically record integrated signal intensities at 18 bit resolution (i.e., a maximum value of 262143). Thus the forward and side scatter gate coordinates are currently (FlowJo 7.2.5) encoded as 1/64 of their actual values for 18 bit FCS files. In these cases the *readFlowJoList* method automatically (internally) multiplies each of the scatter gating coordinates by 64 to adjust for this prior to generating flowCore filter objects.

2.3. Data Summary Objects. The first step in automating the analysis of manually gated data is to ensure uniformity of the naming convention across all of the samples and to confirm

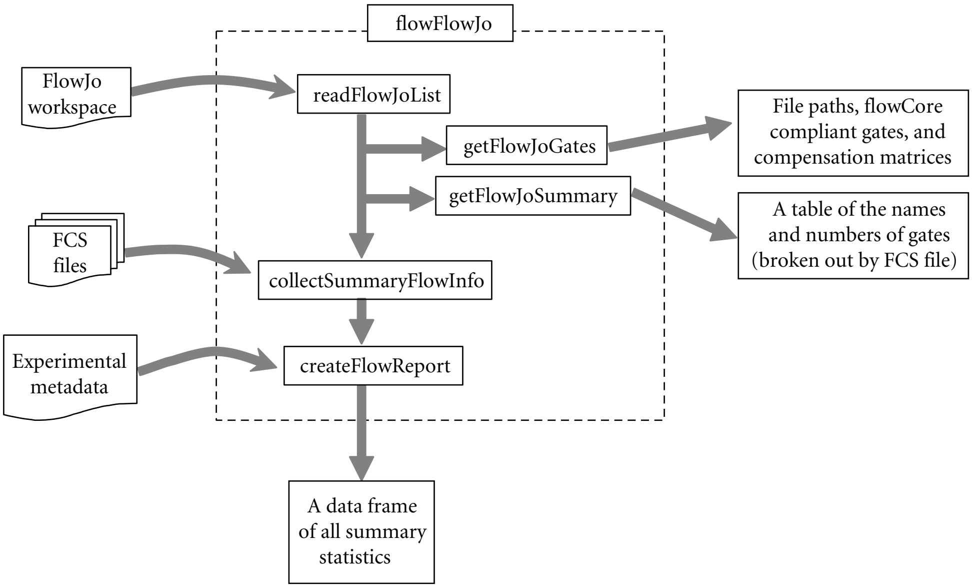

flowFlowJo
FlowJo workspace
readFlowJoList
getFlowJoGates
getFlowJoSummary
File paths, flowCore compliant gates, and compensation matrices
A table of the names and numbers of gates (broken out by FCS file)
FCS files
collectSummaryFlowInfo
Experimental metadata
createFlowReport
A data frame of all summary statistics

Figure 1: Diagram of the major methods of the flowFlowJo package and their relationship in typical use.

that all of the expected data is present. With larger data sets, problems may include (1) different names for the same cell populations, (2) missing gates, (3) missing samples, and (4) unexpected gates or samples. Such unanticipated deviations from the experimental plan can become buried in a large set of data and often compromise the downstream data analysis. A simple summary of the data is useful for identifying these anomalies. Toward this end, the *getFlowJoSummary* method returns a table showing the number and counts of different gate names associated with all of the FCS files in one or more FlowJo workspaces.

In some cases, a data analyst may wish to proceed manually in R with the organized lists of FCS files, filters, and spillover matrices extracted by flowFlowJo. This can be accomplished with the *getFlowJoGates* method described above. However, in many cases, the analyst may be satisfied with the gating choices created in FlowJo, and may wish to simply acquire a complete set of summary statistics on all of the cell populations. The FlowFlowJo package provides methods to automatically "execute" the gating strategy provided in the workspace. It is only at this point that the flowFlowJo methods actually access the FCS files. The *collectSummaryFlowInfo* method systematically employs standard flowCore methods to create a data structure summary object with median fluorescent intensities and cell counts for each of the channels for each of the populations, as well as any requested header information from each of the FCS files.

Each FCS file is composed of several sections in addition to the raw list-mode data. The header section of each FCS file typically contains 100 or more pieces of information about each flow run, including laser settings, photomultiplier tube voltages, run times, and other information. The *collectSummaryFlowInfo* method can be configured to collect one or more of these items from each FCS file. As a practical matter, since each FCS file may be quite large, the code only reads one FCS file into memory at a time, extracts the appropriate information, and frees its memory before moving on to the next file.

Finally, the *createFlowReport* method can combine the summary object with additional metadata about the experiment such as sample information or treatment conditions. The resulting flow report will contain one line for each channel of each cell population of each FCS file along with any associated metadata and keywords from the header section of the FCS file.

There are a wide variety of possible gate types within FlowJo. The current version of flowFlowJo can process range, rectangle, polygon, quadrant, and "auto" gates. Elliptical gates are not currently supported. With the advent of FlowJo version 7.5, the gate descriptions in the workspace are expected to be consistent with the Gating-ML standard [8], and we will be upgrading flowFlowJo to handle all gate types produced by FlowJo. Additional detail on the use of flowFlowJo is contained in the vignette that is available through Bioconductor (http://www.bioconductor.org/).

3. Applications

In the following sections we describe two applications in which we believe it is beneficial to have computational access to manually defined gates. These two applications are intended to illustrate how flowFlowJo, by allowing for computational access to manually defined gates, will make it easy to address questions and concerns about gates and the gating process. We hope that flowFlowJo will provide for an easier comparison of manual and automated gating approaches and improve our confidence in different gating procedures.

3.1. Supporting Reproducible, Semiautomated Flow Cytometry. In our experience, flow cytometry is commonly practiced in one of two ways. The first way occurs when a small number of samples are evaluated as part of an ongoing process of hypothesis generation and testing. The second way occurs in the clinical lab, a highly-regulated, high-throughput environment, in which there is little room for exploration or follow-up. In our view there is an important need for a third option in flow cytometry. This third way (which we call reproducible, semiautomated flow cytometry) supports the manual, exploratory analysis which is easy to do in software tools such as FlowJo or FCSExpress, but also allows for the type of modeling and analysis that has proven beneficial in the microarray arena. In addition, reproducible, semiautomated flow cytometry should have the potential to retain, and even improve upon, many of the benefits available in the highly regulated clinical environment.

The flowFlowJo package supports a reproducible, semiautomated system in three primary ways. First, the package supports the use of FlowJo, which provides the bench researcher with a familiar tool for the visualization and exploration of flow data. Secondly, flowFlowJo moves all computations on the original data set into the R programming environment—thus allowing for automation and reproducibility of analysis statistics [9]. These summary statistics can then be readily exported into other visualization tools such as SpotFire. Third, the availability of all the data in R allows the use of a wide range of sophisticated statistical analysis tools. Data visualization and analysis often raises questions pertaining to the gating of cell populations. These questions can be readily explored because all of the postgating analyses can be automated.

3.2. Gating Robustness. Gating is an important and often time-consuming component of the analysis of large flow cytometry data sets. The delineation of the boundaries of cell populations is often made difficult by variable numbers, size, shape, and location of both target and nontarget cell populations. This variability may be due to debris arising from problems with sample handling or reagents, or may be due to changes in cell populations arising from disease or specific genetic differences. These problems may only become apparent in the midst of a large project, and it can be problematic to preemptively design an algorithm or model capable of handling such unforeseen problems. The difficulty of automating the pattern recognition of (potentially) distorted objects in the presence of noise is recognized in other fields [10] as well, in which the human ability to identified distorted words and characters is relied upon. While manual gating is relatively robust to unanticipated cell population distributions, it suffers from the potential for operator bias. In fact all gating methods have their drawbacks in particular cases, and tools and procedures are needed for evaluation of the results of the gating process.

It is important to be able to assess the robustness of gating results irrespective of the method employed, and some relatively robust approaches do exist [11, 12]. In general, the results of an experiment are considered robust if they are not sensitive to small changes in the assumptions or methods used to arrive at the results. To assess gating robustness in flow cytometry, it is extremely useful to be able to work with gates in a computational framework. There are at least three intertwined aspects of gating robustness that are important to assess: gating method, gate method tuning, and gate homogeneity. For illustrative purposes, we focus on the first and only briefly comment on the other two.

As an illustration of the assessment of gating method robustness, we examined a set of human blood samples run in a single 96-well plate. These samples originated from blood drawn from four healthy donors that were stimulated *ex-vivo* with various levels of TNF-α by three operators (resulting in a total of 96 samples) as part of an assay development program. The samples were stained with a variety of different antibodies, of which we only consider the antibodies for CD3/CD14 and P-p38 (the phosphorylated form of mitogen-activated protein kinase 14) as expressed by the monocytes. The antibodies to CD3 and CD14 were both conjugated to the same dye because cells staining for either of these markers can be distinguished in the SSC channel, thus allowing for the use of more channels for other markers of interest. P-p38 is intracellular and was detected by an experimental protocol in which the cells were permeabilized.

Monocytes were gated in several ways in order to assess the robustness of results to the choice of gating method employed. For method I gates were created manually in FlowJo using a polygon gate drawn on a SSC versus $\log_{10}$(CD3/CD14) bivariate plot. The gates for method II were obtained with a robust normal fit via the *fitNorm2* method from the R package prada [13] on the cells gated with method I. The *fitNorm2* method uses a contour level for the resulting bivariate normal distribution chosen as the gate boundary [13]. Method III found the intersection of manually gated cells from method I with regions of significant curvature obtained via the *featureSignif* method in the R feature package which fit a two dimensional probability density function [14] to all of the SSC and CD3/CD14 data for each flow file. Results were then compared across the three methods for all samples. Each of these methods has one (or more) tuning parameters which can be used to make results match very closely for any individual sample between the three different methods. It is the agreement across methods for all the samples that is of importance. For method I (manual gating) the operator created a polygon gate using as many vertices placed in whatever locations were deemed appropriate. For method II and III, the various tuning parameters were selected so as to provide results close to those obtained with method I. It should be noted that the results for methods II and III are by design subsets of the results obtained by method I.

Figure 2 shows the gates obtained by the three methods for two of the blood samples. These two cases bracket the range of observed agreement between gates; very good for sample H03, and poorer for sample B10. For every gate, the response of the cells in that gate as measured by their mean P-p38 levels was computed (the level of response for each sample is driven primarily by the TNF-α stimulation level). Comparison of response measures between the three gating

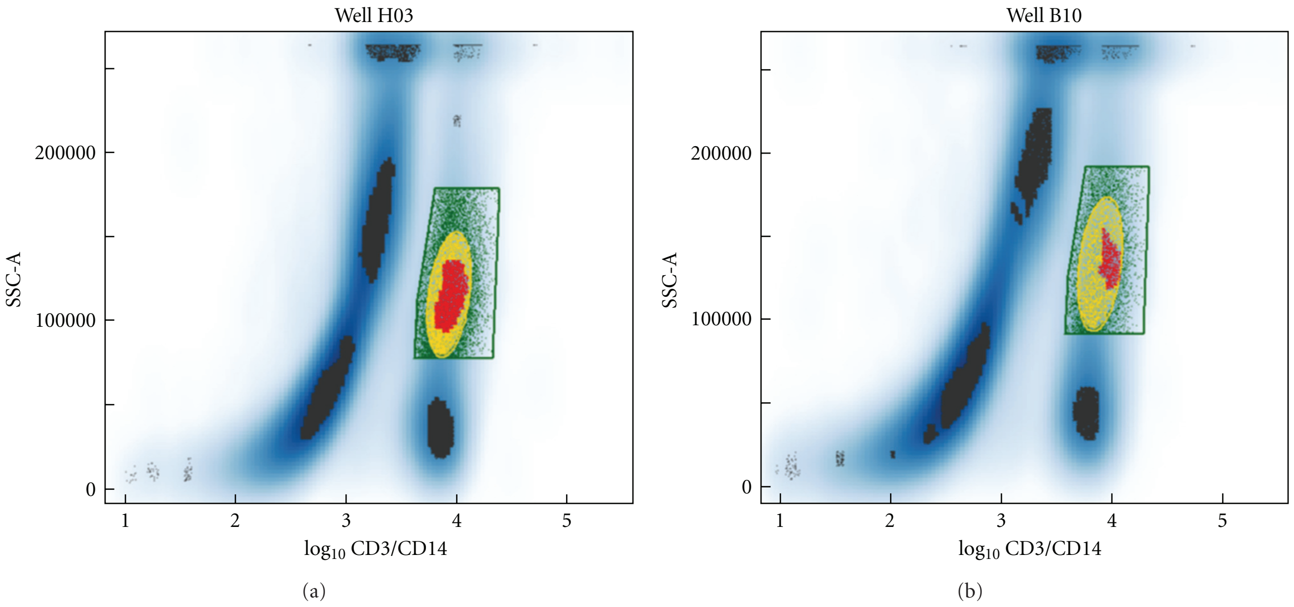


Figure 2: Gates for the monocyte population as produced by the three gating procedures applied to two of the 96 whole blood samples. The distribution of the cells is indicated by the blue shading with darker blue corresponding to regions containing higher numbers of cells. Regions where a probability density function fit to the data was calculated to have significant curvature are indicated in black, except where they lie within the manual gate and are colored red. Gating methods I, II, and III are shown in green, yellow, and red, respectively. The regions were colored in order of largest to smallest for visual display because the gates overlap with each other.

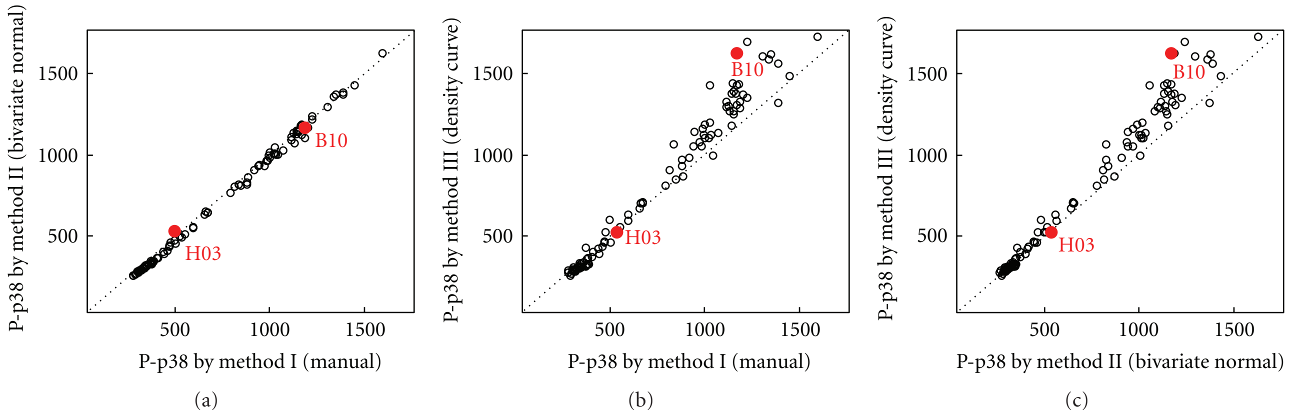


Figure 3: Comparison of the monocyte P-p38 mean fluorescent intensity as determined by the different gating methods for the 96 samples of whole blood. The apparent P-p38 response in any particular sample may be affected by the donor, the person running the assay, and the amount of TNF-α stimulation applied to the cells. The points corresponding to the two samples shown in Figure 2 are labeled in red.

methods is shown in Figure 3. It is clear that method I and method II agree very closely, while method III is moderately different from the first two methods. These simple graphs illustrate both a bias and variance between methods that should be taken into consideration in evaluating the strength of any conclusions drawn. Thus we have obtained an indication of the level of uncertainty due to gating strategy and can readily identify cases in which further investigation is warranted. Conversely, of course, we may decide one of the gating methodologies performs poorly and remove it from use for a particular application.

It is also important to look at the homogeneity of the cells within a gate. In some cases the monocyte population of the samples examined in this experiment was actually composed of two populations of cells as distinguished by different P-p38 expression. This difference was not apparent when examining the cells in the SSC versus CD3/CD14 domain. These two populations have offset centers in SSC versus CD3/CD14 space which causes a gradient in mean P-p38 expression across the manually drawn gate. Such mixed populations might be observed, for example, through use of 3 dimensional viewing tools. Alternatively one might color each point within a gate as a function of its distance in all measured parameters from the center of the gate, thereby providing a simple visual measure of cell homogeneity within a gate. Another approach is to divide the gate up into a number of subsets and compute the desired summary statistic for each subset. The variability across these subsets provides an assurance of the strength of the assumption of homogeneity within the gate. These types of visualizations and analyses are readily explored when the data is available in the R statistical programming environment.

The P-p38 heterogeneity of some of the samples illustrates the strengths and weaknesses of the three gating methods depending on the nature of the question being asked. If the goal is to further subdivide a population, it is important to be as inclusive as possible because the subtypes in other gating parameters may not be uniformly scattered across the parent gate. If the goal is to perform a dose-response assay by measuring, for example, the phosphorylation state of an internal signaling protein, a more restricted population such as the bivariate normal gate (method II) might be more appropriate. The curvature gradient approach (method III) is particularly sensitive to the distribution of cells within a region and might be valuable in assays where detecting slight changes in the population structure is important. The gating method and tuning parameters chosen should be chosen based on the question being addressed.

Finally, other aspects of the gating process may likewise be assessed for robustness. Automated and partially automated approaches have tuning parameters that are usually set to work well for test cases. The sensitivity of the results in these test cases can be helpful in judging how well the approach is going to work in a full study. To assess this sensitivity, an approach similar to that shown above can be used to systematically vary the tuning/controlling parameter and assess the variability in the results as a function of the controlling parameters.

4. Discussion and Conclusion

As the number of flow cytometry data sets grow in a study, it becomes increasingly difficult to explore "what-if" questions. It is common to uncover a behavior that can only be investigated by creating new gates or adjusting existing gates. Exploration and analysis of a data set can also reveal problems with an initial gating strategy that can be easily fixed computationally, but would be tedious to fix manually. Examples of this include the case in which manual gates are refined computationally and the case in which the robustness of gates (drawn manually or computationally) is assessed. We have also experienced cases in which a population that initially appeared to be of little importance turned out to be of substantial interest. In one case, the population had been poorly gated, and the events at the maximum possible intensity were included, but should not have been. Rather than re-gating manually, it was simple to adjust each gate computationally to exclude the boundary region.

The flowFlowJo package provides a set of methods for extracting and organizing information from FlowJo workspaces and the FCS files to which they refer. In its most basic application, it allows the user to retrieve all of the gates and spillover matrices for all of the FCS files described within one or more FlowJo workspaces. The gates are returned as flowCore compliant filter objects, and the spillover matrices are returned as numeric matrices. Additional functionality is gained by the ability of the user to effectively run all of the compensation and gating functions described by the workspace(s) and automatically retrieve all of the relevant summary statistics into a concise data structure. These data may also be easily combined with any metadata describing the nature or source of each sample and any experimental conditions to which they were subjected.

There has been limited involvement by the bioinformatics, statistical, and machine learning communities in the problems of flow cytometry [15]. Programmatic access to both raw data and gates in flow cytometry allows us to ask many questions about flow cytometry data that traditionally were tedious or effectively impossible. The ability to assess gate choice assumptions is expected to lead to better assessments of the quality of our methods. In some cases, more sophisticated approaches such as mixture modeling may be called for when seemingly uniform cell populations actually include two or more cell types. This is especially important when examining cell populations for which a subset of the cells with no known defining antibodies respond differently to stimuli than the rest of the cells in the population.

At the present time, flowFlowJo is known to work with FlowJo version 7.2.5 running on the Windows operating system. We expect FlowJo to continue to evolve and we intend to maintain flowFlowJo in such a way that it can handle the current FlowJo workspaces. A major change in the FlowJo workspace structure will be the transition to FlowJo 7.5 when the use of the Gating-ML standard is expected to replace the current XML format. The flowFlowJo package and supporting vignette and documentation is available from the Bioconductor web site (http://www.bioconductor.org/).

Acknowledgments

The authors thank Lauren Buchholz, Mark Dalphin, Adam Treister, and Florian Hahne for their assistance without which none of this work would have been possible. The authors declare no competing financial interests with FlowJo or any other commercial flow cytometry software packages. The authors are employees of Amgen.

References

[1] M. M. Hammer, N. Kotecha, J. M. Irish, G. P. Nolan, and P. O. Krutzik, "WebFlow: a software package for high-throughput analysis of flow cytometry data," *Assay and Drug Development Technologies*, vol. 7, no. 1, pp. 44–55, 2009.

[2] L. E. Bonilla, G. D. Means, K. A. Lee, and S. D. Patterson, "The evolution of tools for protein phosphorylation site analysis: from discovery to clinical application," *BioTechniques*, vol. 44, no. 5, pp. 671–679, 2008.

[3] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan, "Causal protein-signaling networks derived from multiparameter single-cell data," *Science*, vol. 308, no. 5721, pp. 523–529, 2005.

[4] N. LeMeur and F. Hahne, "Analyzing flow cytometry data with bioconductor," *Rnews*, vol. 6, no. 5, pp. 27–32, 2006.

[5] F. Hahne, N. LeMeur, R. R. Brinkman, et al., "flowCore: a bioconductor package for high throughput flow cytometry," *BMC Bioinformatics*, vol. 10, article 106, pp. 1–8, 2009.

[6] K. Lo, F. Hahne, R. R. Brinkman, and R. Gottardo, "flowClust: a bioconductor package for automated gating of flow cytometry data," *BMC Bioinformatics*, vol. 10, article 145, pp. 1–8, 2009.

[7] Data File Standards Committee of the International Society for Analytical Cytology (ISAC), "Data File Standard for Flow Cytometry, Version FCS3.0," http://www.isac-net.org/content/view/101/150.

[8] J. Spidlen, R. C. Gentleman, P. D. Haaland, et al., "Data standards for flow cytometry," *OMICS*, vol. 10, no. 2, pp. 209–214, 2006.

[9] R. Gentleman and D. T. Lang, "Statistical analyses and reproducible research," *Journal of Computational and Graphical Statistics*, vol. 16, no. 1, pp. 1–23, 2007.

[10] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: human-based character recognition via web security measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.

[11] K. Lo, R. R. Brinkman, and R. Gottardo, "Automated gating of flow cytometry data via robust model-based clustering," *Cytometry Part A*, vol. 73, no. 4, pp. 321–332, 2008.

[12] C. Chan, F. Feng, J. Ottinger, D. Foster, M. West, and T. B. Kepler, "Statistical mixture modeling for cell subtype identification in flow cytometry," *Cytometry Part A*, vol. 73, no. 8, pp. 693–701, 2008.

[13] F. Hahne, D. Arlt, M. Sauermann, et al., "Statistical methods and software for the analysis of highthroughput reverse genetic assays using flow cytometry readouts," *Genome Biology*, vol. 7, no. 8, article R77, pp. 578–588, 2006.

[14] T. Duong, A. Cowling, I. Koch, and M. P. Wand, "Feature significance for multivariate kernel density estimation," *Computational Statistics and Data Analysis*, vol. 52, no. 9, pp. 4225–4242, 2008.

[15] G. Lizard, "Flow cytometry analyses and bioinformatics: interest in new softwares to optimize novel technologies and to favor the emergence of innovative concepts in cell research," *Cytometry Part A*, vol. 71, no. 9, pp. 646–647, 2007.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2009, Article ID 193947, 11 pages
doi:10.1155/2009/193947

Research Article

FlowFP: A Bioconductor Package for Fingerprinting Flow Cytometric Data

Wade T. Rogers and Herbert A. Holyst

Department of Pathology and Laboratory Medicine, School of Medicine, University of Pennsylvania, 207 John Morgan Bldg., Philadelphia, PA 19104-6082, USA

Correspondence should be addressed to Wade T. Rogers, rogersw@mail.med.upenn.edu

Received 1 April 2009; Accepted 18 June 2009

Recommended by Raphael Gottardo

A new software package called flowFP for the analysis of flow cytometry data is introduced. The package, which is tightly integrated with other Bioconductor software for analysis of flow cytometry, provides tools to transform raw flow cytometry data into a form suitable for direct input into conventional statistical analysis and empirical modeling software tools. The approach of flowFP is to generate a description of the multivariate probability distribution function of flow cytometry data in the form of a "fingerprint." As such, it is independent of a presumptive functional form for the distribution, in contrast with model-based methods such as Gaussian Mixture Modeling. FlowFP is computationally efficient and able to handle extremely large flow cytometry data sets of arbitrary dimensionality. Algorithms and software implementation of the package are described. Use of the software is exemplified with applications to data quality control and to the automated classification of Acute Myeloid Leukemia.

1. Introduction

Flow cytometry (FC) produces multidimensional biological information at the level of the cellular compartment, and over very large numbers of cells. As such it is ideally suited to a wide variety of investigations for which cellular context and large sample observations are important. In recent years the technology of FC has undergone appreciable development [1, 2] with the introduction of digital signal processing electronics [3], multiple lasers, increasing numbers of fluorescence detectors, and robotic automation, both in sample preparation [4] and in instrumental data collection [5]. The recent development of new reagents [6] that enable increasing assay complexity has also been rapid and accelerating. Given the scope and pace of these developments, the bottleneck in many FC experiments has shifted from the wet laboratory to the computer laboratory; that is to say, data analysis [1].

FC data are typically analyzed using graphically driven approaches. Subsets of cells (events) are delineated usually in one- or two-dimensional histograms or "dotplots" in a procedure termed "gating." Gates of differing shapes including rectangular, circular, elliptical, or arbitrary polygonal contours may be specified. The gating process is frequently applied in a sequential fashion, with the numbers of events inside successive gates falling monotonically from step to step. Subsets determined via gating are typically then quantified with respect to their expression patterns in the dimensions of multiparameter space not utilized for gating, often by simply counting proportions of the subsets that are positive or negative for each of the markers of interest for that subset. Several commercially available software packages have been extensively optimized to support this kind of visually guided analysis workflow, for example, FlowJo (Treestar Inc, Ashland, OR), WinList (Verity Software House, Topsham, ME), and FCSExpress (De Novo Software, Los Angeles, CA).

Manual gating is a highly effective means of analysis of flow cytometry data, especially in cases where the application of expert judgment in the visual design of gating strategies may be able to isolate events of biological interest in the presence of confounding experimental (or biological) variations that will be difficult to account for automatically. Nevertheless, manual gating has three main drawbacks [7–9]. First, the choice of gates is often subjective, particularly in the not-unusual situation where the distribution is broad and smooth. This lack of objective criteria is problematic, especially when different samples may show different types of "excursions" from the average/normal case. Second, because gates are specified by manually drawing regions on a graph using a computer mouse, the process is very labor intensive and time consuming. Finally, because gating and regions of interest are determined by the data analyst based on his or her experience, there may be interesting and informative features that exist within the full ungated multivariate distribution of events but that nevertheless escape detection in this analysis paradigm.

A number of automated gating procedures have been developed with the aim of reducing tedium as well as increasing objectivity in the gating process. Notwithstanding this, a strong need still remains for computational tools that transform and represent multiparameter flow cytometric data in a form efficiently amenable to machine learning and data mining.

We have developed a software package called flowFP to address these limitations in conventional approaches to the analysis of FC data. The broad aim of the package is to directly transform raw FC list-mode data into a representation suitable for direct input to other statistical analysis and empirical modeling tools. Thus, it is useful to think of flowFP as an intermediate step between the acquisition of high-throughput FC data on the one hand, and empirical modeling, machine learning, and knowledge discovery on the other.

2. Materials and Methods

2.1. Algorithm Description. The software package described herein, flowFP, implements and integrates ideas put forth in [10, 11]. FlowFP utilizes the Probability Binning (PB) algorithm [10] to subdivide multivariate space into hyper-rectangular regions that contain nearly equal numbers of events. According to the vernacular of flow cytometry, the axes describing a multivariate space are referred to as "parameters." Here we will use the term "variable" so as to avoid confusion with the nomenclature of "parameter" as used in the statistics literature. Regions (bins) are determined by (a) finding the variable whose variance is highest, (b) dividing the population at the median of this variable which results in two bins, each with half of the events, and (c) repeating this process for each subset in turn. Thus, at the first level of binning the population is divided into two bins. At the second level, each of the two "parent" bins is divided into two "daughter" bins, and so forth. The final number of bins n is determined by the number of levels l of recursive subdivision, such that $n = 2^l$.

This binning procedure is typically carried out for a collection of samples (instances), called a "training set." The result of the process models the structure of the multivariate space occupied by the training set by the way it constructs bins of varying size and shape and is thus termed a "model" of the space (not to be confused with modeling approaches that fit data to a parameterized model or set of models). The model is then applied to another set of samples (which may or may not include instances from the training set). This operation results in a feature vector of event counts in each bin of the model for each instance in the set. These feature vectors are, in the context of a specific model, a unique description of the multivariate probability distribution function for each instance in the set of samples, and thus are aptly referred to as "fingerprints."

Although flowFP generates bins using the PB algorithm, the way it utilizes the resulting fingerprints is similar to the methods described in [11]. Each element of a fingerprint represents the number of events in a particular subregion of the model. Although it may not be known *a priori* which of these regions are informative with respect to an experimental question, it is possible to determine this by using appropriate statistical tests, along with corrections for multiple comparisons, to ascertain which regions (if any) are differentially populated in two or more groups of samples. If we regard the number of events in a bin as one of n features describing an instance, then the statistical determination of informative subregions is clearly seen to be a feature selection procedure.

Fingerprint features are useful in two distinct modes. First, all or a selected subset of features can be used in clustering or classification approaches to predict the class of an instance based on its similarity to groups of instances. Second, the events within selected, highly informative bins can be visualized within their broader multivariate context in order to interpret the output of the modeling process. This step is crucial in that it provides a means to develop new hypotheses for FC-derived biomarkers within the context of existing reagent panels.

2.2. Software Implementation. FlowFP is implemented in the open-source R Statistical Computing Environment [12] and is freely available as part of Bioconductor [13]. Within Bioconductor a framework has been created for handling FC data known as flowCore [14, 15]. FlowFP is one of a growing number of Bioconductor packages integrated within this framework and thus able to interoperate with other flowCore-compliant tools as well as with the full range of downstream statistical analysis and machine learning tools available in R. This integration enables flexible creation of powerful high-throughput analysis procedures for large FC data sets.

FlowFP uses the S4 object-oriented facility of R. Computationally intensive parts are written in the C programming language for efficiency. FlowFP is built around a set of three S4 classes, each with a constructor of the same name as the class name. In addition there are a number of methods for data accession, manipulation, and visualization.

2.2.1. FlowFPModel. FlowFPModel is the fundamental class for the flowFP package. The flowFPModel constructor takes a collection of one or more list-mode instances which are represented in the flowCore framework as a flowFrame (for a single instance) or a flowSet (for a collection of instances), respectively (henceforth we will refer to flowFrames and flowSets, the original list-mode data being implied). In addition to the required argument, flowFPModel has optional arguments that allow control over the number of levels of recursive subdivision and the set of variables to be considered in the binning process. By default all variables in the input flowSet are considered, but if this argument is provided, any variables not listed are ignored. The constructor emits an object of type flowFPModel, which encapsulates a complete representation of the binning process that is used later to construct fingerprints.

2.2.2. FlowFP. The flowFP constructor takes a flowFrame or a flowSet as its only required argument, and an optional flowFPModel. If no flowFPModel is supplied, flowFP computes a model (by calling flowFPModel internally). Regardless the source of the model, flowFP applies the model to each of the instances in its input. The resulting flowFP object extends the flowFPModel class and contains two additional important slots to store a matrix of counts and a list of tags. The counts matrix has dimensions $m \times n$, where m is the number of instances in the input flowSet (or one if a flowFrame is provided), and n is the number of features in the model. The tags slot is a list of m vectors, each of which has e elements, where e is the number of events in the corresponding frame in the input flowSet. The value for each element of the tag vector represents the bin number into which the corresponding event fell during the fingerprinting procedure. This is useful for visualization or gating based on fingerprints as will be illustrated below.

2.2.3. FlowFPPlex. The flowFPPlex is a container object which facilitates combining, processing, and visualizing large collections of flowFP objects which are all derived from the same set of instances. The flowFPPlex constructor takes a list of flowFP objects. The flowFPPlex manages the logical association of a set of flowFP descriptions. In particular, it extends the counts matrices of its members "horizontally" so as to create a unified representation of the entire collection of fingerprints. The main utility of the flowFPPlex is its support for creating a merged representation of a set of instances acquired using a multitube panel, with different flowFPModels for each tube in the panel.

2.2.4. Generic Functions. A number of other methods have been provided to facilitate interaction with and analysis of fingerprinting results. Chief among these are visualization methods that aid in the understanding and interpretation of fingerprinting results (see Figures S1–S3 in Supplementary Material available online at doi:10.1155/2009/193947). A few other accessor methods deserve special mention.

nRecursions(obj). This generic function returns the number of levels of recursive subdivision of its argument. FlowFP, flowFPPlex, and flowFPModel all implement the method. Furthermore, the flowFP class implements the "set" method. This enables the user to compute a model at some fairly high resolution, and then to derive fingerprints at that resolution or any lower resolution without recomputing the model. This is possible because fingerprinting is recursive, so that given any high-resolution model, all models of lower resolution can be derived from it.

counts(obj). This generic function returns a matrix of the number of events per instance and per bin. FlowFP and flowFPPlex classes implement this method, facilitating creation of fingerprint matrices suitable for processing by downstream methods outside of the flowFP package. The method has an optional argument "transformation" that can take on values "raw" (returns the actual event counts for each bin), "normalize" (normalizes by dividing raw counts by the expected number of events), or "log2norm" (like normalize except that it further takes the $\log_2$ of the result).

sampleNames(obj) and sampleClasses(obj). These generic functions set or get sample identifiers for objects of class flowFP or flowFPPlex. By default, for flowFPs, sample names are derived from the flowSet. However they can be overridden by the set method, providing flexibility to handle cases where the sample names in a flowSet are not appropriate. When adding fingerprints to a flowFPPlex, sample names (and if present, sample classes) are compared, and the join operation is not permitted unless names and classes among all fingerprints in the flowFPPlex are identical.

parameters(obj). This generic function returns the light scatter and/or fluorescence variables involved in binning, either for a flowFPModel or a flowFP. The function is able to report both the variables that were considered for binning as well as those that actually participating (if the global variance of a variable is small enough it may never be selected for division).

tags(fp). This generic function returns the tags slot of a flowFP object, described in Section 2.2.2. This is useful for visualization and gating operations.

binBoundary(obj). This generic function reports a list of multivariate rectangles corresponding to the limits of the bins. FlowFP and flowFPModel classes both implement this method. This information is also useful for visualization and gating operations.

2.3. Data and Characteristics. Deidentified flow cytometric data from peripheral blood or bone marrow aspirate samples were provided by Clarient, Inc. (Aliso Viejo, CA) along with primary diagnoses by experienced hematopathologists. After application of QC filters including that described in Section 3.1.1 the data set included 42 cases diagnosed as Acute Myeloid Leukemia (AML) and 309 cases that were determined to be immunophenotypically normal. For the purposes of this study physician diagnosis was regarded as the ground truth.

Data were collected over a one-year period, using the panel described in Table 1. Briefly, samples were lysed with ammonium chloride, then washed with PBS, centrifuged and resuspended. Blocking was accomplished by incubating with RPMI-1640 supplemented with 10% rabbit serum for 30 minutes at 37°C. Cells were then pelleted, resuspended in RPMI-1640, and adjusted to between 4–8 × 10^6 cells/mL. Antibody staining was accomplished by incubating in the dark at room temperature for 15 ± 5 minutes 100 μL of the adjusted cell suspension with 40 μL of pretitrated antibody cocktail per tube. For the viability tube, 10 μL of 7AAD

Table 1: Reagent panel used for immunophenotyping of leukemia/lymphoma samples.

Tube	FL1	FL2	FL3	FL4	FL5
	P3S	P4S	P5S	P6S	P7S
1	IgG1-FITC	IgG1-PE	CD45-ECD	IgG1-PC5	IgG1-PC7
2	(s)Kappa-FITC	(s)Lambda-PE	CD45-ECD	CD19-PC5	CD20-PC7
3	CD7-FITC	CD4-PE	CD45-ECD	CD8-PC5	CD2-PC7
4	CD15-FITC	CD13-PE	CD45-ECD	CD16-PC5	CD56-PC7
5	CD14-FITC	CD11c-PE	CD45-ECD	CD64-PC5	CD33-PC7
6	HLA DR-FITC	CD117-PE	CD45-ECD	CD34-PC5	CD38-PC7
7	CD5-FITC	CD19-PE	CD45-ECD	CD3-PC5	CD10-PC7
8	FL1-Log	FL2-Log	FL3-Log	FL4-Log	FL5 Log

was added in place of the antibody cocktail. After staining each tube was washed with 3 mL PBS, vortexed, pelleted, and resuspended in 500 μL of PBS prior to running on the flow cytometer. Five-color immunofluorescence along with forward and side scatter data were collected on two FC-500 cytometers (Beckman Coulter, Miami, FL). Data were collected for 3×10^4 events for each tube.

3. Results

3.1. Gating Quality Control

3.1.1. Tube Data. FlowFP was used to assess the consistency of event distributions in variables common to a multiple-tube panel. Using the panel described in Table 1, note that CD45 is common to all tubes except the viability tube. Frequently [16–22], the distribution of events in the Side Scatter versus CD45 projection (referenced as parameters 2 and 5 in the code below) from a single tube is used to gate an entire collection of tubes in order to save time. If the CD45 versus SSC distribution differs among the tubes, errors due to incorrect subsetting will occur, but may not be readily apparent without careful study of the gating plots.

Using flowFP, in order to rapidly detect consistency of CD45 versus SSC distributions without the need to look at dotplots, we (1) create a flowSet comprising tubes 1–7 of a sample, (2) create a model, using the common variables CD45 and SSC, from the flowSet, (3) create fingerprints of the same samples with respect to this model, and (4) display the result. The R commands to accomplish this using flowFP are as shown in Algorithm 1 (Code Fragment 1).

Figure 1(a) shows the resulting plot. Each tube is represented by one of the colored plots, with the CD45 versus SSC fingerprint shown as a line. The standard deviation of the fingerprint values around their mean is shown for each tube to provide a quantitative measure of the degree to which a tube deviates from the norm of all tubes combined. The same value is mapped to colors, shown in the color legend above the plots, to provide a quick visual representation of the consistency of the distributions. For comparison, Figure 1(b) shows a similar result for a sample that displayed poor CD45 versus SSC consistency. Note that Tube 5 in that sample differed markedly from the other tubes in the panel, as did Tube 4, but to a lesser extent.

3.1.2. 96-Well Plate Data. High-throughput FC data are flexibly accommodated in the FlowFP package. For data derived from 96-well plates, a plot method of type “plate” can be used to display a qc-style plot in a layout that reflects the structure of the plate. Figure 2 shows such a result. Data were obtained [23, 24] in which SSC, CD3, and CD4 (parameters 2, 5, and 7) were used to gate the entire plate of data. The R commands shown in Algorithm 2 (Code Fragment 2) were used to produce the plot in Figure 2.

Note that in this case we illustrate the use of an implicit model by omitting the model from the flowFP constructor. The utility of such a rapid and straightforward quality assurance tool is most apparent in the case of this sort of high-throughput data.

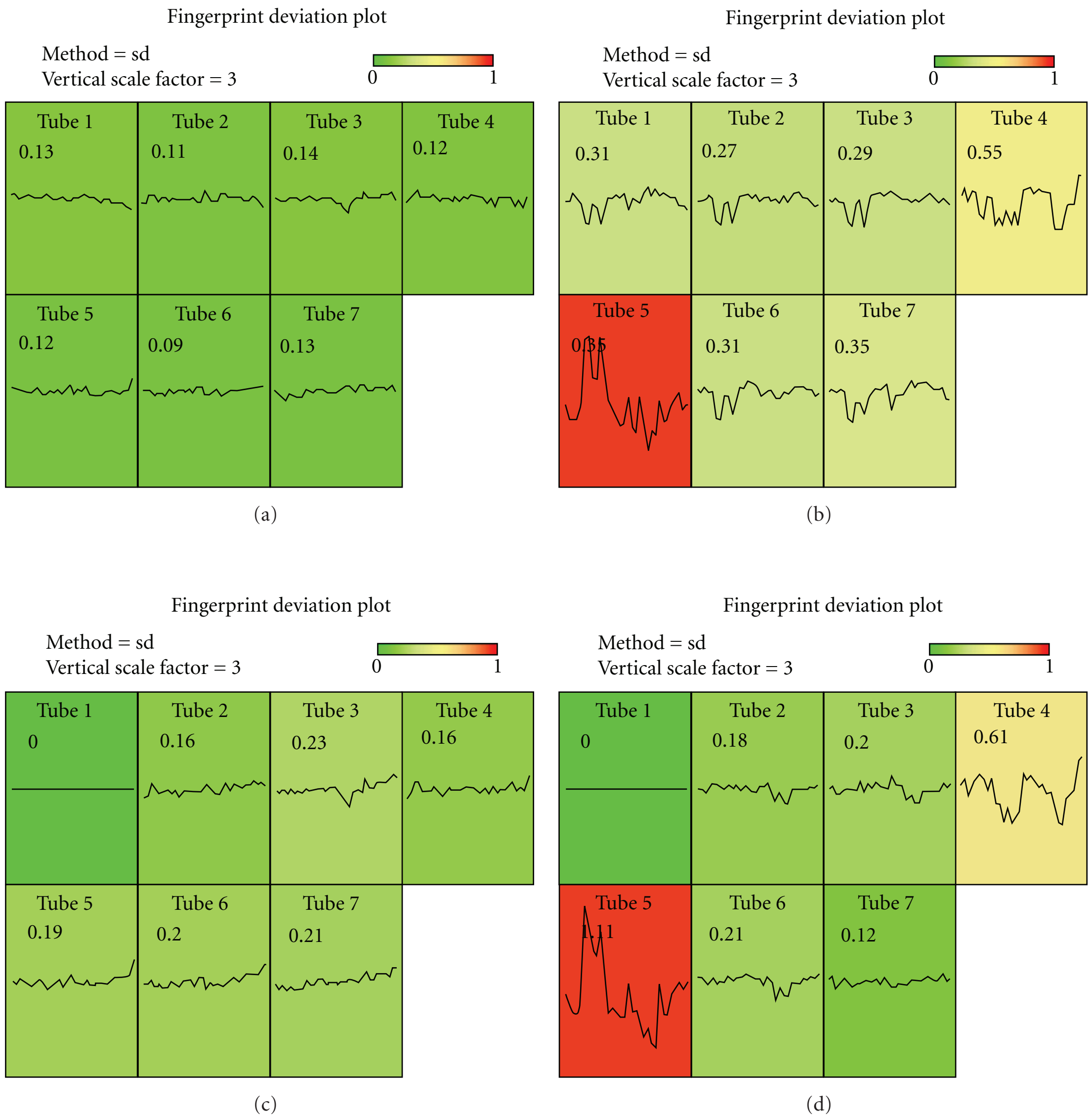
3.2. Automated Classification of Acute Myeloid Leukemia. We now turn to the application of flowFP to support a machine learning workflow. The aim here is to illustrate the utility of fingerprint-based approaches in general, and flowFP in particular, by automatically categorizing samples into one of two *a priori* known classes, AML or Normal. The dataset described in Section 2.3 was used. Tube 1 (isotype control) and Tube 8 (viability) were ignored for the purpose of this analysis, leaving 6 tubes, numbered 2–7.

We divided the samples randomly into a balanced training set comprising 21 of 42 AML cases and 21 of 309 Normal cases. We elected to balance the training set so as not to bias the classifier towards the more heavily represented Normal case. The remaining 21 AML cases and 288 Normal cases were assigned to the test set. Modeling and fingerprinting were done on a per-tube basis. Models were computed from training data only, in order to avoid biasing the prediction of the test set. We also employed a “differential modeling” procedure by creating two separate models, one for the AML training instances and one for the Normal training instances. Then, fingerprints from each tube and for each model were computed and aggregated into a flowFPPlex for further analysis. Fingerprinting was performed on all variables. The R code fragment implementing this procedure is shown in Algorithm 3 (Code Fragment 3).

Models were computed at a resolution level $l = 11$, producing $n = 2048$ bins. This resolution was determined using the default automatic setting of flowFPModel which implements the heuristic that the typical (median) number

```
> fs <- read.flowSet (path="lo_gate_dev", transformation=FALSE)
> mod <- flowFPModel (fs, parameters=c(2,5))
> fp <- flowFP (fs, mod)
> plot (fp, type="qc")
```

Algorithm 1: (Code Fragment 1).



(a)

(b)

(c)

(d)

Figure 1: FlowFP plot method to display gating data consistency. Fingerprints were computed using CD45 and SSC which are common variables in all tubes. Fingerprint similarity is indicated by color and in the similarity metric shown in each panel. The color wedge shows mapping of colors to values of the similarity metric (values above the maximum indicated on the wedge all map to red). The x-axis for each subplot is fingerprint index, and the y-axis is the $\log_2$ transformed fingerprint value plotted with zero at the center and scaled to ± "vertical scale factor" (in this case 3.0). (a) Sample FI05_000942, an example of a sample with good gating consistency. (b) Sample FI05_000599, an example of a sample with poor gating consistency. (c) and (d) as in (a) and (b), except that models were computed from Tube 1 only, rather than the aggregate of Tubes 1–7 for each sample. Note that the fingerprint for Tube 1 in both cases has zero deviation, as expected. Note also the qualitative similarity between (a) and (c) and between (b) and (d).

```
> fs <- read.flowSet (path="96_well", transformation=F)
> fp <- flowFP (fs, parameters=c(2,5,7))
> plot (fp, type="plate")
```

Algorithm 2: (Code Fragment 2).

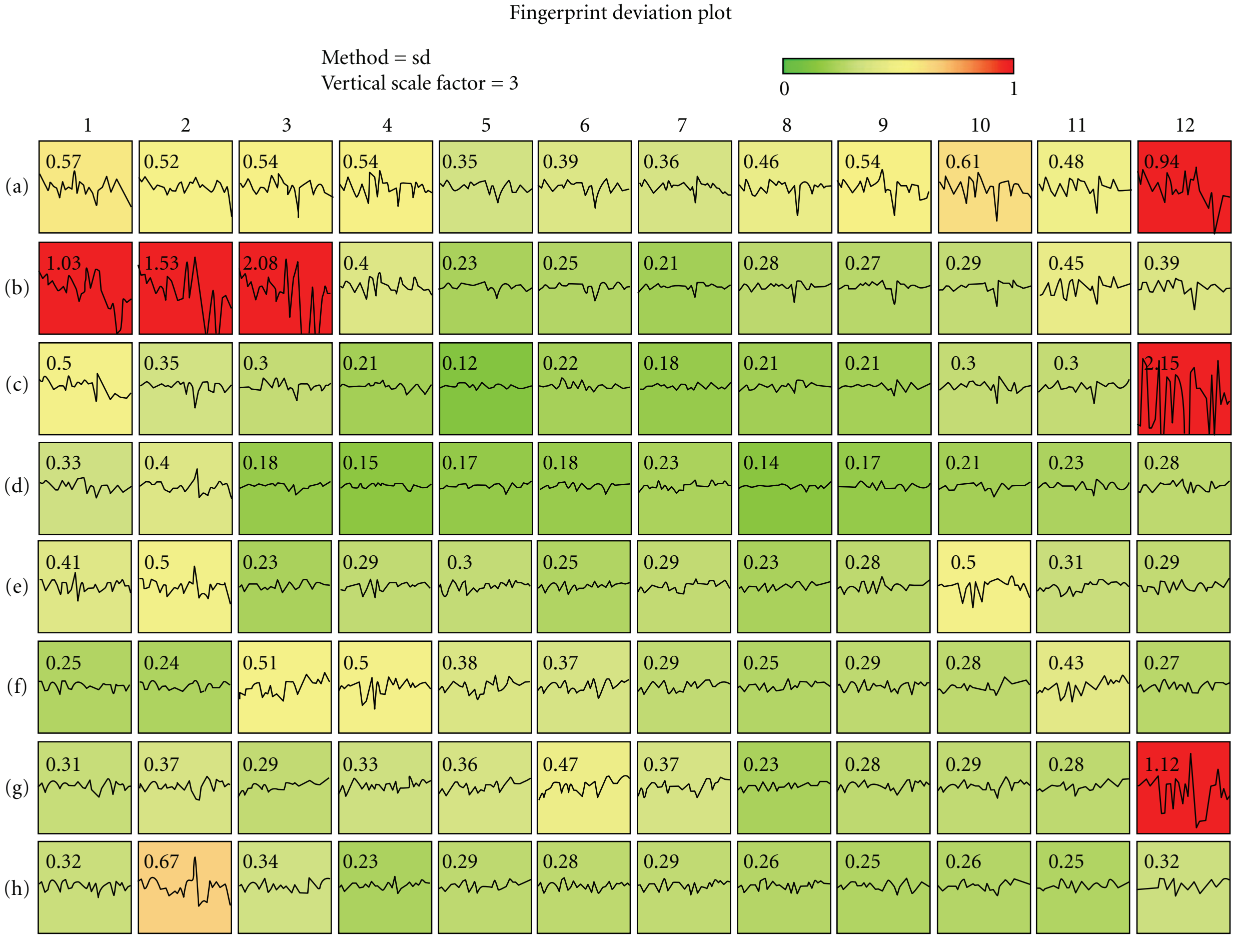


Figure 2: QC plot method for high-throughput data. Data were fingerprinted on variables common to all wells in a 96-well plate. The display maps into colors the degree to which gating data conform to the plate-wide norm.

of events in each instance of the training set is binned such that the number of events per bin is not less than 8. The resulting flowFPPlex therefore had 6 tubes × 2 models × 2048 bins = 24 576 features.

We extracted feature values from the flowFPPlex using the accessor function `counts(plex, transformation="log2norm")` which performs a logarithmic transformation on the normalized counts matrix.

Using only the instances in the training set, we performed a Mann-Whitney test on each feature independently (there are many methods of feature selection, a discussion of which is beyond the scope of this report). We selected those features which had a 99.9% likelihood of being differentially distributed between the two classes, after performing the Benjamini-Hochberg correction for multiple comparisons [25, 26]. This led to the selection of 1681 informative features out the original 24 576 features. Using the reduced feature set we trained a Support Vector Machine (SVM) classifier [27, 28] using a radial basis function kernel. We then blindly predicted the class of the test set using this classifier by assigning the predicted class probabilities into three equal ranges. The results are shown in Figure 3. Sensitivity and specificity are 90.5% (19/21) and 99.3% (278/288), respectively, with 9.5% (2) of AML instances and 2.8% (8) of Normal instances falling into the Uncertain group. No cross validation was performed here for clarity and brevity of presentation. For a better assessment of model performance this would be required. Interestingly, repeating the analysis

```
trainSets <- list(aml=train_aml, norm=train_norm)
plex <- flowFPPlex ()                          # create an empty plex
for (tube in 2:7) {                            # loop over tubes 2–7
      fs <- read_tubes (tube)                  # create a flowSet
      for (trainSet in trainSets) {            # differential modeling
            mod <- flowFPModel (fs[trainSet])  # training set only
            fp <- flowFP (fs, mod)             # create fingerprints
            plex <- append (plex, fp)          # add fingerprints to plex
}
}
```

Algorithm 3: (Code Fragment 3).

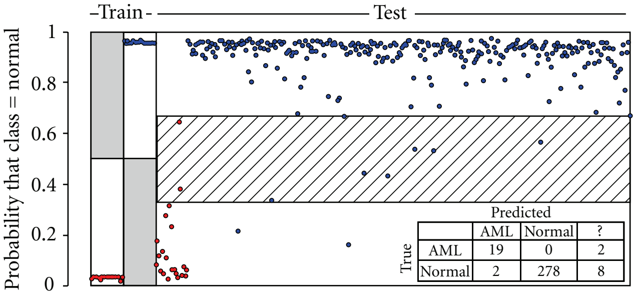


Figure 3: Support Vector Machine classification of AML versus Normal. The classifier was trained with 21 AML and 21 Normal Instances (left-most two regions). The classifier was then used to blindly predict class probabilities for the test set of 21 AML and 288 Normal instances (the right-most region). Ground-truth class assignments are indicated by color, red for AML and blue for Normal. The probability range 0-1 was divided into three equal regions. Instances falling into the lower third were classified as AML, in the upper third as Normal, and in the middle as Uncertain "?".

without the "differential modeling" method described above (i.e., using AML and Normal combined training instances to compute the models for each tube) resulted in a similar result, but with a slightly poorer sensitivity of 85.7% (data not shown).

The time required to compute the fingerprints was 1.8 seconds per sample, requiring 5.2 GB of memory on a machine running the Linux 2.6 SMP 64-bit kernel with a 2.83 GHz processor. Recall that this represents, for each sample, the construction of 2 fingerprints for each of 6 tubes, each of which has 3×10^4 events. Compared with mixture modeling approaches that are used for analysis of FC data (e.g., [8, 29]) flowFP is a computationally inexpensive method of analysis of FC data.

Figure 4(d) shows the distribution of informative features selected as described above with respect to tube number. Tubes 7 and 4 appear to be the most informative for distinguishing AML from Normal. Figures 4(a)–4(c) display the informative subset of features (bins) that fell in Tube 7 and which had higher likelihood, on average, in the AML group compared with the Normal group. Informative features characteristic of AML can be described as low-intermediate SSC, CD45 dim, and negative for CD3, CD19, and CD10. The CD45 versus SSC distribution of the informative bins corresponds to a region containing blasts and monocytes.

A more comprehensive although less detailed picture of information distribution in the panel is illustrated in Figure 5. This parallel coordinate view enables the appreciation of expression patterns across the entire panel of tubes. Notice that the AML pattern in Tube 7 displayed in Figure 5 indicates the same CD45(dim), CD3(−), CD10(−) blast phenotype shown in Figure 4. In Tube 4 the phenotype of AML-informative bins is consistent with blasts expressing CD15(dim to −), CD13(dim to +), CD16(−), CD56(−) (see also Figure S4 in Supplementary Material). Separation of the bundles of trajectories corresponding to AML and Normal events is the widest in Tubes 4, 6, and 7, consistent with the distribution of information across the tubes shown in Figure 4(a). By contrast, Tube 5 has intertwined bundles, apparently in keeping with the fact that Tube 5 held the fewest informative fingerprinting features.

4. Discussion

With recent technological advances, FC is now capable of operating as a true high-throughput technique. A key enabling requirement however is the need to automate data analysis for speed, much as automation in sample preparation and data acquisition have accelerated the rate of generation of data and thereby enabled high-throughput FC. This requirement inevitably drives movement away from human-drawn, visually-based gating which is the single most significant obstacle preventing a true high-throughput FC workflow.

We have shown that fingerprint-based analysis of FC data represents an effective bridge between large amounts of FC data and the world of machine learning and knowledge discovery techniques. It effectively captures informative features of a multivariate probability distribution function and does so in a computationally efficient way. As such it represents one of the tools that may help to bring FC into a new era of application to problems previously not feasible due to limitations in data analysis techniques.

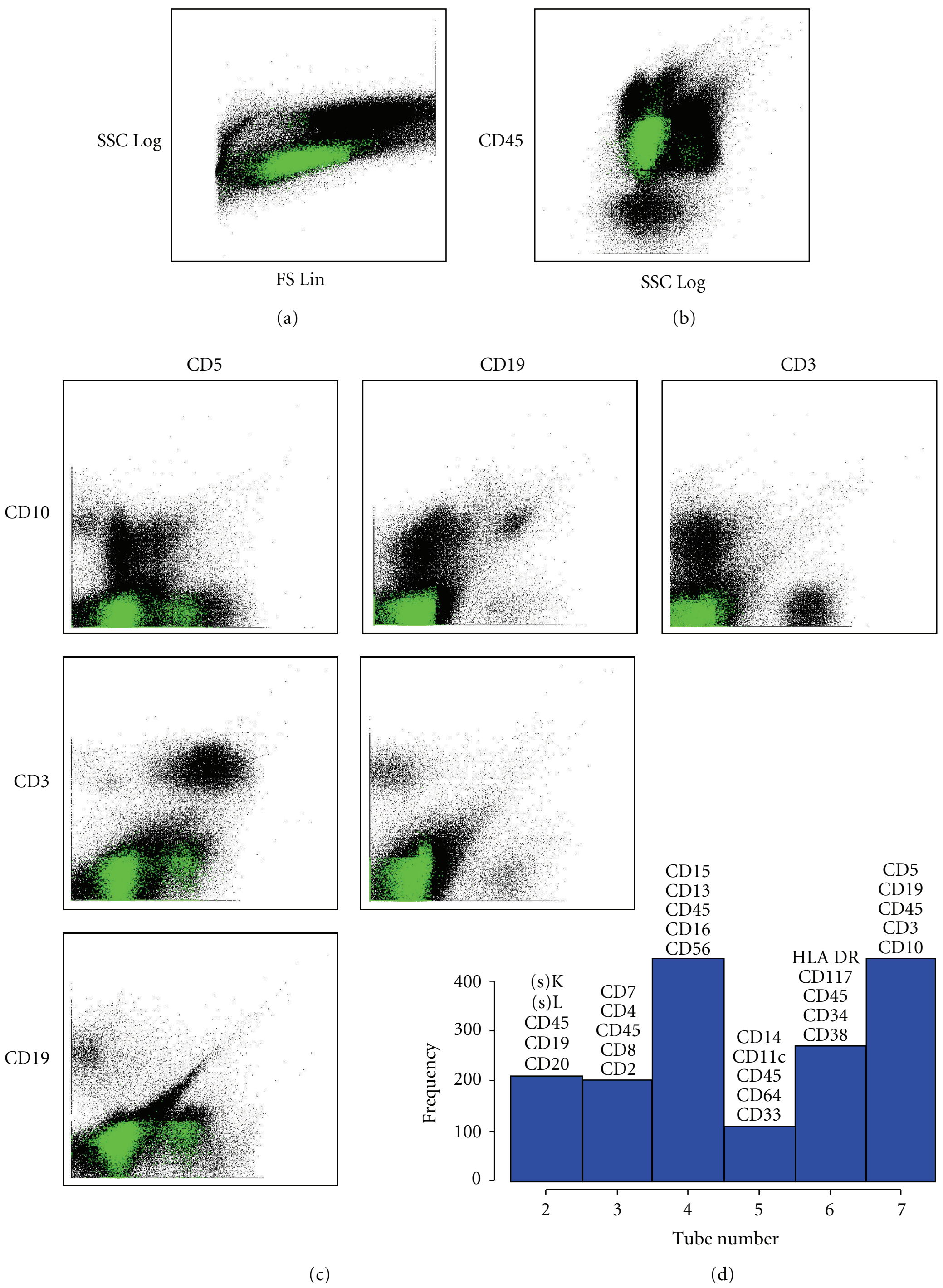


Figure 4: Visualization of informative features. (a)–(c) dotplots for Tube 7. Black dots are aggregated data from 5 AML and 5 Normal instances. Colored dots indicate events in informative bins with higher probability density in AML compared with Normal. (a) Side Scatter versus Forward Scatter. (b) CD45 versus Side Scatter. (c) Pairwise dotplots of fluorescence's CD5, CD19, CD3, and CD10. (d) Histogram of the frequency with which informative features occur in Tubes 2–7.

It is important to note that fingerprinting of FC data is not without limitations. First, we note that fingerprinting approaches are sensitive to differences in multivariate probability distributions no matter their origin. Thus, instrumental, reagent or other systematic variations may cause spurious signals as large or larger than true biological effects. For this reason it is important to measure and control for these effects [1]. In fact, fingerprinting itself can be used to assess and to help control for systematic effects, as was illustrated in Section 3.1.

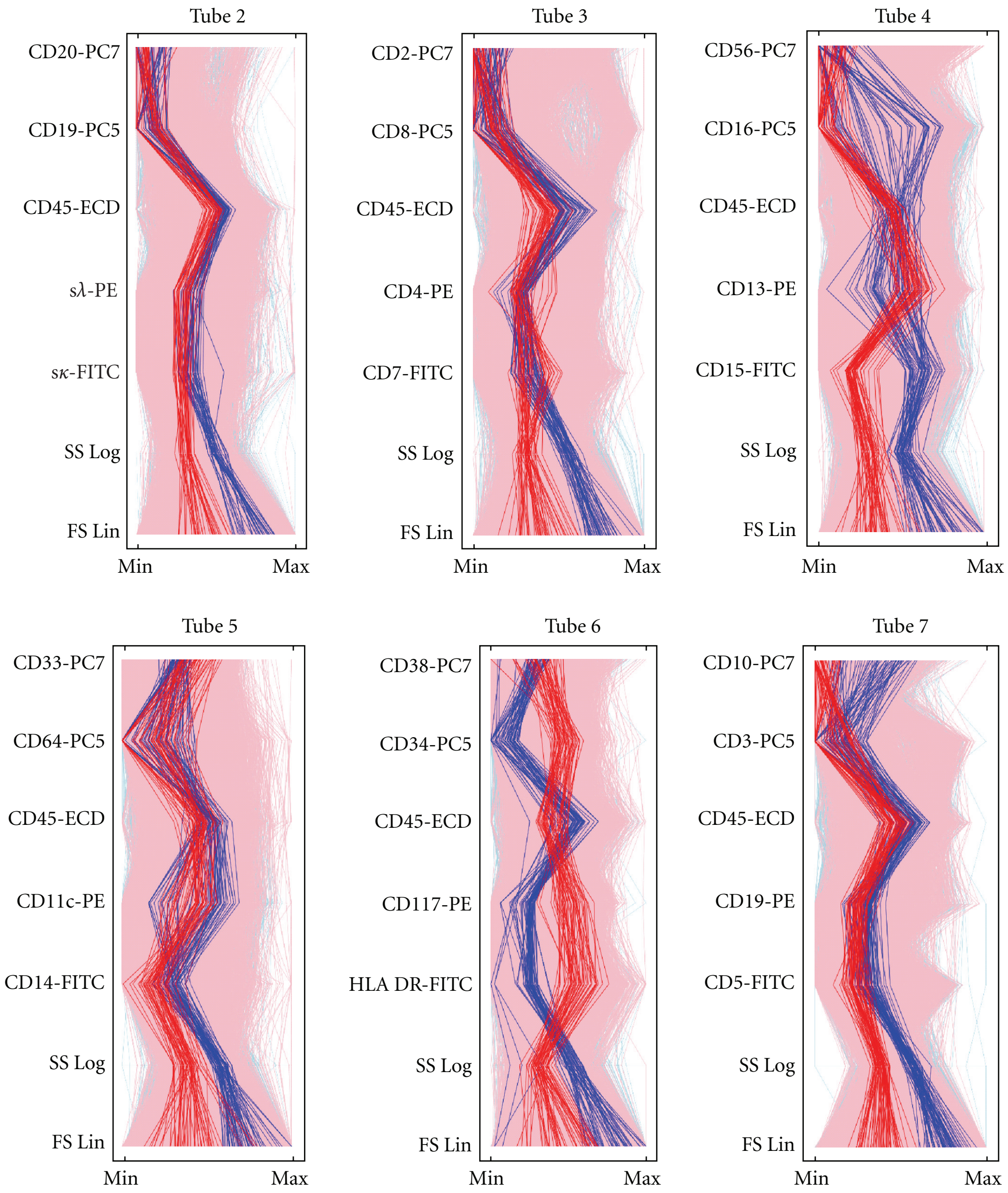


Figure 5: Parallel coordinate view of informative fingerprint features. The expression pattern of an individual event is shown as a vertical trajectory. Events are chosen from informative bins selected as described in the text. Events in bins with excess median probability among AML (Normal) instances are shown in red (blue). The numbers of AML and Normal trajectories are balanced to avoid visual bias.

Second, because fingerprinting is, in essence, the creation of a multivariate histogram, it responds to factors that might artificially emphasize certain bins in preference to others. In particular, events may pile up on either the zero or full-scale axis for one or more variables. This situation frequently results from values that would be negative due to compensation or background subtraction (causing pileup on the zero axis) or at the other end of the scale, values that exceed the dynamic range of the signal detection apparatus causing pileup at full scale. At either end this results falsely in an apparent high density of events. Fingerprinting bins are thus “attracted” to these locations, causing a distortion in the proper characterization of the true multivariate probability distribution function. One might be tempted to simply remove these values. However this is problematic since they can be very important. For example, values piling up at full scale are the brightest of all. A better solution is to adjust detector gains to minimize or eliminate full-scale pileup, to use high-dynamic-range detectors and electronics and to use modified data transformations such as the biexponential transform to smoothly distribute values at or below zero.

Just as scaling and transformation of data are important for visualization of multivariable distributions [30–32], so they are also important for fingerprinting. Data acquired using linear amplifiers such as exist in some modern instruments, or data that have been “linearized” from instruments with logarithmic amplifiers, tend to be heavily skewed to the left, since in most cases data distributions are quasi-log-normally distributed. Bins determined from such data thus have extreme variations in size. A good rule of thumb is to use a data transformation that produces the most spread-out distribution, which also is often the transformation most effective for clear visualization of the distribution. For example, Forward Scatter data are almost always displayed on a linear scale, whereas fluorescence data are usually displayed on a logarithmic or biexponential scale. For a good review of scaling and transformation of flow cytometric data, the reader is referred to [32].

A key limitation for fingerprinting approaches, including flowFP, relates to the number of events available for analysis. Since the objective of probability binning is to find bins containing equal numbers of events, it follows that once the number of bins is on the order of the number of events in an instance, the expected number of events per bin will be of order unity. In this case differences in bin counts will not be statistically significant. On the other hand, if the dimensionality of the data set is high, the average number of times any variable will be divided in the binning process will be small. For example, in a dataset with 18 variables, if we demand at least, say, 10 events per bin for statistical accuracy, about 2.6×10^6 events would be required in order that each variable is divided on average into at least two bins. Thus, the spatial resolution of binning is limited by the number of events collected, and as the number of variables increases, the number of events needed to maintain resolution increases geometrically.

FlowFP has been peer reviewed and accepted for inclusion in the next release of Bioconductor scheduled for October 2009. Prior to that date the development version may be downloaded from http://www.bioconductor.org/. The package is currently available for all architectures supported by Bioconductor. In addition to the functionality illustrated here, the authors plan to improve some of the visualization methods, specifically to enable better use of color, for example to represent statistical significance of bins. One of the advantages of integration with other flow cytometry Bioconductor packages is the ease of comparing and combining analysis methodologies. For example, it will be of interest to compare the performance of fingerprinting with other methods such as clustering and mixture modeling (flowClust). By the same token, such methods might be used in concert. For example, it is possible that clustering could be used to define major cell categories (e.g., B cells, T cells, granulocytes, etc.), within which fingerprinting may efficiently parse subsets correlated with function.

In summary, flowFP provides the flow cytometry community with a new tool that transforms FC data such that a wide range of other data analysis algorithms may be brought to bear. It creates a representation of FC data that preserves information embedded in the multivariate probability distribution function while at the same time presenting the information in a way that can be utilized easily by other software tools. Because it is tightly integrated in Bioconductor with several other FC-related packages and also exists in the broader R statistical computing environment, flowFP can interoperate with a very wide range of open-source analysis techniques. This power and flexibility enables a broad range of new computational analysis approaches that have potential in two distinct areas. First, it will facilitate the retrospective mining of FC data, seeking novel biomarkers that may be lurking in existing data. Second, it breaks the data analysis bottleneck that has up until now limited the full exploitation of FC in clinical applications.

Acknowledgments

The authors wish to express their deep gratitude to Jonni Moore, Hank Pletcher, Andrew Bantly, Rich Schretzenmair, and all of the people of the University of Pennsylvania Flow Cytometry Resource for providing intellectual fuel and many reasons to create flowFP. They thank Florian Hahne, Nolwenn Le Meur, and Ryan Brinkman for advice and assistance in programming in R and integration with flowCore. They thank Bruce Bagwell for stimulating discussions regarding applications of fingerprinting of FC data. They are most especially grateful to Clarient, Inc. for generously making available the data sets used to illustrate the utility of flowFP. Finally, they thank the University of Pennsylvania PathBioresource and the Department of Pathology and Laboratory Medicine for financial support.

References

[1] P. K. Chattopadhyay, C.-M. Hogerkorp, and M. Roederer, "A chromatic explosion: the development and future of multiparameter flow cytometry," *Immunology*, vol. 125, no. 4, pp. 441–449, 2008.

[2] J. P. McCoy Jr., "Basic principles of flow cytometry," *Hematology/Oncology Clinics of North America*, vol. 16, no. 2, pp. 229–243, 2002.

[3] S. Murthi, S. Sankaranarayanan, B. Xia, G. M. Lambert, J. J. Rodriguez, and D. W. Galbraith, "Performance analysis of a dual-buffer architecture for digital flow cytometry," *Cytometry Part A*, vol. 66, no. 2, pp. 109–118, 2005.

[4] A. S. Kelliher, D. W. Parent, D. C. Anderson, et al., "Novel use of the BD FAGS^{TM} SPA to automate custom monoclonal antibody panel preparations for immunophenotyping," *Cytometry Part B*, vol. 66, no. 1, pp. 40–45, 2005.

[5] I. V. Gates, Y. Zhang, C. Shambaugh, et al., "Quantitative measurement of varicella-zoster virus infection by semiautomated flow cytometry," *Applied and Environmental Microbiology*, vol. 75, no. 7, pp. 2027–2036, 2009.

[6] P. K. Chattopadhyay, D. A. Price, T. F. Harper, et al., "Quantum dot semiconductor nanocrystals for immunophenotyping by polychromatic flow cytometry," *Nature Medicine*, vol. 12, no. 8, pp. 972–977, 2006.

[7] R. Achuthanandam, J. Quinn, R. J. Capocasale, P. J. Bugelski, L. Hrebien, and M. Kam, "Sequential univariate gating approach to study the effects of erythropoietin in murine bone marrow," *Cytometry Part A*, vol. 73, no. 8, pp. 702–714, 2008.

[8] C. Chan, F. Feng, J. Ottinger, D. Foster, M. West, and T. B. Kepler, "Statistical mixture modeling for cell subtype identification in flow cytometry," *Cytometry Part A*, vol. 73, no. 8, pp. 693–701, 2008.

[9] P. Lloyd-Evans, A. R. Guest, E. B. Austin, and M. L. Scott, "Use of a phycoerythrin-conjugated anti-glycophorin A monoclonal antibody as a double label to improve the accuracy of FMH quantification by flow cytometry," *Transfusion Medicine*, vol. 9, no. 2, pp. 155–160, 1999.

[10] M. Roederer, W. Moore, A. Treister, R. R. Hardy, and L. A. Herzenberg, "Probability binning comparison: a metric for quantitating multivariate distribution differences," *Cytometry*, vol. 45, no. 1, pp. 47–55, 2001.

[11] W. T. Rogers, A. R. Moser, H. A. Holyst, et al., "Cytometric fingerprinting: quantitative characterization of multivariate distributions," *Cytometry Part A*, vol. 73, no. 5, pp. 430–441, 2008.

[12] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008.

[13] R. C. Gentleman, V. J. Carey, D. M. Bates, et al., "Bioconductor: open software development for computational biology and bioinformatics," *Genome Biology*, vol. 5, no. 10, p. R80, 2004.

[14] B. Ellis, P. Haaland, F. Hahne, et al., "flowCore: basic structures for flow cytometry data," 2009, http://bioconductor.org/packages/2.3/bioc/html/flowCore.html.

[15] F. Hahne, N. LeMeur, R. R. Brinkman, et al., "flowCore: a Bioconductor package for high throughput flow cytometry," *BMC Bioinformatics*, vol. 10, article 106, 2009.

[16] M. J. Borowitz, K. L. Guenther, K. E. Shults, and G. T. Stelzer, "Immunophenotyping of acute leukemia by flow cytometric analysis: use of CD45 and right-angle light scatter to gate on leukemic blasts in three-color analysis," *American Journal of Clinical Pathology*, vol. 100, no. 5, pp. 534–540, 1993.

[17] K. Nishikawa, T. Miyasaki, N. Tsukaguchi, Y. Noma, K. Nakagawa, and N. Narita, "CD45 gating of acute leukemia," *Rinsho Byori*, vol. 44, no. 6, pp. 548–553, 1996.

[18] F. Lacombe, F. Durrieu, A. Briais, et al., "Flow cytometry CD45 gating for immunophenotyping of acute myeloid leukemia," *Leukemia*, vol. 11, no. 11, pp. 1878–1886, 1997.

[19] W. Cui, W. Ma, and Q. Lin, "CD45-gating for flow cytometric immunophenotyping of leukemia," *Zhongguo Yi Xue Ke Xue Yuan Xue Bao*, vol. 22, no. 2, pp. 199–203, 2000.

[20] R. Gelman and C. Wilkening, "Analyses of quality assessment studies using CD45 for gating lymphocytes for $CD3^+4^+\%$," *Cytometry Part B*, vol. 42, no. 1, pp. 1–4, 2000.

[21] R. J. Lock, P. F. Virgo, and R. S. Evely, "Pitfalls of CD45 gating strategies in leukaemia immunophenotyping," *Clinical and Laboratory Haematology*, vol. 25, no. 1, p. 67, 2003.

[22] S. H. Maljaei, I. Asvadi-E-Kermani, J. Eivazi-E-Ziaei, A. Nikanfar, and J. Vaez, "Usefulness of CD45 density in the diagnosis of B-cell chronic lymphoproliferative disorders," *Indian Journal of Medical Sciences*, vol. 59, no. 5, pp. 187–194, 2005.

[23] M. Inokuma, C. dela Rosa, C. Schmitt, et al., "Functional T cell responses to tumor antigens in breast cancer patients have a distinct phenotype and cytokine signature," *Journal of Immunology*, vol. 179, no. 4, pp. 2627–2633, 2007.

[24] M. Inokuma, C. dela Rosa, C. Schmitt, et al., 96-well plate data deposited with Flow Informatics and Computational Cytometry Society website, 2008, http://www.ficcs.org/software.html#Data_Files.

[25] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate-a practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society Series B*, vol. 57, no. 1, pp. 289–300, 1995.

[26] K. I. Kim and M. A. van de Wiel, "Effects of dependence in high-dimensional multiple testing problems," *BMC Bioinformatics*, vol. 9, article 114, 2008.

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[28] E. Dimitriadou, K. Hornik, F. Leisch, et al., "e1071: Misc Functions of the Department of Statistics," 2008, http://cran.r-project.org/src/contrib/e1071_1.5-19.tar.gz.

[29] K. Lo, R. R. Brinkman, and R. Gottardo, "Automated gating of flow cytometry data via robust model-based clustering," *Cytometry Part A*, vol. 73, no. 4, pp. 321–332, 2008.

[30] D. R. Parks, M. Roederer, and W. A. Moore, "A new "logicle" display method avoids deceptive effects of logarithmic scaling for low signals and compensated data," *Cytometry Part A*, vol. 69, no. 6, pp. 541–551, 2006.

[31] J. W. Tung, D. R. Parks, W. A. Moore, L. A. Herzenberg, and L. A. Herzenberg, "New approaches to fluorescence compensation and visualization of FACS data," *Clinical Immunology*, vol. 110, no. 3, pp. 277–283, 2004.

[32] D. Novo and J. Wood, "Flow cytometry histograms: transformations, resolution, and display," *Cytometry Part A*, vol. 73, no. 8, pp. 685–692, 2008.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2009, Article ID 235320, 9 pages
doi:10.1155/2009/235320

Research Article

Tree-Based Methods for Discovery of Association between Flow Cytometry Data and Clinical Endpoints

M. Eliot,[1] L. Azzoni,[2] C. Firnhaber,[3] W. Stevens,[4] D. K. Glencross,[4] I. Sanne,[3] L. J. Montaner,[2] and A. S. Foulkes[1]

[1] *Division of Biostatistics, University of Massachusetts, Amherst, MA 01003, USA*
[2] *Immunology Program, Wistar Institute, Philadelphia, PA 19104, USA*
[3] *Clinical HIV Research Unit, University of Witwatersrand, Johannesburg, South Africa*
[4] *Department of Hematology and Molecular Medicine, National Health Laboratory Service and University of Witwatersrand, Johannesburg, South Africa*

Correspondence should be addressed to A. S. Foulkes, foulkes@schoolph.umass.edu

Received 19 May 2009; Revised 14 August 2009; Accepted 12 October 2009

Recommended by George Luta

We demonstrate the application and comparative interpretations of three tree-based algorithms for the analysis of data arising from flow cytometry: classification and regression trees (CARTs), random forests (RFs), and logic regression (LR). Specifically, we consider the question of what best predicts CD4 T-cell recovery in HIV-1 infected persons starting antiretroviral therapy with CD4 count between 200 and 350 cell/μL. A comparison to a more standard contingency table analysis is provided. While contingency table analysis and RFs provide information on the importance of each potential predictor variable, CART and LR offer additional insight into the combinations of variables that together are predictive of the outcome. In all cases considered, baseline CD3-DR-CD56+CD16+ emerges as an important predictor variable, while the tree-based approaches identify additional variables as potentially informative. Application of tree-based methods to our data suggests that a combination of baseline immune activation states, with emphasis on CD8 T-cell activation, may be a better predictor than any single T-cell/innate cell subset analyzed. Taken together, we show that tree-based methods can be successfully applied to flow cytometry data to better inform and discover associations that may not emerge in the context of a univariate analysis.

1. Introduction

Advances in flow cytometry, and particularly technological developments that facilitate acquisition of multiparameter defined phenotypes, present new and exciting opportunities for predicting patient outcomes based on individual specific cell subset changes. This is specifically relevant in the context of studying human immunodeficiency virus (HIV), where there exists a great potential to draw from the rich array of data on host cell-mediated response to infection and drug exposures, to inform and discover patient level determinants of disease progression and/or response to antiretroviral therapy (ART). We describe three existing analytic approaches, designed specifically for uncovering complex structures, and their applications to high density multiparameter cell subset data arising from the use of flow cytometry technology. We demonstrate the usefulness of each approach for novel discovery in this context as well as the contrasting clinical associations that each approach is tailored to address.

The data motivating our research were collected during the pre-randomization stage of the South Africa Structured Treatment Interruption (SASTI) trial, an on-going non-inferiority trial that aims to determine whether patients whose ART is interrupted after achieving immune control on therapy will continue to retain the immune reconstitution benefits of therapy. Data on multiple immunological parameters were collected, by way of flow cytometry, on all study participants at start of ART and periodically over the course of the trial. The aim of our present investigation is to illustrate how tree-based machine learning algorithms can be applied to characterize the predictive capacity of a large number of immunological variables, collected at therapy initiation, with regard to a single, clinically relevant measure of immune reconstitution at a fixed time point on continuous therapy and prior to randomization.

We begin by presenting briefly a commonly applied, univariate analysis approach for testing the association between each immunological parameter, individually, and the outcome of interest. We then present three tree-based methods that are designed for discovery of complex structures of association in high-dimensional data settings: (1) classification and regression trees (CARTs) [1]; (2) random forests (RFs) [2]; (3) logic regression (LR) [3, 4]. These methods have been described recently for many high-throughput data settings, including most notably gene chip arrays [5–12]; however, to our knowledge, the application of these analytic approaches to discover predictors of clinical outcomes based on data arising from flow cytometry technologies has not been reported previously.

Notably, the usefulness of CART for immunophenotyping is discussed in Beckman et al. [13], with a review in Boddy et al. [14]. In our setting, the underlying goal differs in that we aim to explore the clinical utility of a large number of a priori defined phenotypes, rather than identify new phenotypes based on a comparatively small number of measurements. Also of note, in an earlier manuscript, Ganju et al. apply CART to identify predictors of censored survival time among patients with cerebral gliomas [15]. Inputs in the analysis include five flow cytometry variables, as well as cytogenetic, molecular and clinical markers. Our investigation extends this research, through consideration of a large number of multiparameter subsets, and by offering a discussion of multiple tree-based approaches, as well as their comparative interpretations, for discovery of associations between these subsets and a clinical endpoint.

2. Data and Laboratory Methods

The SASTI trial began in 2006 and led to the successful recruitment of $n = 127$ HIV-1 infected individuals, of whom $n = 78$ individuals completed the 36-week prerandomization phase of the trial. Eligibility criteria for the study included documented HIV-1 infection, 18 years of age or older, and a CD4+ count between 200 and 350 cells/μL in the absence of therapy and within 60 days of the start of the study. All individuals in the trial received a similar ART regimen for the first 36 weeks, and then were randomized to either multiple short-term treatment interruptions or continuous therapy. The present investigation focuses only on prerandomization data, when all subjects are still on ART, as the trial is still ongoing as of August 2009.

Cellular immunophenotypes were studied using flow cytometry. Stainings were performed on fresh whole blood at the Department of Hematology and Molecular Medicine, National Health Laboratory Service and University of the Witwatersrand, Johannesburg, South Africa. Briefly, whole blood samples were stained for surface marker detection using fluorochrome-labeled monoclonal antibodies (mAbs) lyophilized on 96-well plates (Lyoplates, BD Biosciences, San Jose, CA). Fluorochrome binding was detected using a 4-color FacsCalibur flow cytometer (BD Biosciences). Cellular subests were analysed using proprietary software (CellQuest, BD Biosciences). Percent of positive cells was calculated based on isotype-matched control mAb binding. Whole blood samples were stained with monoclonal antibody (mAb) combinations (given in Table 1) for 30 minutes, followed by lysis and analysis on a FACScaliber flow cytometer (BD Biosciences). Given the limitation of the instrument (simultaneous detection of 4-color fluorescence), multiple stainings were performed to assess subsets of CD3+ T lymphocytes. The gating strategy is summarized as follows.

(1) Background staining was assessed using isotype-matched mAb (staining 1—this method is generally considered acceptable for surface flow cytometry of lymphocytes).

(2) Postrun electronic event gating was performed using CellQuest software (BD Biosciences), based on the use of multiple 2-color quadrants. A first gating assessed expression of CD3 and CD8 (stainings 2, 3, 4, 6), CD3 and HLA-DR (staining 5), CD3 and CD45 (staining 7), and Lin-1 and HLA-DR (staining 8). Events falling in the quadrants of interest were further gated using quadrants to explore the expression of the remaining markers. The number of events falling in each quadrant was collected. Results are expressed as percent of gated/total events unless otherwise specified.

(3) For T cell subset assessment, the CD4+ T lymphocyte subset was directly stained using CD4 mAb only in staining 7 (single platform CD4 count [16]). Based on the mutually exclusive expression of CD8 and CD4 in the vast majority of T cells (as also assessed in staining 7), in all remaining T cell stainings (2, 3, 4, and 6) CD4+ T cells were defined as CD3+ cells lacking expression of CD8.

In this paper, we focus on assessing the relationships among multiple baseline flow cytometry variables collected at initiation of ART and the variability in achieving a robust CD4+ T-cell count rise on ART, in the context of restricting the range of starting CD4 count between 200 and 350 cells/μL. A complete listing of the baseline flow variables is given in the first column of Table 2. These are fluorescence-based cell phenotypes following intensity threshold gates using two to four fluorochromes. Four replicates, based on independent data acquisitions, were recorded for each of the phenotypes, CD3-CD8-, CD3+CD8-, CD3-CD8+, and CD3+CD8+ and averaged for the analysis. After combining these data, there are a total of 63 flow variables. All variables are measured as a percent of gated at baseline, with the exception of CD4+ which is a cell count. CD4+ T-cells, which are targeted in the viral replication cycle, play an important role in the functioning of the host immune system and are a well-described marker for disease progression when decreasing and as a response to ART based on its inverse relation to viral replication [17]. A CD4+ cell count of greater than 450 cells/μL at 36 weeks on ART is considered a positive response to ART within this study and serves as

Table 1: 4 Color stainings employed for flow cytometric analysis.

Staining no.	FITC	PE	PerCP cy5.5	APC
1	Ig	Ig	Ig	Ig
2	CD45RA	CD62L	CD3	CD8
3	CD38	CD28	CD3	CD8
4	HLA-DR	CD95	CD3	CD8
5	CD56	CD16	CD3	HLA-DR
6	CD7	CD154	CD3	CD8
7	CD8	CD4	CD45	CD3
8	Lin-1	CD123	HLA-DR	CD11c

the outcome in our present investigation. Notably, while this dichotomized version of CD4+ cell count is used in our study, the analytic methods we present are equally applicable to both binary and quantitative outcomes.

3. Methods

We present a univariate analysis and three tree-based algorithms. The tree-based approaches involve recursive splitting of the data, based on the value of predictor variables, in a manner that broadly captures the variability in a single outcome. All three approaches are nonparametric and can be applied in the context of a large number of predictors and a single binary or quantitive trait. Both CART and RFs can handle both quantitative and binary predictor variables, while logic regression requires dichotomous inputs. For clarity of presentation, we dichotomize all of the potential predictors a priori. Further discussion of this, including model sensitivity to choice of inputs, is given in Section 5. We begin by briefly defining our notation.

3.1. Notation and Univariate Analysis. Suppose we have p predictor variables based on the outcome of flow cytometry at a single time point. We denote these with the vector $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$ for individual i, where $i = 1, \dots, n$. The $n \times p$ matrix $\mathbf{X}$ is used to denote the full data design matrix with (i, j)-element corresponding to the value of variable j for individual i. Subjects are assumed to be independent, though we expect correlation among the predictors. Interest lies in characterizing the association between $\mathbf{X}$ and a measured trait, which we denote with the vector $\mathbf{y} = (y_1, y_2, \dots, y_n)$ for the n individuals in our study. In our setting, each of the columns of $\mathbf{X}$, denoted X_j, is a measure of the flow variables and the outcome of interest is a binary indicator for CD4+ cell count >450 cells/μL. We define each X_j as an indicator for being above or below the sample median value for that variable.

Measuring and testing the association between a single categorical predictor and a binary outcome is typically achieved through a contingency table analysis. The odds ratio, defined as the odds of disease given exposure, divided by the odds of disease given no exposure, is a well-described measure of association in the this context and is given formally by

$$OR = \frac{Pr(D^+ \mid E^+)/[1 - Pr(D^+ \mid E^+)]}{Pr(D^+ \mid E^-)/[1 - Pr(D^+ \mid E^-)]}. \quad (1)$$

In our setting, we report the odds of having a CD4+ cell count of more than 450 cells/μL (D^+) given that a specific baseline flow variable is in the upper half of its distribution (E^+), over the odds of having a CD4+ cell count >450 cells/μL given that this flow variable is in the lower half (E^-). Pearson's χ^2-test can be applied as a test of the null hypothesis of no association between exposure and disease for each flow variable independently. An adjustment of the resulting P-values, that accounts for the number of tests performed, is needed in this setting for assessing statistical significance. We report the q-value which is based on a positive false discovery rate adjustment [18, 19].

3.2. Classification and Regression Trees. Classification and Regression Trees (CARTs) are an alternative, nonparametric approach that allows us to model simultaneously the relationship between an outcome and *multiple* potential predictor variables. This approach provides us with information on variable importance as well as the structure of association. Classification trees are constructed for binary outcomes while regression trees apply to continuous traits. Both binary and continuous predictor variables are acceptable inputs, though trees are constructed based on binary splits of these data. The first step in generating a tree is to determine the most predictive variable of the trait, which we denote $X_{(1)}$, based on a prespecified splitting rule. Secondly, we divide individuals into groups based on the value of $X_{(1)}$ and determine the most predictive variable of the outcome within each of these groups. This process is repeated recursively until a stopping criterion is met and then the resulting tree is pruned back to avoid over-fitting. Tree construction is sensitive to the choice of splitting rule, and ultimately, we want to define such a rule so that we partition our data in a manner that minimizes the within group heterogeneity in the outcome. Here we describe the CART methodology generally, though in the example we present a classification tree since we are considering a binary outcome.

Formally, let the node Ω represent the full set of data and suppose after splitting the data based on one of the predictor

Table 2: Univariate associations with CD4+ count at 36 weeks on ART.

Predictor	Odds ratio	*P*-value
CD3-DR-CD56+CD16+	0.183	.008
Lin-DR-	0.228	.018
CD45+CD3+	0.274	.035
CD3+CD8+CD38+CD28+	0.281	.047
CD3-CD8+	0.323	.084
CD3-DR+CD56+CD16+	0.339	.087
CD3+CD8-CD7+CD154+	0.339	.087
CD3-DR-CD56-CD16-	0.364	.113
CD3+CD8+CD7+CD154+	0.388	.463
CD3+CD8-CD7+CD154-	0.389	.146
CD3+CD8-DR+CD95-	0.429	.189
CD3+DR-	0.460	.236
CD3+CD8-CD45RA+CD62L+	0.477	.283
CD3+CD8-DR+CD95+	0.477	.283
CD45-CD3+	0.494	.632
CD3+CD8-	0.494	.632
Lin-DR+CD123+CD11c+	0.564	.424
CD3+CD8+DR+CD95+	0.628	.571
CD3-DR+	0.646	.586
CD3+CD8-DR-CD95-	0.646	.586
CD45+CD3+CD8-CD4-	0.703	.690
CD3+CD8+CD38-CD28+	0.709	.699
CD3+CD8-CD7-CD154-	0.740	.774
CD3+CD8+	0.752	.786
CD3+CD8-CD38+CD28+	0.759	.797
CD3+CD8+CD38+CD28-	0.760	.812
CD3-DR+CD56-CD16-	0.805	.887
CD3-DR+CD56+CD16-	0.805	.887
CD3+CD8+CD38-CD28-	0.813	.898
CD45-CD3-	0.862	.989
Lin-DR+CD123+CD11c-	0.913	.917
CD3+CD8+CD45RA+CD62L-	0.923	.908
CD3+CD8+CD45RA-CD62L	0.931	.898
CD45+CD3+CD8-CD4+	0.931	.898
CD3+CD8+DR+CD95-	0.938	.887
CD3+CD8-CD7-CD154+	0.962	.696
CD3+CD8-CD38+CD28-	0.996	.797
CD3+CD8+DR-CD95-	1.004	.797
Lin+DR+	1.074	.898
CD3-DR-CD56-CD16+	1.074	.898
CD3-CD8-	1.074	.898
Lin+DR-	1.149	1.000
CD45+CD3-	1.160	.989
CD3+CD8-CD38-CD28-	1.160	.989
CD3+CD8-CD45RA+CD62L-	1.230	.898
CD45+CD3+CD8+CD4-	1.317	.797
CD3+DR+	1.329	.786
Lin-DR+CD123-CD11c-	1.329	.786
CD3+CD8-DR-CD95+	1.329	.786
CD3+CD8+DR-CD95+	1.410	.699
CD45+CD3+CD8+CD4+	1.410	.699
CD3+CD8+CD45RA-CD62L+	1.422	.690
CD3-DR-	1.446	.677
CD3-DR+CD56-CD16+	1.486	.661
Lin-DR+	1.511	.605
CD4+	1.522	.598
Lin-DR+CD123-CD11c+	1.522	.598
CD3+CD8-CD45RA-CD62L+	1.630	.512
CD3-DR-CD56+CD16-	1.630	.512
CD3+CD8+CD7+CD154-	1.657	.502
CD3+CD8+CD7-CD154-	1.707	.487
CD3+CD8-CD38-CD28+	1.898	.354
CD3+CD8-CD45RA-CD62L	2.011	.294
CD3+CD8+CD45RA+CD62L+	2.152	.238

variables, we have two groups, Ω_L and Ω_R, called the left and right daughter nodes, respectively. If the node impurity, or heterogeneity, for Ω is denoted $\mathcal{I}(\Omega)$, then we aim to identify the split that maximizes

$$\phi = \mathcal{I}(\Omega) - \mathcal{I}(\Omega_L) - \mathcal{I}(\Omega_R). \tag{2}$$

That is, we want to choose a split that maximizes the reduction in node impurity. In the context of a binary outcome ($y = 0$ or 1), we let $\mathcal{I}(\Omega) = \pi(\Omega)i(\Omega)$ where π is the probability of belonging to Ω, so that (2) reduces to

$$\phi = i(\Omega) - \pi_L i(\Omega_L) - \pi_R i(\Omega_R). \tag{3}$$

The impurity, $i(\Omega)$, is commonly measured using the Gini index [12], defined as

$$i(\Omega) = 2p_\Omega(1 - p_\Omega), \tag{4}$$

where $p_\Omega = \Pr(y = 1 \mid \Omega)$ is the conditional probability that y is equal to 1 within the node Ω.

Once a tree is constructed, as shown in Figure 1, we prune it to ensure its applicability to external datasets. Importantly, increasing the number of splits in a tree will inevitably decrease the prediction error for the data used to generate the tree. However, a smaller tree may better describe the underlying structure in the population at large. Therefore, after we build a tree, as described above, we prune it in order to get an optimal subtree, using cost-complexity pruning. Briefly, for tree $\mathcal{T}$ of size $|\mathcal{T}|$ and complexity parameter $\alpha \geq 0$, the cost complexity is given by

$$R_\alpha = R(\mathcal{T}) + \alpha|\mathcal{T}|, \tag{5}$$

where

$$R(\mathcal{T}) = \sum_{\tau \in \tilde{\mathcal{T}}} \Pr(\tau) r(\tau), \tag{6}$$

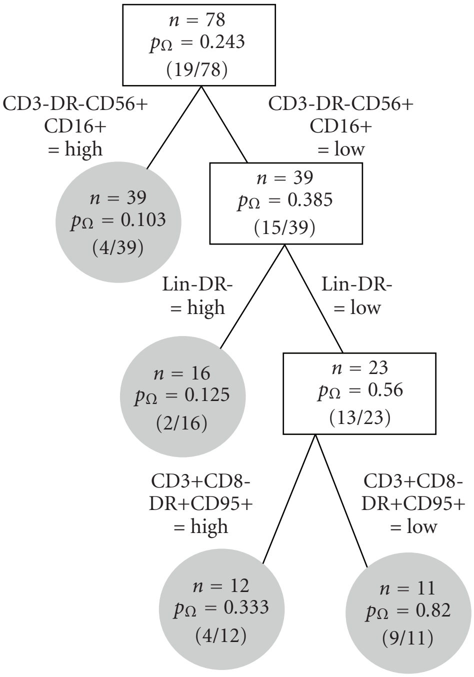


Figure 1: Classification tree (unpruned).

$\tilde{\tilde{\mathcal{T}}}$ is the set of terminal nodes in tree $\mathcal{T}$ and $r(\tau)$ is the measure of error for the node τ. In the case of a binary outcome, we let $r(\tau)$ equal the misclassification rate.

3.3. Random Forests. Random Forest (RF), originally proposed by [2], is an alternative approach that involves generating a collection of trees. Since this approach results in an ensemble of trees, which tend to vary in structure, RFs serve to quantify the importance of variables, rather than depicting the specific structure of association among variables. A primary advantage of RFs is that, through sampling a subset of variables at each split, it offers a natural approach to handling collinearity among the predictors. In this paper, we demonstrate the application of RFs as an exploratory tool, although methods for determining statistical significance based on variable importance scores have been described recently [20, 21].

The RF algorithm is summarized by the following step-by-step procedure: (1) generate a learning sample by sampling n_1 individuals with replacement from our data (usually about two-thirds of the data). We call the remaining $n_2 \approx n - n_1$ data the out-of-bag (OOB) data; (2) using the learning sample data, generate an unpruned tree by randomly sampling a subset of the predictors at that node. These predictors will be used as our variables on which our splitting decisions are based (3) based on the OOB data, find the overall tree impurity, and call this π_b. Permute the predictor X_j and record the overall tree impurity for each $j = 1, \ldots, p$. Call tree impurity for the jth predictor π_{bj} and call variable importance for this predictor $\delta_{bj} = \pi_{bj} - \pi_b$. (4) repeat steps (1)–(3) for $b = 2, \ldots, B$ in order to obtain $\delta_{1j}, \ldots, \delta_{Bj}$ for each j.

For each predictor, j, the overall variable importance score is given by the average importance over the B trees. Formally, we write

$$\hat{\theta}_j = \frac{1}{B}\sum_{b=1}^{B}\delta_{bj}. \qquad (7)$$

Notably, for each tree, a learning sample is used in the tree construction, while an independent test sample, called the OOB data, is used to evaluation variable importance.

3.4. Logic Regression. Logic regression (LR) is another tree-based approach that is increasingly popular for the analysis of high-dimensional data. LR searches specifically for models that are comprised of combinations of Boolean expressions of the predictors [3, 4]. Boolean expressions take on the value of either 0 or 1, and are themselves functions of binary variables, related to each other by "and," "or," and "complement" statements. Formally, LR models are of the form

$$g(E[Y \mid \mathbf{X}]) = \beta_0 + \sum_{j=1}^{t}\beta_j L_j, \qquad (8)$$

where L_j is a Boolean combination of the binary predictors. Suppose that we have binary predictor variables $X_1, X_2, \ldots, X_p$ which we want to use to predict some outcome. An example of a Boolean expression in terms of our group of predictors is $(X_1 \wedge X_2) \vee (X_3 \wedge X_4^c)$, which represents "both $X_1 = 1$ and $X_2 = 1$ or both $X_3 = 1$ and $X_4 = 0$."

4. Example

We report the results of applying a univariate analysis and each of the tree-based methods described above to data arising from the SASTI trial detailed in Section 2. In total, $n = 63$ flow cytometry variables, measured at baseline, are used as potential predictors (in addition to CD4+ count at baseline). Each variable is dichotomized to indicate whether the value is above or below the median of the observed (nonmissing) values for that predictor. That is, an observation is set equal to 1 if it is greater than the median value for all observations in our sample of that predictor and 0 otherwise. A single imputation is used such that missing data points are assigned the most common value of 0 or 1, based on the nonmissing data for the corresponding variable. The outcome of our analysis is an indicator for whether CD4+ cell count is greater then 450 cells/μL at 36 weeks after initiation of ART, which represents the last time point prior to randomization.

The univariate analysis results are provided in Table 2. Here the OR is reported as a measure of association between each flow variable at baseline and CD4+ cell count at 36 weeks on ART. The P-value corresponds to Pearson's χ^2-test of association. Based on this analysis, we see that CD3-DR-CD56+CD16+ is the most predictive variable with an OR $= 0.183$ (unadjusted $P = .008$). This suggests that the odds of having a CD4+ cell count >450 cells/μL while on therapy

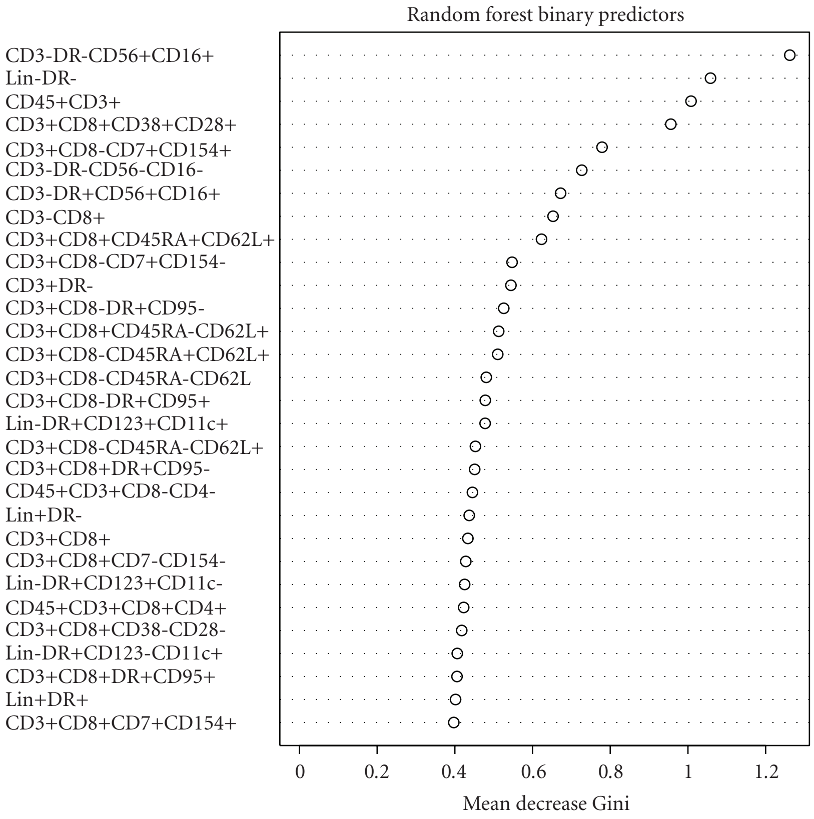
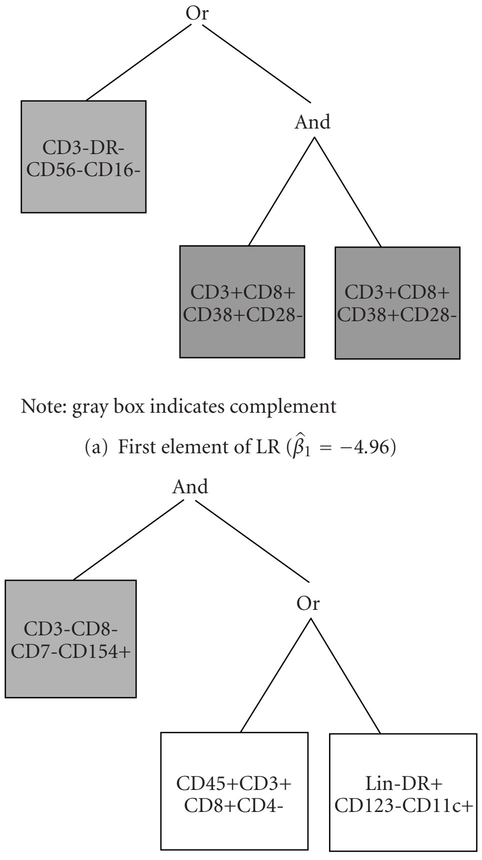


Figure 2: Variable importance scores from application of an RF.

is higher among individuals with a baseline observed CD3-DR-CD56+CD16+ that is in the lower half of our sample. Lin-DR- at baseline is the next most predictive variable, with an OR = 0.23 (unadjusted $P = .018$). After adjusting for multiple testing using the approach of Benjamini and Yekutieli [22], we cannot conclude that any of the flow variables alone are significantly associated with CD4+ count after 36 weeks. The repeated ORs reported in this table are likely due to the limited sample size in our study, as clear relationships among these pairs and triplets of variables are not generally well-established.

An unpruned classification tree, based on a stopping rule of $n = 5$ individuals per node, is illustrated in Figure 1. This model yields five terminal nodes, indicated by the shaded circles, resulting from splits based on CD3-DR-CD56+CD16+, Lin-DR- and CD3+CD8-DR+CD95+. The first split indicates, for example, that for high CD3-DR-CD56+CD16+ (i.e., CD3-DR-CD56+CD16+ greater than the median), only $p_{\Omega} = 4/39 = 10.3\%$ of the individuals in our sample have an observed CD4+ count that is greater than 450, while for low CD3-DR-CD56+CD16+ (i.e., CD3-DR-CD56+CD16+ less than the median), $p_{\Omega} = 15/39 = 38.5\%$ of individuals have a CD4+ cell count that is greater than 450 cells/μL. Among those individuals who fall to the right daughter node (i.e., low CD3-DR-CD56+CD16+), the next most important predictor is Lin-DR-. When CD3-DR-CD56+CD16+ is low and Lin-DR- is high, $2/16 = 12.5\%$ of the subjects in our sample have an observed CD4+ count that is greater than 450. On the other hand, when both CD3-DR-CD56+CD16+ and Lin-DR- are low, a much higher percentage ($13/23 = 56.5\%$) of individuals have a CD4+ count greater than 450 cells/μL. Application of cost-complexity pruning resulted in a tree with no splits, suggesting that these findings may not be reproducible in an independent sample. This may be a consequence of limited power in our small sample setting.

The results of applying the RF algorithm to these data are given in Figure 2. Here we see that the most important baseline predictor of CD4+ count on ART is again CD3-DR-CD56+CD16+, with a mean decrease in node impurity of 1.26. The next most important variable is Lin-DR- (also the second split in our classification tree), with a corresponding mean decrease in node impurity of 1.05. These results are generally consistent with the univariate analysis of Table 2 and to some extent with the classification tree of Figure 1; however, some notable differences are apparent. First, the RF analysis places more emphasis on CD45+CD3+ as an important predictor than the CART analysis. Interestingly, CD45+CD3+ is also the third most important variable in the univariate analysis. Since the classification tree is considering a series of conditional analyses, this difference may be a result of CD45+CD3+ not having a strong association *within* levels of the first splitting variable, CD3-DR-CD56+CD16+. Secondly, the classification tree analysis places greater



Note: gray box indicates complement

(a) First element of LR ($\hat{\beta}_1 = -4.96$)

Note: gray box indicates complement

(b) Second element of LR ($\hat{\beta}_2 = -3.79$)

Figure 3: Logic regression trees.

emphasis on CD3+CD8-DR+CD95+ than either the RF or univariate approaches. This specifically lends some insight into a potential effect of the combination of CD3-DR-CD56+CD16+, Lin-DR-, and CD3+CD8-DR+CD95+.

Finally, we applied LR to the data and the resulting trees are presented in Figure 3. Here we applied a logit link function, specified that we wanted two trees and restricted the total number of "leaves" (across both trees) to 6 for ease if interpretation. The coefficient estimates for the trees in Figures 3(a) and 3(b) are $\hat{\beta}_1 = -4.96$ and $\hat{\beta}_2 = -3.79$, respectively. In this case, the variable CD3-DR-CD56-CD16- is an important predictor of CD4+ count on therapy. Notably, this variable is highly negatively correlated with CD3-DR-CD56+CD16+ (Pearson's $\rho = -0.71$), which was identified as the most important predictor of immune reconstitution based on the other approaches described above. In addition to CD3-DR-CD56-CD16- being an important predictor of immune reconstitution, we have, for example, based on the second tree, that when CD3+CD8-CD7+CD154+ is low (less than the median) and either CD45+CD3+CD8+CD4- or Lin-DR+CD123-CD11c+ is high (greater than the median) the log odds that CD4+ count is greater than 450 cells/μL decreases by 3.79, compared to when this does not hold.

5. Discussion

The goal of this study is to compare a number of tree-based methods for their capability to select immunological predictors of CD4 reconstitution in HIV-infected subjects initiating antiretroviral treatment. Earlier studies from our group have demonstrated that pre-ART CD95 expression on CD8+ T cells is negatively associated with the frequency of plasmacytoid Dendritic Cells (PDCs) after 52 weeks of treatment [23]. Conversely, a positive association was also demonstrated between levels of baseline CD28 expression in CD4+ T cells and PDC recovery. Other studies have also suggested that baseline CD4 count may predict the degree of post-ART immune reconstitution [24]. However, the selection of immunologic predictors of immune reconstitution has so far been based on known biologic associations between variables (e.g., association of a certain variable with diseases stages, etc.), and data-mining methods for automated unbiased selection from a large numbers of variables remain underutilized.

We describe the application of a univariate approach and three tree-based methods for the analysis of the association between a single trait and multiple variables arising from flow cytometric analysis. Interestingly, for this data example, the univariate contingency table analysis and RFs resulted in similar findings in terms of the ranking of important variables. This may not always be the case, since as we describe in Section 3, the variable importance scores derived within the context of RFs are based on the individual effects of variables, as well as their effects within levels of other variables. In the example provided, CART and LR provided complementary information about the structure of association, and particularly the combinations of variables that are informative. Specifically, while all of the approaches suggest that CD3-DR-CD56+CD16+ is an important predictor of CD4+ count on therapy, the CART model further suggests that among individuals for whom CD3-DR-CD56+CD16+ is in the lower half of our sample, Lin-DR- is an important variable in differentiating between responders and nonresponders. Similarly, the LR analysis revealed several combinations of variables that lend further insight into determining the individual level characteristics that together are predictive of response to ART in this population. The added information on variables that are predictive of outcome, beyond those identified by univariate analysis, provides greater understanding of multiple combinations among variables that may equally predict an outcome, reflecting the potential complexity of responses among human study groups.

Notably, a high degree of correlation is intrinsic to the variables included in our analysis of flow cytometry data. Specifically, events passing a certain logical gate are assessed for co-expression of two fluorochromes, and separated in quadrants based on the intensity (above or below a certain level) of each fluorochrome. Thus, any increase in the percent of events falling in one quadrant must correspond to a decrease in the percent of events that fall in one or more of the other quadrants. For example, the variables CD3-CD8-, CD3+CD8-, CD3-CD8+, and CD3+CD8+ arise from four quadrants on the same plate for each individual and thus always sum to 100%. While each variable represents a distinct cell subset, and application of the described approaches to data with such a correlation structure is reasonable, further extensions of these methods that account for the correlation structure may offer new insights. At the same time, interpreting variable importance must be done in light of the existing correlations. For example, we saw in the example provided above that both CART and the RF identified CD3-DR-CD56+CD16+ as an important predictor of immune reconstitution, while LR identified the highly correlated variable, CD3-DR-CD56-CD16-. RFs offer a natural approach to handling correlations by sampling a subset of predictors at each stage of the tree splitting; however, using any of the approaches described, the importance of a variable may be obscured in the presence of other, very highly correlated variables. One alternative approach is to choose a priori a subset of uncorrelated variables to include in the analysis. This is reasonable if prior knowledge suggests multiple variables are defining the same underlying construct but may be less optimal if the precise relationship among variables is unknown.

This paper represents an attempt to utilize data from experimental and clinical laboratory settings that are available in resource constrained settings. While it is general good scientific practice to avoid unnecessary assessment, limiting stainings and maximizing the usefulness of current resource capacity is paramount in the settings in which these experiments were conducted. Because the use of multicolor flow cytometers is restricted to resource-rich clinical and research settings, we have elected to use the output of more commonly available 4-colour analytical instruments, in the hope that any information gained from this approach is applicable in the resource-constrained settings such as those in which the study was conducted. We also agree that the clinical interpretability of the findings in this data setting is limited. Specifically, the full panel of mAb used for this paper would not be applicable to general practice, particularly in resource constrained settings, due to issues of cost and laboratory capacity. This panel was in fact used in an experimental setting, to investigate in detail the effects of ART on individual immune subsets. However, the purpose of this paper is to identify which, among the baseline, pre-ART stainings performed, could be useful to predict the desired outcome (in this case immune reconstitution as assessed by CD4 counts). We demonstrated how tree-based approaches can be applied to identify a small number of phenotypes that contribute to the selected CD4 recovery outcome. Importantly, many of the cellular subsets (e.g., mature NK cells, myeloid Dendritic cells, CD95-expressiong activated T lymphocytes) selected using the three tree-based methods presented here as being predictive of immune reconstitution have been previously shown to be individually correlated with disease progression and/or immune reconstitution [25, 26], thus further supporting the reasonableness of our approach. CD4 is presently the only validated tool to monitor immune competence in HIV-infected individuals. However, because pre-ART CD4 counts are notoriously poor predictors of clinical response to ART, the identification of a limited number of variables that could be used as additional predictors in larger prospective studies represents an important contribution to the field. Selected stainings can be recombined in smaller panels, reducing cost and capacity consumption; for example, based on the logic regression trees presented in Figure 3, the use of only two staining combinations (e.g., CD3/CD8/CD7/CD154 and CD45/CD3/CD4/CD8) would be sufficient to predict a CD4 immune reconstitution outcome.

Importantly, differences in the insights offered by each of the approaches presented are a reflection of the specific algorithms employed and not the result of one approach being more or less correct than another. The univariate analysis, while methodologically sound, only considers associations that exist between single variables and the outcome. Univariate analyses are not designed to discover variables that are only important conditional on the level of another variable. The CART and RF algorithms, on the other hand, are specifically searching for conditional associations, that is, associations of variables with the outcome within levels of other variables. Finally, logic regression trees allow for discovery of combinations of variables that are predictive, even in the setting in which no single element of the combination is important on its own. That is, both CART and RF split initially on the single most important variable; however, if a combination of two or more variables is important, none of which are predictive individually, then both CART and RF may not find this association [12, 27]. The LR algorithm, on the other hand, is designed specifically to capture this information.

In summary, each of the tree-based approaches described herein complement univariate analyses of multiparameter defined flow cytometry subsets. These methods are designed specifically to uncover complex structures, and as demonstrated in the example above, allow for discovery of combinations of variables that are together predictive of an outcome. While extensions of these methods, including, for example, the recently proposed approach of [20], would allow for measuring statistical significance of variable importance scores, their strength lies in the discovery of combinations of variables that are potentially associated with the outcome. In all of the approaches presented, a type of cross-validation algorithm is applied, which renders the results theoretically applicable to independent samples. However, as with all exploratory analyses, further hypothesis driven research will enable further validation of true underlying associations.

Acknowledgments

Support for this research was provided by NIH/NIAID R01AI056983 to M. Eliot and A. S. Foulkes and U01AI51986 to L. J. Montaner.

References

[1] L. Breiman, J. Friedman, C. Stone, and R. A. Olshen, *Classiffication and Regression Trees*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 1984.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] C. Kooperberg, I. Ruczinski, M. L. LeBlanc, and L. Hsu, "Sequence analysis using logic regression," *Genetic Epidemiology*, vol. 21, supplement 1, pp. S626–S631, 2001.

[4] I. Ruczinski, C. Kooperberg, and M. LeBlanc, "Logic regression," *Journal of Computational and Graphical Statistics*, vol. 12, no. 3, pp. 475–511, 2003.

[5] M. R. Segal, J. D. Barbour, and R. M. Grant, "Relating HIV-1 sequence variation to replication capacity via trees and forests," *Statistical Applications in Genetics and Molecular Biology*, vol. 3, no. 1, article 7, 2004.

[6] A. Bureau, J. Dupuis, K. Falls, et al., "Identifying SNPs predictive of phenotype using random forests," *Genetic Epidemiology*, vol. 28, no. 2, pp. 171–182, 2005.

[7] C. Kooperberg and I. Ruczinski, "Identifying interacting SNPs using Monte Carlo logic regression," *Genetic Epidemiology*, vol. 28, no. 2, pp. 157–170, 2005.

[8] C. Kooperberg, J. C. Bis, K. D. Marciante, S. R. Heckbert, T. Lumley, and B. M. Psaty, "Logic regression for analysis of the association between genetic variation in the renin-angiotensin system and myocardial infarction or stroke," *American Journal of Epidemiology*, vol. 165, no. 3, pp. 334–343, 2007.

[9] H. Schwender and K. Ickstadt, "Identification of SNP interactions using logic regression," *Biostatistics*, vol. 9, no. 1, pp. 187–198, 2008.

[10] K. Ickstadt, M. Schäfer, A. Fritsch, et al., "Statistical methods for detecting genetic interactions: a head and neck squamous-cell cancer study," *Journal of Toxicology and Environmental Health, Part A*, vol. 71, no. 11-12, pp. 803–815, 2008.

[11] M. García-Magariños, I. López-de-Ullibarri, R. Cao, and A. Salas, "Evaluating the ability of tree-based methods and logistic regression for the detection of SNP-SNP interaction," *Annals of Human Genetics*, vol. 73, no. 3, pp. 360–369, 2009.

[12] A. Foulkes, *Applied Statistical Genetics with R for Population-Based Association Studies*, Springer, Berlin, Germany, 2009.

[13] R. J. Beckman, G. C. Salzman, and C. C. Stewart, "Classification and regression trees for bone marrow immunophenotyping," *Cytometry*, vol. 20, no. 3, pp. 210–217, 1995.

[14] L. Boddy, M. F. Wilkins, and C. W. Morris, "Pattern recognition in flow cytometry," *Cytometry*, vol. 44, no. 3, pp. 195–209, 2001.

[15] V. Ganju, R. B. Jenkins, J. R. O'Fallon, et al., "Prognostic factors in gliomas: a multivariate analysis of clinical, pathologic, flow cytometric, cytogenetic, and molecular markers," *Cancer*, vol. 74, no. 3, pp. 920–927, 1994.

[16] D. K. Glencross, G. Janossy, L. M. Coetzee, et al., "Large-scale affordable PanLeucogated CD4$^+$ testing with proactive internal and external quality assessment: in support of the South African national comprehensive care, treatment and management programme for HIV and AIDS," *Cytometry Part B*, vol. 74, supplement 1, pp. S40–S51, 2008.

[17] L. E. Harrington, R. D. Hatton, P. R. Mangan, et al., "Interleukin 17-producing CD4$^+$ effector T cells develop via a lineage distinct from the T helper type 1 and 2 lineages," *Nature Immunology*, vol. 6, no. 11, pp. 1123–1132, 2005.

[18] J. D. Storey, "A direct approach to false discovery rates," *Journal of the Royal Statistical Society, Series B*, vol. 64, no. 3, pp. 479–498, 2002.

[19] J. D. Storey, "The positive false discovery rate: a Bayesian interpretation and the *q*-value," *The Annals of Statistics*, vol. 31, no. 6, pp. 2013–2035, 2003.

[20] M. J. van der Laan, "Statistical inference for variable importance," *The International Journal of Biostatistics*, vol. 2, no. 1, article 2, 2006.

[21] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinformatics*, vol. 9, article 307, 2008.

[22] Y. Benjamini and D. Yekutieli, "The control of the false discovery rate in multiple testing under dependency," *The Annals of Statistics*, vol. 29, no. 4, pp. 1165–1188, 2001.

[23] J. Chehimi, L. Azzoni, M. Farabaugh, et al., "Baseline viral load and immune activation determine the extent of reconstitution of innate immune effectors in HIV-1-infected subjects undergoing antiretroviral treatment," *The Journal of Immunology*, vol. 179, no. 4, pp. 2642–2650, 2007.

[24] D. Nash, M. Katyal, M. W. G. Brinkhof, et al., "Long-term immunologic response to antiretroviral therapy in low-income countries: a collaborative analysis of prospective studies," *AIDS*, vol. 22, no. 17, pp. 2291–2302, 2008.

[25] S. R. Søndergaard, H. Aladdin, H. Ullum, J. Gerstoft, P. Skinhøj, and B. K. Pedersen, "Immune function and phenotype before and after highly active antiretroviral therapy," *Journal of Acquired Immune Deficiency Syndromes*, vol. 21, no. 5, pp. 376–383, 1999.

[26] J. Chehimi, D. E. Campbell, L. Azzoni, et al., "Persistent decreases in blood plasmacytoid dendritic cell number and function despite effective highly active antiretroviral therapy and increased blood myeloid dendritic cells in HIV-infected individuals," *The Journal of Immunology*, vol. 168, no. 9, pp. 4796–4801, 2002.

[27] A. S. Foulkes, V. De Gruttola, and K. Hertogs, "Combining genotype groups and recursive partitioning: an application to human immunodeficiency virus type 1 genetics data," *Journal of the Royal Statistical Society, Series C*, vol. 53, no. 2, pp. 311–323, 2004.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2009, Article ID 103839, 4 pages
doi:10.1155/2009/103839

Resource Review

iFlow: A Graphical User Interface for Flow Cytometry Tools in Bioconductor

Kyongryun Lee, Florian Hahne, Deepayan Sarkar, and Robert Gentleman

Program in Computational Biology, Division of Public Health Sciences, Fred Hutchinson Cancer Research Center,
1100 Fairview Avenue N M2-B876, P.O. Box 19024, Seattle, WA 98109-1024, USA

Correspondence should be addressed to Florian Hahne, fhahne@fhcrc.org

Received 3 April 2009; Accepted 10 August 2009

Recommended by George Luta

Flow cytometry (FCM) has become an important analysis technology in health care and medical research, but the large volume of data produced by modern high-throughput experiments has presented significant new challenges for computational analysis tools. The development of an FCM software suite in Bioconductor represents one approach to overcome these challenges. In the spirit of the R programming language (Tree Star Inc., "FlowJo"), these tools are predominantly console-driven, allowing for programmatic access and rapid development of novel algorithms. Using this software requires a solid understanding of programming concepts and of the R language. However, some of these tools—in particular the statistical graphics and novel analytical methods—are also useful for nonprogrammers. To this end, we have developed an open source, extensible graphical user interface (GUI) *iFlow*, which sits on top of the Bioconductor backbone, enabling basic analyses by means of convenient graphical menus and wizards. We envision *iFlow* to be easily extensible in order to quickly integrate novel methodological developments.

1. Introduction

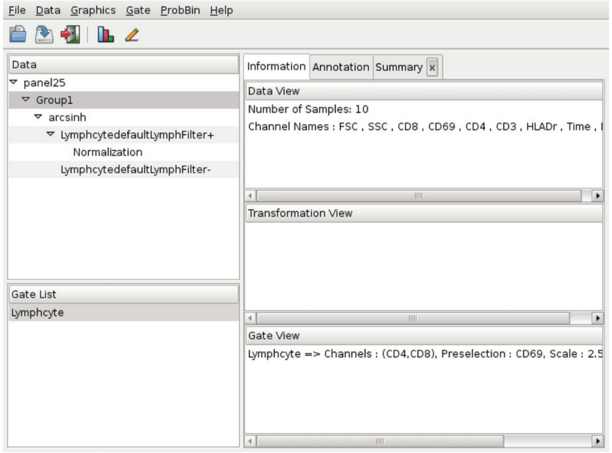
The analysis of large and highly complex datasets produced by modern high-throughput biomedical research can be a daunting task. Programmatic approaches, batch processing, and targeted analysis pipelines are typically employed to deal with this growing complexity. These solutions usually require considerable programming proficiency, or a rigid workflow structure that can be bundled into a static pipeline.

Flow cytometry (FCM) is an important emerging technology in immunology, cancer research, and health care. The technology is extremely versatile, and a multitude of different applications have been developed, which is reflected in a complicated and multilayered data analysis process. The analysis of FCM data has traditionally relied heavily on manual decision-making, and FCM software platforms typically present an interactive graphical user interface (GUI) as their primary interface [1, 2].

However, the sheer volume of data in high-throughput FCM experiments makes it impossible for an expert to efficiently perform fully manual analyses, and a certain degree of automation has become essential [3–5]. In the Bioconductor project [6], we have implemented a set of flexible command-line tools to facilitate the analysis of complex FCM data [7, 8]. The goal of the software is to foster the development of novel analytic methods by providing an open and extensible research platform that enables collaboration between bioinformaticians, computer scientists, statisticians, biologists, and clinicians. In order to succeed in this goal, we need to engage experienced practitioners who do not necessarily have programming skills. *iFlow* is a cross-platform software application meant to expose the tools and methods available in the Bioconductor project to such an audience by means of an interactive, extensible, and locally customizable GUI.

2. Results

iFlow is implemented using the Gtk2 toolkit [9, 10], and sits on top of R and Bioconductor. It allows convenient management, visualization, and analysis of FCM data. On startup, *iFlow* will open an application window (Figure 1). Subsequently, one or more additional graphics windows may also be opened (Figure 2). The application window consists of a control panel and the main panel. The control panel lists all available datasets and gates, and allows the user to select one. All operations are pefomed on the currently selected



Data
- panel25
  - Group1
    - arcsinh
      - LymphcytedefaultLymphFilter+
        - Normalization
      - LymphcytedefaultLymphFilter-

Gate List
Lymphcyte

Data View
Number of Samples: 10
Channel Names : FSC , SSC , CD8 , CD69 , CD4 , CD3 , HLADr , Time , I

Transformation View

Gate View
Lymphcyte => Channels : (CD4,CD8), Preselection : CD69, Scale : 2.5

Figure 1: iFlow's main application window. Details on the components are given in the main text.

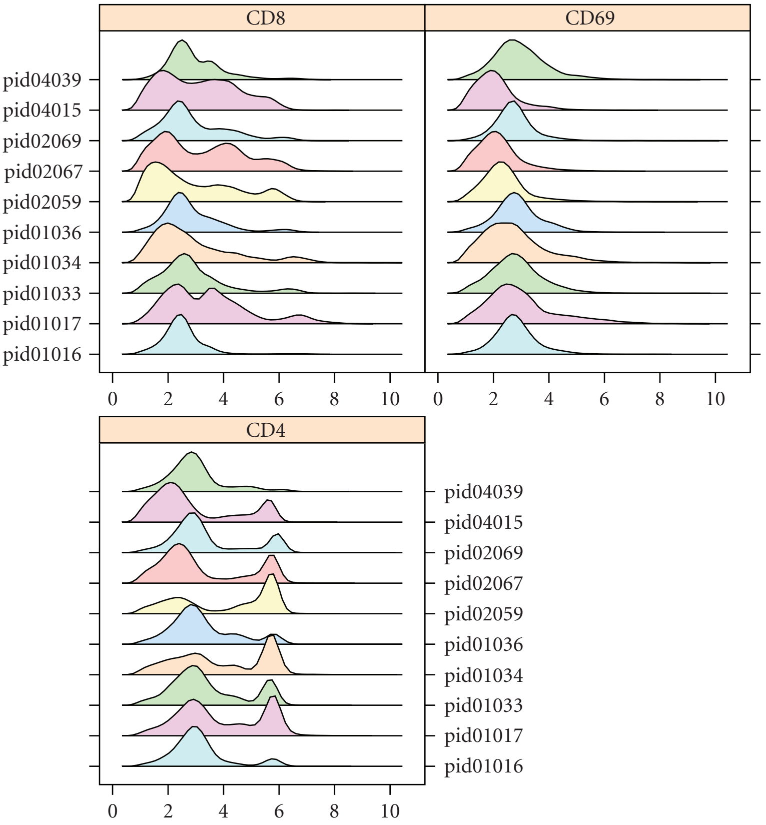


Figure 2: An iFlow graphics window displaying data of three FCM channels in the form of stacked density plots for ten different samples.

dataset. The main panel consists of a notebook with three types of tabs: *Information*, *Annotation*, and *Summary*; their contents are context-dependent.

The *Information* tab provides details about the currently selected data-set. It also displays information about previously defined gates and transformations, which can be reused in other tasks. The *Annotation* tab provides phenotypic information about the individual samples in the current data-set. It is possible to subset based on these covariates. Various summaries of the data can be displayed in additional *Summary* tabs.

A range of visualization methods are available from the *Graphics* menu. These include contour plots, density plots, scatter plots, ECDF plots, histograms, parallel coordinate plots, *Q-Q* plots, scatter plot matrices, and time series plots. Typical gating operations like rectangular and polygonal selections are available from the *Gate* menu. *iFlow* also supports basic interactive drawing of gates. In addition, a number of automated gating algorithms are available, offering data-driven selection of distinct cell populations. More general data manipulation operations are available in the *Data* menu, including various transformation options and data subsetting based on previously defined gates.

2.1. A Sample Session. Detailed usage instructions for *iFlow* are available in the manual accompanying the package. Here, we highlight some of the features one might use in a typical session. Data in the form of FCS files or R binary data files can be read in using the `File|Load` menu item. This step adds one or more data entries to the *Data* tab in the control panel. Selecting one such entry brings up a brief description of the associated dataset in the *Information* tab, as well as a tabular view of the sample covariates (e.g., Group ID, Patient ID, Visit number, etc.) in the *Annotation* tab.

As a next step, we may wish to create a new data-set with the subset of samples from a particular patient group. To do this, we first select the appropriate rows in the *Annotation*, and use the `Subset` item in the context menu that can be brought up using the right mouse button. Alternatively, one can use the `Data|Subset By|Sample Covariates` menu item. This creates a new data entry in the control panel that is nested within the original dataset.

We can next inspect the data graphically using items in the `Graphics` menu. A useful overview is given by stacked density plots (Figure 2). Such inspection may indicate the need to transform the data, which can be achieved using the `Data|Transformation` menu item.

The usual next step is to select a specific cell subtype for further analysis, for instance lymphocytes. Various types of gates can be created using the `Gate|Create` menu item; the list includes a *Lymphocyte gate* which tries to automatically select lymphocytes given a pair of channels and a third preselection channel. Once a gate is created, it can be applied to any dataset, and the results summarized in a *Summary* tab.

In large experiments, there is often a need for normalization before comparisons can be made across samples. The `Data|Normalization` menu item provides access to several normalization methods. As when creating subsets or transformations, normalization leads to the creation of a new dataset nested within its parent.

A video of *iFlow* demonstrating the above steps is provided as Supplementary Material to this manuscript (see Supplementary Material available online at doi: 10.1155/2009/103839).

3. Discussion

Most FCM software implemented in the Bioconductor project was developed to address the growing need for automation in the data analysis process. However, command-line driven tools exclude a large group of potential users who are more familiar with GUI software. Moreover, in the course of working with high-throughput FCM datasets, it has become apparent that complete automation is not yet a practical goal, and some degree of manual interaction is crucial. We developed *iFlow* in order to make our methods accessible to a broader audience, and to combine the advantages of automated or semiautomated analysis with interactive data analysis.

The *iFlow* package contains all the code necessary to create and run the GUI, but it does not contain any code for the analysis of FCS data. Rather, it relies on functionality implemented in other R packages, which are installed and loaded at the same time as *iFlow*. It currently provides access to data visualization, manual and automated gating, transformations and basic data manipulations. This is sufficient for initial exploratory data inspection, as well as for prototyping large analysis projects.

Some of the capabilities exposed by *iFlow*, such as automated gating, already go beyond what is available in standard FCM GUI software. However, the primary long-term advantage of our software is its open and extensible nature. Additional functionality may easily be added in response to user feedback, or once common use cases have emerged. FCM is a field of active research, and we expect many novel analytical methods to be developed in the future. It would be relatively simple to incorporate these new methods into *iFlow* once they are implemented within the R/Bioconductor framework. This involves modification of the appropriate menu items and association of a particular R command with the extended menu. In this way, the extensive facilites already provided by various R add-on packages can be leveraged to expand the capabilities of *iFlow* with little additional work; for example, FCM data stored in relational databases could easily be imported using existing R packages for database access.

We believe that *iFlow* can serve as a useful interface for advanced statistical processing of FCM data, and that it will help bridge the gap between bench scientists, statisticians, and FCM data analysts. It is available as an R package on Bioconductor (http://www.bioconductor.org/).

Acknowledgments

This work was supported by NIH grant 1R01EB008400. The authors would like to thank Ryan Brinkman and his group, Chaow-Jen Wong and Nishant Gopalakrishnan for their valuable input to help improve early prototypes of our software.

References

[1] Tree Star Inc., “FlowJo,” http://www.owjo.com/.

[2] Verity Software House, “WinList,” http://www.vsh.com/.

[3] K. Lo, R. R. Brinkman, and R. Gottardo, “Automated gating of flow cytometry data via robust model-based clustering,” *Cytometry Part A*, vol. 73A, pp. 321–332, 2008.

[4] M. J. Boedigheimer and J. Ferbas, “Mixture modeling approach to flow cytometry data,” *Cytometry Part A*, vol. 73A, pp. 421–429, 2008.

[5] J. Frelinger, T. B. Kepler, and C. Chan, “Flow: statistics, visualization and informatics for flow cytometry,” *Source Code for Biology and Medicine*, vol. 3, 2008.

[6] R. C. Gentleman, V. J. Carey, D. M. Bates, et al., “Bioconductor: open software development for computational biology and bioinformatics,” *Genome Biology*, vol. 5, p. R80, 2004.

[7] N. Le Meur, F. Hahne, B. Ellis, and P. Haaland, "FlowCore: data structures package for flow cytometry data," *Bioconductor Project*, 2007.

[8] D. Sarkar, N. Le Meur, and R. Gentleman, "Using flowViz to visualize flow cytometry data," *Bioinformatics*, vol. 15–24, no. 6, pp. 878–879, 2008.

[9] The GTK+ Team, 2009, http://www.gtk.org/.

[10] M. Lawrence and D. Temple Lang, "RGtk2: R bindings for Gtk 2.8.0 and above," R package version 2.12.9, http://www.ggobi.org/rgtk2/.

Hindawi Publishing Corporation
Advances in Bioinformatics
Volume 2009, Article ID 356141, 10 pages
doi:10.1155/2009/356141

Research Article

Analysis of High-Throughput Flow Cytometry Data Using plateCore

Errol Strain,[1] Florian Hahne,[2] Ryan R. Brinkman,[3] and Perry Haaland[4]

[1] *FDA-Center for Food Safety and Nutrition, HFS-013 5100 Paint Branch Parkway, College Park, MD 20740, USA*
[2] *Fred Hutchison Cancer Research Center, M2-B514 P.O. Box 19024 Seattle, WA 98109, USA*
[3] *Terry Fox Laboratory, British Columbia Cancer Agency, 675 West 10th Avenue, Vancouver, BC, Canada V5Z 1L3*
[4] *BD Technologies, 21 Davis Dr, Research Triangle Park, NC 27709, USA*

Correspondence should be addressed to Errol Strain, estrain@gmail.com

Received 2 April 2009; Accepted 21 July 2009

Recommended by Matt Wand

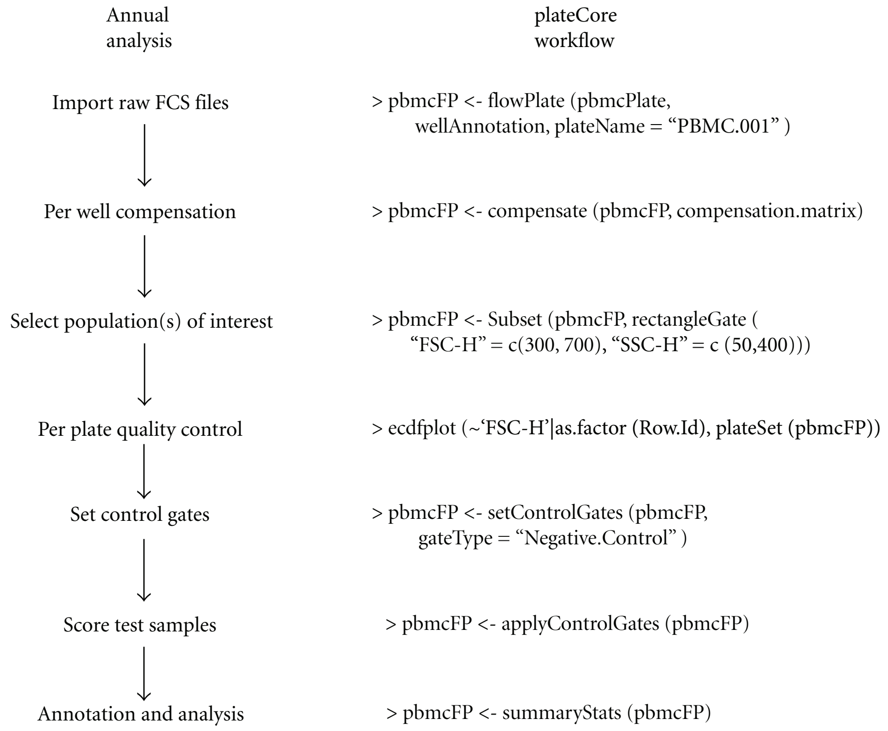
Flow cytometry (FCM) software packages from R/Bioconductor, such as flowCore and flowViz, serve as an open platform for development of new analysis tools and methods. We created plateCore, a new package that extends the functionality in these core packages to enable automated negative control-based gating and make the processing and analysis of plate-based data sets from high-throughput FCM screening experiments easier. plateCore was used to analyze data from a BD FACS CAP screening experiment where five Peripheral Blood Mononucleocyte Cell (PBMC) samples were assayed for 189 different human cell surface markers. This same data set was also manually analyzed by a cytometry expert using the FlowJo data analysis software package (TreeStar, USA). We show that the expression values for markers characterized using the automated approach in plateCore are in good agreement with those from FlowJo, and that using plateCore allows for more reproducible analyses of FCM screening data.

1. Introduction

While there are a number of different software packages available for analysis of FCM data, these programs are often ill-suited to the development of new methods needed for analyzing high-throughput FCM studies. Flow Cytometry-High-Content Screening (FC-HCS) experiments generate large volumes of data [1, 2], which requires a systematic approach to preprocessing, gating (i.e., filtering), and summarizing results for robust analyses. Current FC-HCS data analysis methods often use a combination of software packages for different parts of the analysis. The raw FCM files are processed and gated using FCM specific software, such as FlowJo or FCS Express (De Novo Software, USA). Results are then exported, and statistical analysis is performed in packages like MATLAB (USA) and R (http://www.r-project.org/) [3]. Unfortunately, this approach to FC-HCS analysis results in methods that are semiautomated at best, and they often require significant subjective and error-prone manual intervention to identify cells of interest [4]. It is therefore desirable to develop programmatic approaches to process FCM data so that FC-HCS analysis pipelines are robust, objective, and able to match the high-throughput capacity of modern cytometers.

FCM packages available through the Bioconductor [3] project provide an open platform that can be used by cytometrists, bioinformaticians, and statisticians to collaboratively develop new methods for automated FC-HCS analysis. The basic data processing tools for importing, transforming, gating, and organizing raw FCM data are in the flowCore package [5] and the visualization functions are in flowViz [6]. The Bioconductor model for FCM data analysis facilitates the development of new analysis methods, since the overhead associated with accessing and visualizing FCM data is handled by flowCore and flowViz. The availability of flowCore and flowViz has enabled the creation of new tools for quality assessment of large FCM experiments, such as flowQ [7], and for model-based clustering and automated gating, such as flowClust [8].

We have developed an R package (plateCore) that also takes advantage of the functionality in flowCore and flowViz to create methods and data structures for processing



```
> pbmcFP <- flowPlate (pbmcPlate,
     wellAnnotation, plateName = “PBMC.001” )
> pbmcFP <- compensate (pbmcFP, compensation.matrix)
> pbmcFP <- Subset (pbmcFP, rectangleGate (
     “FSC-H” = c(300, 700), “SSC-H” = c (50,400)))
> ecdfplot (~‘FSC-H’|as.factor (Row.Id), plateSet (pbmcFP))
> pbmcFP <- setControlGates (pbmcFP,
     gateType = “Negative.Control” )
> pbmcFP <- applyControlGates (pbmcFP)
> pbmcFP <- summaryStats (pbmcFP)
```

Figure 1: Typical FC-HCS plate workflow on the left and corresponding steps from a PBMC lymphocyte plateCore analysis on the right.

large, plate-based FCM data sets. Additionally, we have implemented new tools to make it easier to integrate textual descriptions of plate layouts and also to perform automated gating based on nonparametric analysis of negative control wells. This study presents results from an automated plateCore analysis of a PBMC lymphocyte BD FACS CAP (Combinational Antibody Profile) data set, which included 189 different antibody-dye conjugates and their controls arranged on 5 replicate 96-well plates. The output of plateCore was compared to an analysis by an expert cytometrist using FlowJo, one of the standard FCM analysis programs, to evaluate the performance of the automated approach.

plateCore is not designed to be a graphical user interface driven tool, but rather to help develop a standardized platform for the analysis of FC-HCS data. These analyses often represent a collaborative effort between cytometry experts who generate the data and the quantitative individuals who help deal with the large volume information. In order for this collaboration to work, the cytometrists must have confidence in the results of the automated analysis. To this point, we demonstrate the equivalence of our results to those produced by an expert cytometrist using FlowJo.

2. Materials and Methods

2.1. Flow Cytometry Data. The data analyzed in this study was part of the initial set of experiments used to validate the BD FACS CAP platform. BD FACS CAP was designed as a cell characterization tool to screen for the presence of a large number of different human cell surface markers, and it was important to show that the assay was able to correctly identify positive and negatively staining markers on a well-studied cell population, such as PBMC lymphocytes. Previously frozen PBMC samples from two donors were analyzed on a BD FACS Calibur using BD FACS CAP staining plates. The analysis was performed on 96-well plates with 189 different antibodies arrayed three per well in 63 test wells, along with 30 isotype control wells and three unstained controls. The complete list of BD FACS CAP antibodies can be found at http://www.bd.com/technologies/discovery_platform/BD_FACS_CAP.asp. FCM files for the five plates (two for Donor 1 and three for Donor 2) are available for download from http://www.ficcs.org/data/plateData.tar.gz.

2.2. Data Analysis. FCM output was analyzed in parallel using FlowJo and plateCore. Short descriptions of the steps in each software package are provided below. Additionally, the plateCore script used to perform the analysis is provided in Supplementary Materials available online at doi: 10.1155/2009/356141, and an example of the progression from raw FCM data files to a completed plateCore analysis for a single plate is shown in Figure 1.

2.3. plateCore

(1) Template Construction. A tab delimited text file was created that describes the contents of each well on the replicate plates. This information includes the marker name, fluorophore, antibody type, and the isotype group assignment. In this early version of BD FACS CAP the combination of antibodies in a well was based on available

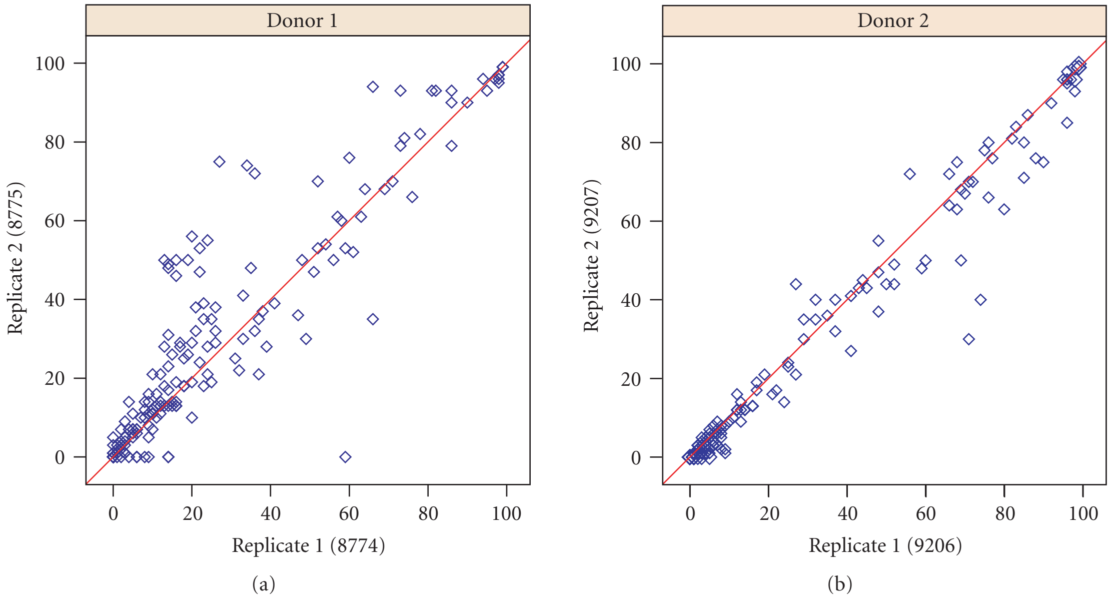


Figure 2: FlowJo estimates for the percentage of cells above the isotype threshold for 189 markers on replicate plates for donor 1 and donor 2. Estimates from markers where the center of the cell population was near the isotype threshold, around 50%, were more variable than samples which were clearly positive (≥99%) or negative (≤1%). The correlation for replicate plates was strong in both donors, with donor 1 at 0.92 and donor 2 at 0.98. Plate 9208 for donor 2 is not shown, since the results are very similar to 9206 and 9207.

antibody-dye combinations. Newer versions of BD FACS CAP use biological information to assign markers to wells and are able extract more useful coexpression information.

(2) Data Import. FCM files for each plate were imported using flowCore. The import operation produces 5 flowSet objects, one for each plate, which were then integrated with the layout information in the template to create 5 flowPlates.

(3) Gating. flowPlates were processed using a combination of static gates (rectangleGate) and data driven gates (using norm2filter in flowCore) to pick out the lymphocytes in the forward (FSC) and side scatter (SSC) channels.

(4) Plate Level Quality Assessment. The quality of the data was then assessed by looking for fluidic events such as bubbles, pressure drops, or large aggregates that can shift the baseline fluorescence readings. Fluidic events can often be identified by plotting the empirical cumulative distribution function (ecdf) plots of FSC values for each well and looking for distributions shifted relative to other wells [9]. Based on the ecdf plots, several wells were further investigated by cytometry experts who determined that the shifts were in an acceptable range.

(5) Isotype-Based Gating. The threshold between positive and negative cells was determined using the isotype controls, which provided a gross estimate of nonspecific binding in the primary antibodies. One-dimensional gates were created using the isotype thresholds, and these gates were applied to identify cells that had specific staining in channels of interest. Details about the nonparametric isotype gating strategy implemented in plateCore are provided in the results section.

(6) Summarization. The 5 flowPlates were then aggregated into a single flowPlate using the fpbind operation from plateCore. Having the data in this format makes it easier to plot replicate wells from different plates, perform statistical analyses, and to export a single, experiment level results text file.

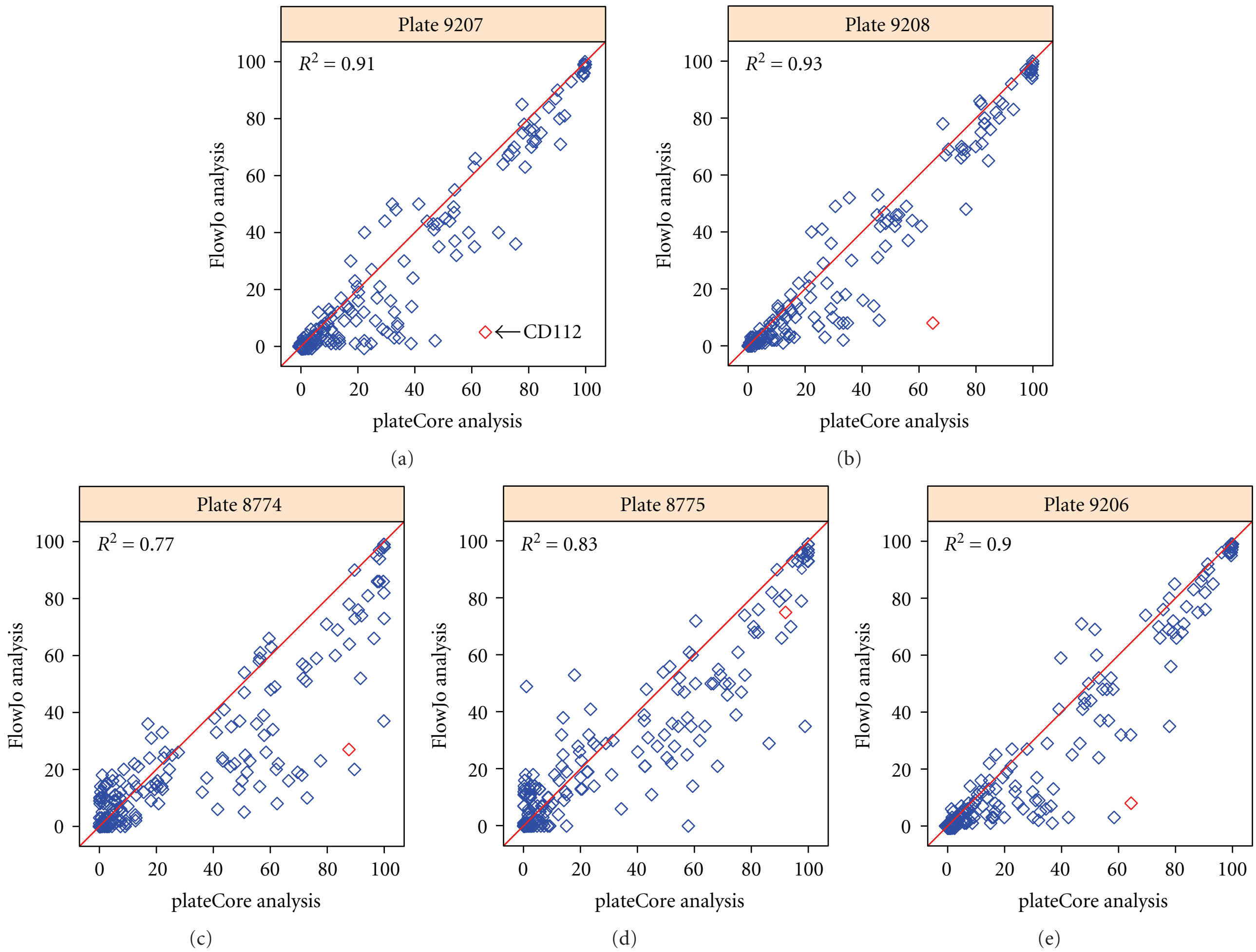
2.4. FlowJo

(1) Template Construction. An XML-based FlowJo template was created where test wells and their corresponding isotype control well were assigned to one of 30 groups. Wells in each group contained similar sets of antibody-dye conjugates.

(2) Data Import. FCM files were imported using the FlowJo template.

(3) Gating. Lymphocytes were selected using polygonal gates in the FSC-SSC view.

(4) Plate Level Quality Assessment. Quality assessment was performed by looking for wells where the FSC-SSC location of the lymphocyte population shifted relative to other wells on a plate.



(a)

(b)

(c)

(d)

(e)

Figure 3: Plot showing the percentage of cells above the isotype threshold from plateCore (x-axis) and FlowJo (y-axis) for each of the 189 markers on the 5 PBMC plates. If the two methods produce similar estimates, then the values should be near the red line ($y = x$). In plateCore the isotype threshold was determined using only information from the isotype control well, while the threshold in FlowJo may be adjusted after identifying either positively or negatively staining test samples. Generally, these FlowJo adjustments resulted in the isotype gate being set a higher level to exclude a negative test sample. The effect of increasing the isotype threshold can be seen in these plots, where most disagreements are cases where plateCore estimates are higher than FlowJo. Detailed plots for one marker, CD112 (red diamond), where the two methods give different results are shown in Figure 5.

(5) Isotype-Based Gating. Event data for isotype wells was visualized on a log scale, and the expression threshold for each stained channel was set by picking a value that lies above the bulk of the events. Isotype gates were initially set so that approximately 0.5% of the events in the isotype well were above the threshold. These gates were then applied to the test wells, and the gates were moved up or down depending upon positive and negative test well populations. If the population of cells in positive wells was much higher than the isotype gate, then the gate was moved up to help reduce false positives associated with nonspecific staining. Similarly, if the isotype gate was higher than negative samples, the gate would be moved down to ensure that positive cells were classified correctly.

(6) Summarization. The percentage of cells above the threshold for each of the 189 antibodies was then exported for each plate, and these results were merged to create the analysis report.

3. Results

Although this study focuses on comparing two different FC-HCS analysis methods, it is important to consider the original goal of the experiment used to generate the data when interpreting the results. BD FACS CAP was designed to provide a standard assay platform for screening a large number of markers on many different cell types. The validation effort for BD FACS CAP included running the assay on well-characterized cell types to find markers with either positive or negative staining and comparing these results to published cell expression profiles in literature. The PBMC lymphocyte staining results presented in the

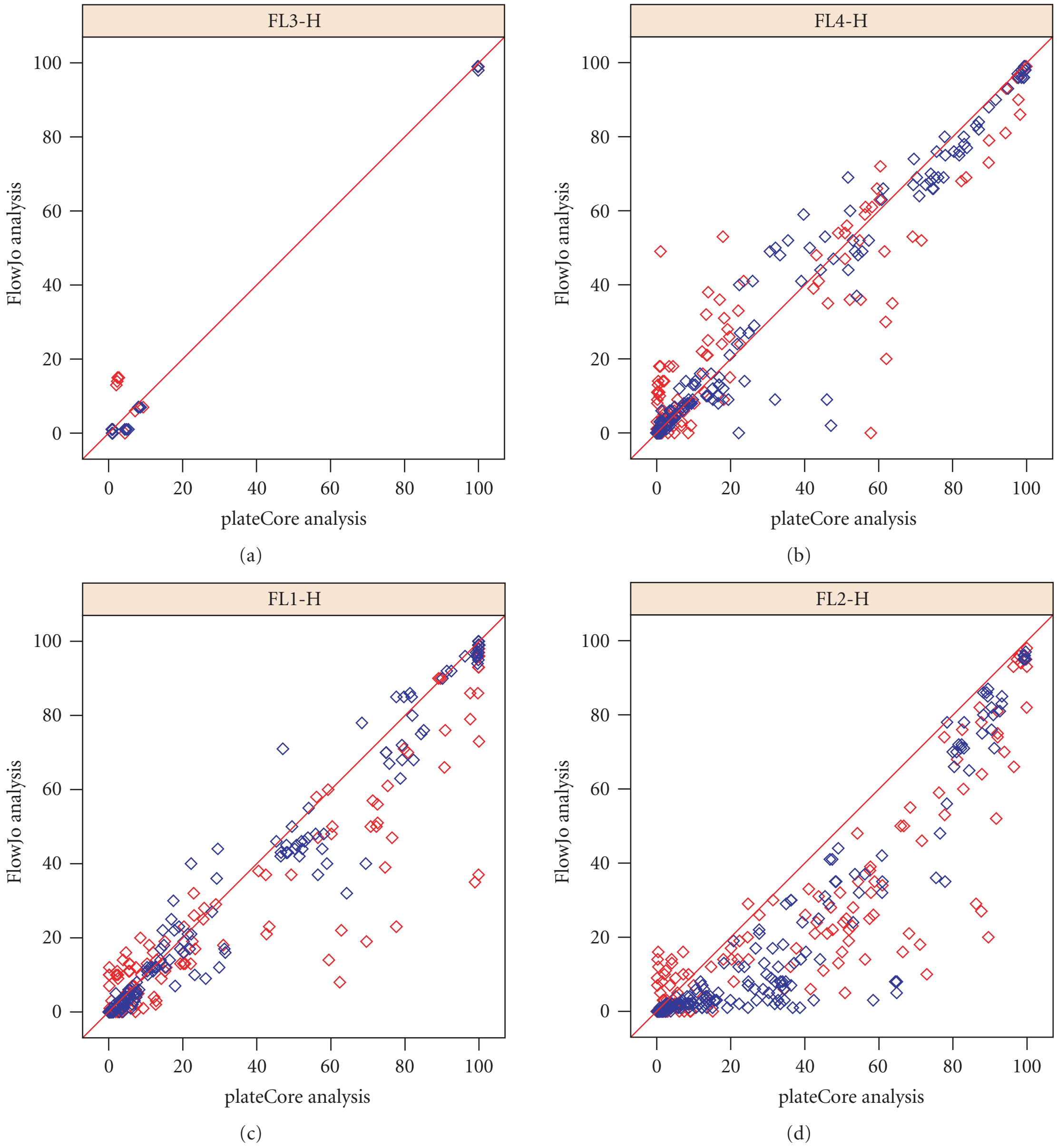
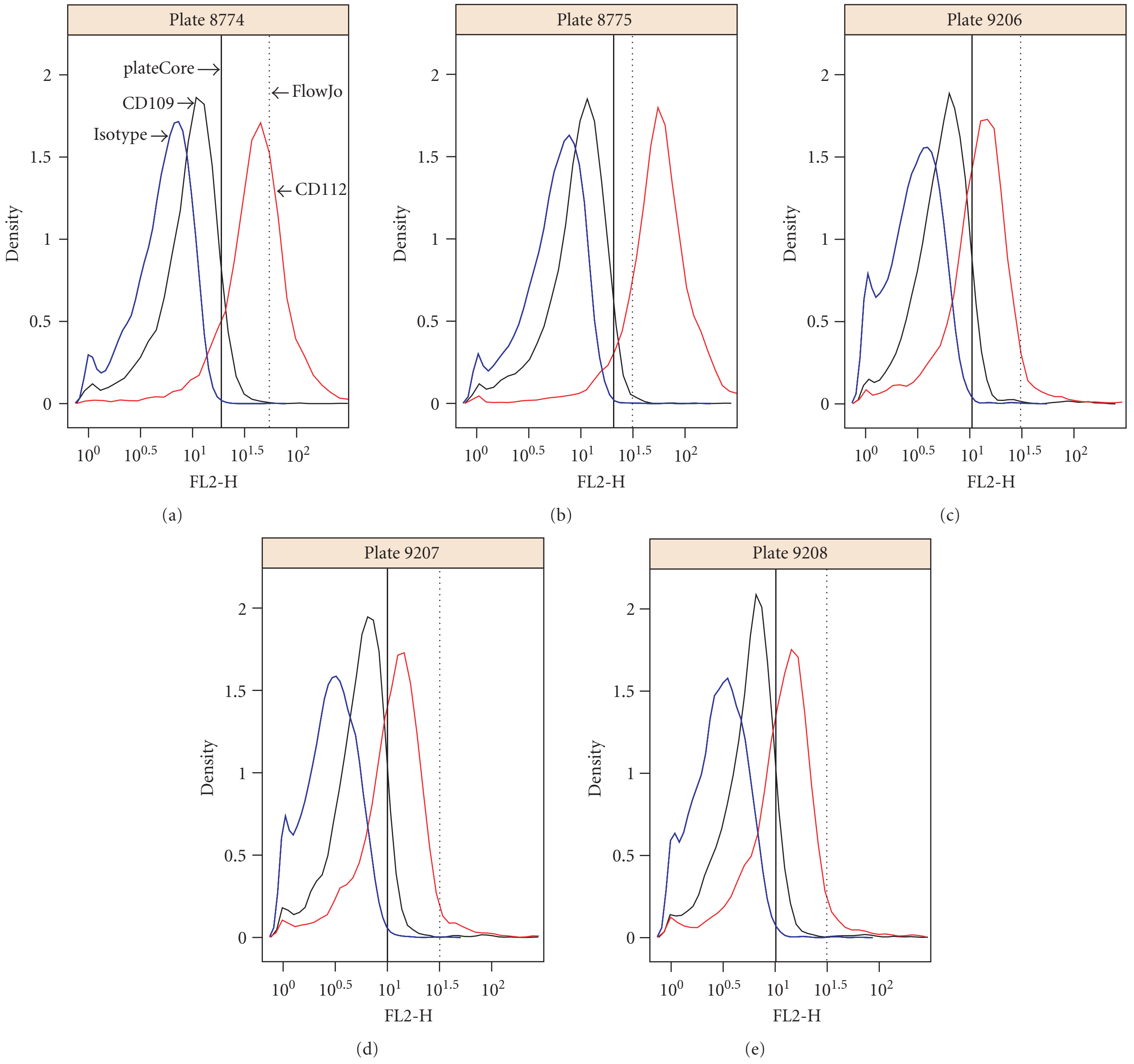


Figure 4: Plot showing the percentage of cells above the isotype threshold from plateCore (x-axis) and FlowJo (y-axis) for donor 1 (red) and 2 (blue) in channels FL1-H through FL4-H. plateCore gating for Phycoerythrin (PE) conjugated antibodies (FL2-H) was consistently lower than FlowJo, resulting in more cells above the isotype gate.

following section represent one of the cell types used for validating the technology.

3.1. FlowJo Output. Descriptions of marker expression profiles for particular cell populations in flow cytometry often use terms like positive-negative, or bright-dim, to qualify the amount of target present. Since BD FACS CAP is a standard platform for screening a wide range of cell types, and antibody concentrations were not optimized for these particular PMBC samples, results are reported as the percentage of cells above the isotype gate rather than positive or negative. Followup studies, including single color titrations and competition experiments, are needed to definitively show that a marker is present. Markers that have been previously characterized using BD FACS CAP with ≥90% of the cells above the isotype threshold are usually confirmed as positive using titration and competition experiments, while staining in markers with ≤10% of cells above the isotype threshold is often the result of nonspecific binding (data not shown). Note that these percentages refer to the fraction of cells above the isotype threshold, but this does not necessarily imply heterogeneous staining in multiple populations.

Automating the creation and modification of isotype gates made by cytometrists analyzing BD FACS CAP data using FlowJo is challenging. Cytometrists adjust gates based



(a)

(b)

(c)

(d)

(e)

Figure 5: Density plots showing the plateCore (solid black) and FlowJo (dashed black) isotype gates for CD112 and CD109, which shared the same isotype control (IgG1-PE). The plateCore and FlowJo analyses gave different estimates for CD112 (see Figure 3), which was caused by the gate being moved higher in FlowJo based on the presumed negative staining for CD109.

on expert knowledge about the performance of specific antibody types and dyes, or after identifying positive or negative test samples. If the isotype gate cut off the bottom portion of a positive cell population in a test well, then the gate was moved down. Similarly, if the isotype gate included too many cells from negative test wells, it was moved up. Results from the FlowJo-based gating of replicate PBMC plates are shown in Figure 2. Detailed results for each marker are not presented in this study, but since the majority of antibodies on the BD FACS CAP staining plate are known to bind different leukocytes, it is not surprising that a large fraction would be identified as positive on PBMCs. Markers such as CD44, CD45, CD47, and CD59 are broadly expressed on lymphocytes and were positive (>99%) in this study.

3.2. plateCore versus FlowJo. Isotype controls are used to determine the threshold between background staining and specific binding of an antibody conjugate to its target. For the FlowJo analysis, the gate was initially set at the 99.5th quantile of the fluorescence signal in each stained channel of the isotype and then adjusted based on results from test wells. In plateCore, we have implemented two approaches to automatically creating gates based on negative controls. The first simply replicates the initial creation of the FlowJo gates and determines the threshold based on a set quantile, while the second uses a nonparametric approach where the gate (G_{ij}) for isotype i, channel j was set according to

$$G_{ij} = \mathrm{MFI}_{ij} + 4\mathrm{MAD}_{ij}, \quad (1)$$

where MFI is the Median Fluorescence Intensity and MAD is Median Absolute Deviation in the raw data (linear scale). Although FCM fluorescence signals are approximately lognormal, as evident from density plots shown in this study (Figures 5 and 8), it is difficult to reliably make distributional assumptions, and the choice of 4 MADS represents a conservative attempt to set the gate above the 99th quantile of cells in the isotype stained wells.

The nonparametric gating approach is obviously more robust to outliers than a static gate based on the 99.5th quantile, but in practice both methods produce very similar results if the data is good quality and there are a sufficient number of cells (over 1000) in the isotype well. The plateCore analysis presented in this study used the nonparametric approach to gating, and while this relatively simple method works surprisingly well for BD FACS CAP, advances in model-based clustering methods, such as those in flowClust, should lead to future performance improvements in automated gating.

Comparisons of the output from the plateCore and FlowJo analyses are shown in Figure 3. Both methods produce nearly identical estimates for markers that were either clearly positive (≥99%) or clearly negative (≤1%), and R-squared values for all makers were between 0.83 and 0.93 (Figure 3). These cell populations are not close to the isotype threshold, and therefore different isotype gate settings have little or no effect on estimates of the percentage of cells above the gate. In situations where the isotype gate splits a test cell population, small changes to the gate can dramatically change these estimates. This effect is evident in the results from replicate plates using FlowJo (Figure 2) and in comparisons of FlowJo and plateCore (Figure 3), where estimates for markers having approximately 50% of the cells above the isotype gate are more variable than markers having ≤1% or ≥99%.

Figure 4 shows the plateCore and FlowJo comparison broken down by channel, and we can see that a large portion of the markers that disagree were stained with Phycoerythrin (PE) in FL2-H. plateCore estimates for antibodies conjugated to PE were almost always higher than FlowJo, indicating that the isotype gates in FlowJo were moved above their initial setting. Looking in detail at one PE conjugate where the two methods disagree, CD112 IgG1-PE, we can see how the gate for was changed in the manual analysis based on what looks like nonspecific staining in a related test sample, CD109 IgG1-PE (Figure 5). Since the gene for CD112 (PVRL2) has been shown to be expressed on a subset of lymphocytes in healthy donors using microarrays [10], the plateCore results showing 65%–92% of the cells above the isotype gate may actually represent specific staining. Unfortunately, increasing the isotype (IGg1-PE) threshold in FlowJo to eliminate what looks like background staining in CD109 also seems reasonable. More focused studies will have to be performed to determine if the staining for CD112, and other markers that disagreed, was positive or negative.

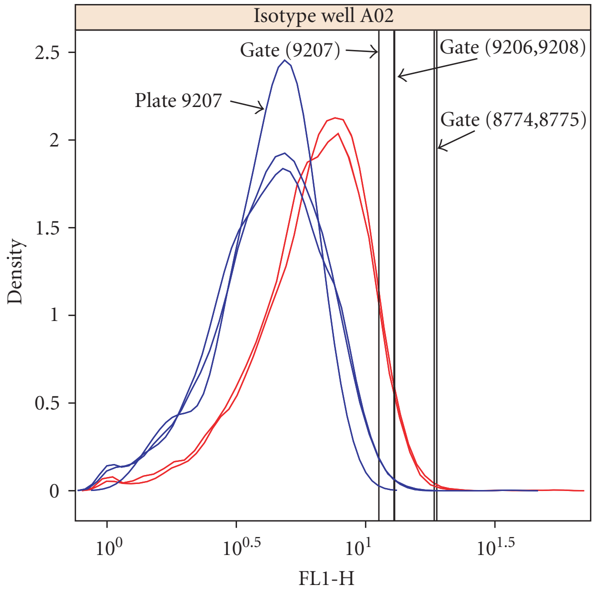


Figure 6: Density plot showing an example of one case where the isotype (IgG1-Alexa 488) gate settings differed between replicate plates for donor 2 (blue). In this case, the low setting for plate 9207 did not result in a significant difference between plates for the percentage of cells above the gate in the corresponding test well (CXCR5), so the gate was not modified. Plates 9206, 9207, and 9208 had 14%, 16%, and 15% percent of cells above the gate, respectively.

3.3. Gating Quality Assessment. Since we may not always have access to output from expert cytometrists to help determine if our automated gating is reasonable, we need alternative approaches to assessing the quality of our isotype-based gates. The strategy we used for this PBMC study involves visually checking density plots of the isotype wells for replicate plates and also comparing the percentage of cells above the isotype gates versus the MFI ratio to see if the gating was consistent across the experiment. Plates for each PBMC donor are purely technical replicates; so any differences should be due to variation in cell staining or changes in instrument settings.

An example of the plots used to check replicate isotype gates is shown in Figure 6. In this case the threshold for one of the 3 replicate plates for donor 2 was lower than the other 2, indicating that the marker expression values from this isotype should be further evaluated. Fortunately, the difference is relatively small and did not change the estimate for the test well associated to this isotype (CXCR5 IgG1-Alexa 488). If the difference between replicates had been larger, we would have averaged the isotype thresholds from the remaining replicates and replaced the setting for plate 9207.

The MFI ratio is defined as the ratio of the MFI for a marker to the MFI of its isotype control. Essentially, this ratio tells us how well separated a population of stained test cells is from the population of cells in the isotype control. The distance between these two populations is related to the percentage of cells above the isotype gate (Figure 7). To evaluate isotype gating at the experiment level for these

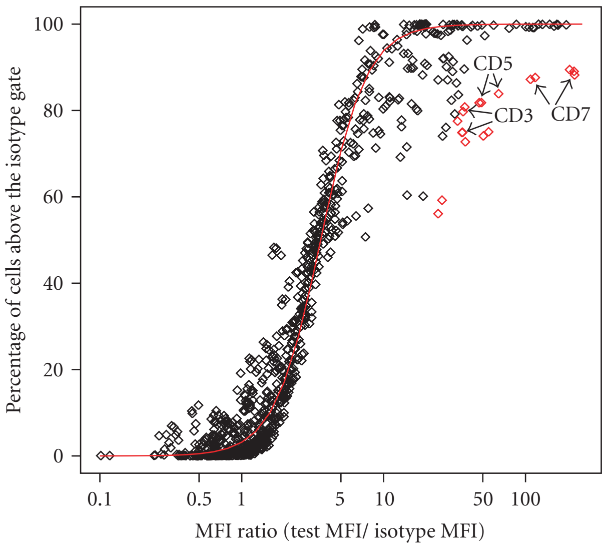


Figure 7: Quality of the automated gating was assessed by performing a robust logistic regression of the percentage of cells above the isotype gate on the log transformed MFI ratio and looking for estimates that were more than 2 standardized residuals away from the best fit line (red line). There were 18 estimates flagged in this study (red diamonds) where the value was different than we would predict from the MFI ratio. Detailed examination of these 18 cases showed that the isotype gate settings were reasonable, but they differed from other markers in that they had more than one population of stained cells. Sample density plots for one of these markers, CD3, are provided in Figure 8.

5 plates we performed a robust logistic regression for the percentage of positive cells on the MFI ratio and looked for values that were more than 2 standard residuals from the best fit line. We chose 2 standard residuals in a conservative attempt to ensure that any questionable automated gating decisions were examined in detail. Deviation from the best fit line can indicate either a problem with the isotype gate or that the sample has multiple cell populations (Figure 8). If the percentage of cells above the gate is significantly different than we would predict from the MFI ratio, then the isotype gate was checked. We note that this approach does not actually tell us if the gating was correct, simply whether or not the isotype gating was consistent.

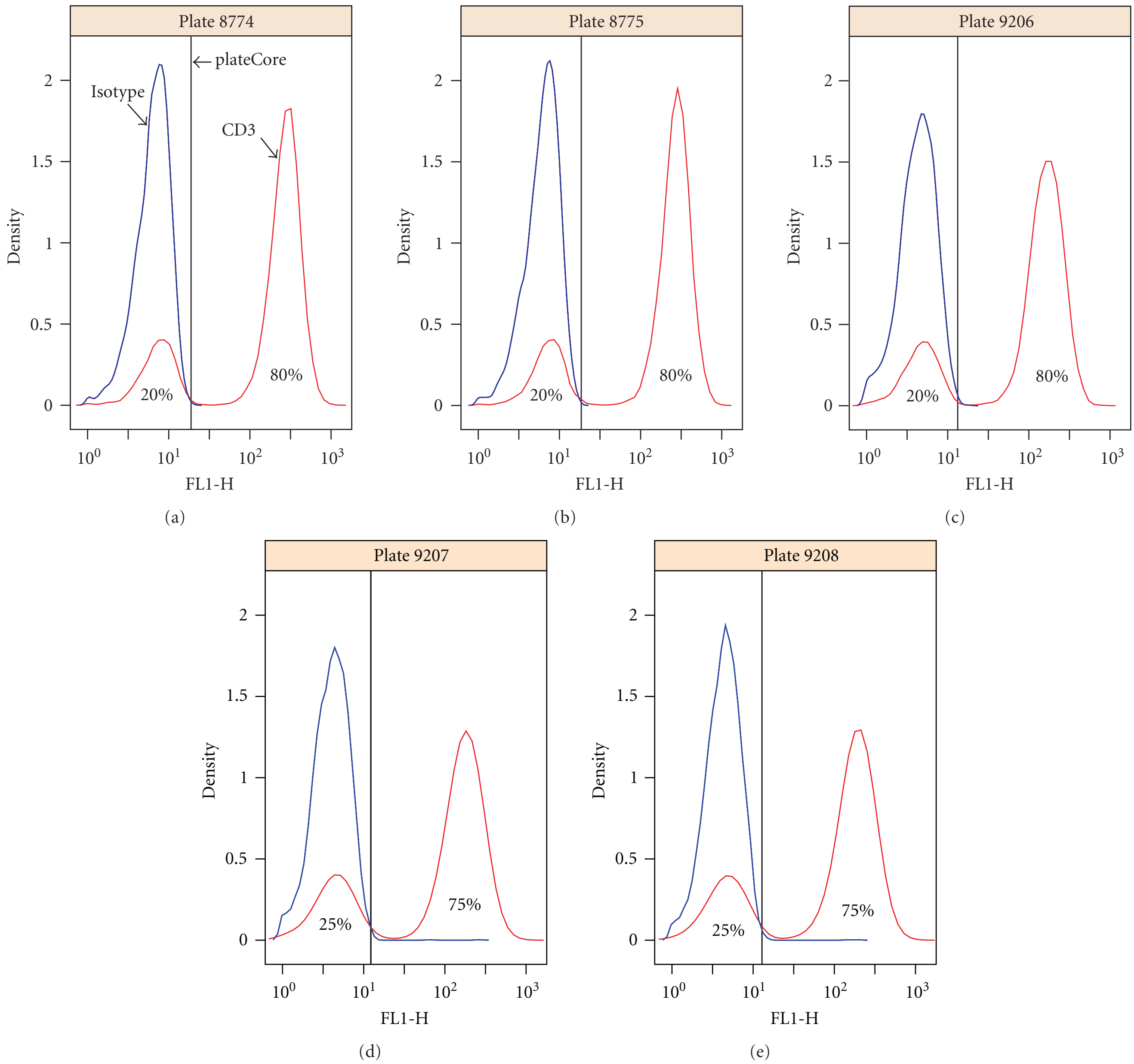
The bulk of the measured responses for the markers (927 out of 945) is within two standard residuals from the best fit line (Figure 7), which is surprising since the 189 different antibodies were conjugated to different fluorophores (either Alexa 488, FITC, PE, PerCP, APC, or Alexa 647) and matched against different isotypes (either IgG1, IgG2, IgG2a, IgG2b, IgG3, or IgM). We expected that differences in fluorescence intensity between dyes, and variation in nonspecific binding by different antibody types, would make direct comparisons difficult. The 18 values that were more than two standard deviations away from the line were examined in detail, and the isotype gate settings were found to be reasonable. In this case the flagging was the result of a positive and negative staining population of cells, which made the relationship between the MFI ratio and the fraction of cells above the isotype gate look very different than markers staining a single population. Density plots for one of the flagged markers, CD3, are shown in Figure 8.

4. Discussion

We were motivated to use the flowCore package for BD FACS CAP data analysis by a desire to reduce subjectivity associated with isotype gating and also to make the more analyses more reproducible. We found that while flowCore was very powerful, both in terms of efficient use of memory for large data sets and an extensive collection of FCM functions, it did not scale well to BD FACS CAP experiments with multiple plates and a complex layout. plateCore was developed to make it easier to perform operations and produce visualizations that are technically challenging to do in flowCore and flowViz. For example, creating a set of threshold gates based on negative control wells, either isotype or unstimulated cells, and then applying those gates to test wells on a plate is a relatively common FC-HCS operation. In this study, the PBMC isotype gates were created and applied to test wells in two steps, using setControlGates and applyControlGates (Figure 1). Replicating this same operation in flowCore would require either many individual custom gating steps or users to develop their own methods that duplicate the functionality in plateCore.

plateCore provided the ability to quickly analyze complex BD FACS CAP plates and produce useful visualizations (such as Figures 2–8), which facilitated discussions with the cytometry experts and helped to develop approaches to automate the gating process. Since this was a screening assay, the goal was to quickly and reproducibly process a large volume of data to get an approximate expression value for each of the 189 human cell surface markers and then perform more in-depth analysis for markers that were of biological interest. Using plateCore, we were able to reduce the level subjectivity in setting isotype gates, eliminate mistakes associated with manual data annotation and export, and automate the creation of plots and data quality reports that summarized the experiment. Additionally, the plateCore scripts and experimental annotation can be shared with other cytometry groups, allowing them to reproduce our analysis.

An important realization from our experience developing plateCore and analyzing BD FACS CAP experiments was that individual isotype gates should not be changed by cytometrists when performing FC-HCS experiments. The cytometrist does not have any information other than expert opinion about where a gate should go for a particular set of values, and making adjustments adds both bias and noise to the end result. In addition, the use of a more uniform gating approach facilitates the use of plateCore to combine and analyze results across many samples, which is one of the important new capabilities of this software. The functionality in plateCore enables cytometrists and statisticians to work together and make higher level decisions about gating strategies, based on methods like the gating quality assessment shown in Figure 7. Also, the gating in



(a) (b) (c) (d) (e)

Figure 8: Density plot for CD3 (IgG1-Alexa 488), which was flagged for further evaluation by our gating quality assessment (Figure 7). The isotype gate settings look reasonable; however the MFI ratio for CD3 was very different from other markers that also had 75%–80% of their cells above the isotype gate. Looking at Figure 7, other markers with 75%–80% had MFI ratios near 5, while CD3 has an MFI ratio of 31–37. The flagging was the result of 2 cell populations for CD3, whereas most other markers stain a single population.

this experiment is relatively simple since we were only concerned with one dimension at a time. Developing new methods to reproducibly gate samples in three or more dimensions requires tools like flowCore and flowClust. plateCore provides infrastructure that makes the data available to quantitative scientists to further develop and apply these research tools.

The complexity of large FCM experiments, like BD FACS CAP, highlights the difficulty of applying existing FCM analysis platforms to high-throughput studies. Generating and interpreting results from this PBMC study required extensive collaboration between flow cytometrists, bioinformaticians, and statisticians. At various points in the analysis, each group needed to access the raw data, annotation, and details about the experimental design. Providing this access using standalone FCM platforms is expensive in terms of the price of multiple software licenses and in time spent training statisticians and bioinformaticians to use the programs. Fortunately the Bioconductor FCM packages are modeled on standard data structures used for microarrays, which should already be familiar to most quantitative individuals working on high-throughput biological problems. In addition, this approach allows scientists to use modern software development tools, including version control software, to manage plateCore scripts and make the analysis reproducible in a way that is generally not possible with GUI-based tools. Finally, we found that flowCore, flowViz, and plateCore provide an open analysis platform that facilitates communication between the flow cytometrists generating the data and the computational experts analyzing the data.

Acknowledgments

The authors would like to thank Megan Gottlieb, Ruiling Yuan, and Ling Wang for collecting and sharing the data and for performing the FlowJo analysis and sharing the results with them in a compatible format. Their patience and expertise were critical to the development of plateCore. Support for this research was provided by Grant no. EB005034 from the National Institute of Biomedical Imaging and Bioengineering and by the Michael Smith Foundation for Health Resarch.

References

[1] M. Gasparetto, T. Gentry, S. Sebti, et al., "Identification of compounds that enhance the anti-lymphoma activity of rituximab using flow cytometric high-content screening," *Journal of Immunological Methods*, vol. 292, no. 1-2, pp. 59–71, 2004.

[2] R. R. Brinkman, M. Gasparetto, S. Lee, et al., "High-content flow cytometry and temporal data analysis for defining a cellular signature of graft-versus-host disease," *Biology of Blood and Marrow Transplantation*, vol. 13, no. 6, pp. 691–700, 2007.

[3] R. Gentleman, V. J. Carey, D. M. Bates, et al., "Bioconductor: open software development for computational biology and bioinformatics," *Genome Biology*, vol. 5, no. 10, article R80, 2004.

[4] H. T. Maecker, A. Rinfret, P. D'Souza, et al., "Standardization of cytokine flow cytometry assays," *BMC Immunology*, vol. 6, article 13, 2005.

[5] F. Hahne, N. Le Meur, R. R. Brinkman, et al., "flowCore a bioconductor package for high throughput flow cytometry," *BMC Bioinformatics*, vol. 10, no. 1, article 106, 2009.

[6] D. Sarkar, N. Le Meur, and R. Gentleman, "Using flowViz to visualize flow cytometry data," *Bioinformatics*, vol. 24, no. 6, pp. 878–879, 2008.

[7] R. Gentleman, F. Hahne, J. Kettman, and N. Le Meur, "Bioconductor package flowQ," http://www.bioconductor.org/.

[8] K. Lo, F. Hahne, R. R. Brinkman, and R. Gottardo, "flowClust: a bioconductor package for automated gating of flow cytometry data," *BMC Bioinformatics*, vol. 10, no. 1, article 145, 2009.

[9] N. Le Meur, A. Rossini, M. Gasparetto, C. Smith, R. R. Brinkman, and R. Gentleman, "Data quality assessment of ungated flow cytometry data in high throughput experiments," *Cytometry A*, vol. 71, no. 6, pp. 393–403, 2007.

[10] R. J. Critchley-Thorne, N. Yan, S. Nacu, J. Weber, S. P. Holmes, and P. P. Lee, "Down-regulation of the interferon signaling pathway in T lymphocytes from patients with metastatic melanoma," *PLoS Medicine*, vol. 4, no. 5, article e176, 2007.