Comparing and Clustering Flow Cytometry Data*

Lin Liu, Li Xiong, James J. Lu
Department of Math/CS
Emory University
lliu24, lxiong, jlu@mathcs.emory.edu

Kim M. Gernert
BimCore
Emory University
gernert@emory.edu

Vicki Hertzberg
Department of Biostatistics
Emory University
vhertzb@sph.emory.edu

Abstract

Flow cytometry technique produces large, multi-dimensional datasets of properties of individual cells that are helpful for biomedical science and clinical research. This paper explores an approach for comparing and clustering flow cytometry data. To overcome challenges posed by the irregularities and the high dimensions of the data, we develop a set of data preprocessing techniques to facilitate effective clustering of flow cytometry data files. We present a set of experiments using real data from the Protective Immunity Project (PIP) showing the effectiveness of the approach.

1. Introduction

Flow Cytometry (FCM) [7] is a technique used in clinical research for studying the immunological status of patients with vaccines or other immunotherapies, for characterizing cancer, HIV/AIDS infection and other diseases, as well as for research and therapy involving stem cell manipulations. The technique measures the characteristics of single cells, determined by visible and fluorescent light emissions from the markers on the cells. As the liquid flow moves the suspended, labeled cells past a laser that emits light at a particular wavelength, the specific markers attached to the cell fluoresce. The fluorescence emission from each cell is collected, and subsequent electrical events are analyzed on a computer that assigns a fluorescence intensity value to each signal in Flow Cytometry Standard (FCS) data files. Each FCS data file thus consists of multi-parametric descriptions of thousands to millions of individual cells.

How such large sets of data points in a highly multidimensional space can be efficiently and systematically analyzed represents a basic yet important challenge. The classical method of analyzing cytometry data files is to plot the different combinations of channels (parameters) two at a time in a 2D scatter plot and then select subgroups of cells using *gates*. The gates are regions that can have any shape, but usually are rectangular. The cells within the gate are included for further analysis and viewed in another 2D scatter plot with different axis, i.e. other channels. Comparisons between different cytometry datafiles often depends on human inspection of such 2D visualizations of all the different combinations of channels [4]. The major disadvantage of this method is that it is not only tedious, it can also miss potential subgroups of cells due to projection of higher dimensional data down to two dimensional spaces which makes them indiscernible as separate clusters. In addition, the shape, size and position of the gate are largely dependent on the experience and expectations of the researcher.

Recently, some clustering algorithms and tools have been developed for FCM data that cluster cells into cell groups based on their intensity patterns using all dimensions (channels) at once [11, 8, 1]. However, comparison between different cytometry datafiles still remains a challenge as it is not straightforward how the clusters of cells can be directly compared across different data files.

In this paper, we explore an approach for comparing and clustering FCS datafiles or samples. Such sample-based clustering presents a number of challenges due to the *high dimensions* and *irregularities* of the data. First, while there may be only tens to hundreds of samples available, the space of potential features consists of thousands to millions of cell intensity data values at multiple channels. This induces an extraordinarily large search space for the parameters of the model. Second, the cells are not ordered uniformly across samples; they may be in any random order. This makes feature modeling a challenge as the data (cell intensity values) are not directly comparable across samples because of the unknown order of the cells.

To address the above challenges, we developed a set of data preprocessing techniques to facilitate effective clustering of FCS data files. The key is to summarize each FCS data file in a way so that they can be compared and clustered

*The research is partially supported by the Protective Immunity Project through the NIH grant NO1-AI-50025.

Table 1. Raw FCS Data (Cell Intensity Values)

	FSC-A	FSC-H	Comp-PE-A	SSC-H	Comp-APC-A	Comp-FITC-A	SSC-A	Comp-PerCP-A	Comp-PE CY-7-A	Comp-Pacific Blue-A
Cell 1	634	547	1381	258	2602	2946	204	944	1300	1956
Cell 2	393	319	1465	904	2017	1886	1130	1252	2059	2056
Cell 3	634	537	1231	1092	2353	2342	1312	1626	2286	2264
...	...	...	...	...	...	...	...	...	...	...

Table 2. Feature Data (Cell Counts for Given Intensity Values)

	FSC-A	FSC-H	Comp-PE-A	SSC-H	Comp-APC-A	Comp-FITC-A	SSC-A	Comp-PerCP-A	Comp-PE CY-7-A	Comp-Pacific Blue-A
...	...	...	...	...	...	...	...	...	...	...
150	1	333	0	156	0	0	137	0	7	0
151	0	290	4	159	0	0	113	0	7	0
152	0	275	0	188	0	0	123	1	5	0
...	...	...	...	...	...	...	...	...	...	...

effectively and efficiently. Specific contributions are as follows. First, we model the features by converting the original cell intensity values to cell-intensity distribution (cell counts for each intensity value) at each channel so that they are comparable across samples. Second, we perform data reduction through regression analysis to reduce the number of features significantly which allows effective and efficient clustering of the data files. Finally, we evaluate the approach using a set of real data, from the Protective Immunity Project, to demonstrate the effectiveness of our approach. In particular, we show that the feature reduction significantly improves the clustering analysis both in quality and efficiency.

The rest of the paper is organized as follows. Section 2 describes the flow cytometry datasets and the methodology used in our study. Section 3 presents our experimental results. Section 4 presents a brief review of related work. Section 5 concludes the paper with a brief summary and a discussion of future directions of our research.

2. Methodology

In this section, we describe the flow cytometry data files and present our methodology. Our goal is to cluster a set of samples based on their raw FCS files containing thousands to millions of cell-intensity values. We first model the features by encoding the cell-intensity values into cell-intensity distribution values so that they are comparable across samples. We then perform data reduction through regression analysis to reduce the large number of features. We then cluster the samples using the original features and the reduced features respectively. For evaluation, we measure the tightness of the clusters and also compare the computed results against manually clustered data. Note that manual clustering is possible because of the relatively small set of samples available in our study. Our computational technique, on the other hand, can be applied to potentially very large sets of samples.

2.1. Data Description

In our study, a *sample* is the flow cytometry data collected for a patient at a certain time point. Each sample corresponds to a FCS file that measures the cell intensity for hundreds of thousands of cells on a set of channels. Each raw FCS data file contains a cell-channel intensity matrix where each row corresponds to a cell, each column corresponds to a channel, and each entry is the intensity value of a certain cell at a certain channel. In our study, there are 10 channels. The number of cells are typically in the order of 10^5. A snippet of the sample file containing the intensity values is shown in **Table 1**. Note that the cells are ordered arbitrarily in a FCS file and, as a result, the intensity values are not directly comparable across samples.

2.2. Feature Modeling

As the cells are not uniquely identified and can be ordered randomly in a raw FCS file, the intensity values are not directly comparable across samples and can not be used directly as features for clustering. To address this, we first transform the absolute intensity values contained in the raw FCS data file into intensity distribution by counting the number of cells for each intensity value. This way, they can be compared across samples and used as features for clustering. For example, if sample 1 and sample 2 have the same or similar number of cells for each intensity value at each channel, they can be considered similar to each other. The transformed feature data now contains a intensity-channel distribution matrix where each row corresponds to a particular intensity value, each column corresponds to a channel, and each entry is the number of cells with a certain intensity value at a certain channel. The intensity values typically have a range of $[0, 5000]$. A snippet of the cell intensity distribution data for the sample FCS file (**Table 1**) is illustrated in **Table 2**. For example, there is 1 cell that has intensity value of 150 at FSC-A channel.

2.3. Feature Reduction

After we convert the intensity values into intensity distribution values, each sample now contains thousands of data points (features) for each channel, determined by the range of intensity values. This large number of dimensions poses a challenge for effective and efficient clustering.

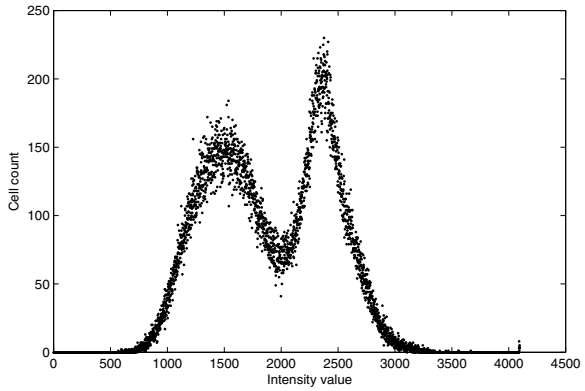

Figure 1. Scatter Plot of the Sample Feature Data at FSC-A Channel

We first generated a scatter plot of the intensity values and their corresponding cell counts for each channel in order to obtain a visual understanding of the patterns. The plot revealed that the data points follow a rough curve with peaks and valleys. A scatter plot of the sample intensity distribution file (**Table 2**) at FSC-A channel is presented in **Figure 1**. This motivates us to apply regression analysis techniques, in particular, polynomial fitting, to reduce the data. By storing the parameters of the polynomial that represent the original data and discarding the original data points, we can reduce the number of features significantly.

At the first glance, polynomial fitting may be ill-posed in our context due to the large degree of freedom for the parameter settings (the large number of feature points), i.e. small disturbances can lead to drastic changes in coefficients whereas more or less same shape can be described by very different polynomials. For exactly this reason, our goal is to find an approximate fit with a low-order polynomial even if a better fit is possible with a high-order polynomial. By approximately following the peaks and valleys of the data, the polynomials would still be useful for clustering even if they do not fit the data precisely. Moreover, this is how humans compare and cluster curves intuitively.

The remaining question is how to determine the optimal order for the polynomial fitting for our purposes. We adopted the least square method and tested polynomials with different orders to uncover the optimal order based on the tradeoff of fitting error and computation efficiency.

Table 3. Polynomial Fitting for Feature Data

channel	order	relative error	CPU time (sec)
FSC-A	8	25%	4
	9	14%	5
	10	12%	10
FSC-H	8	30%	4
	9	15%	5
	10	13%	10
SSC-A	8	28%	4
	9	17%	5
	10	12%	10
SSC-H	7	28%	4
	8	14%	4
	9	12%	8
Comp-FITC-A	7	33%	4
	8	20%	4
	9	17%	8
Comp-PE-A	7	25%	4
	8	17%	4
	9	15%	8
Comp-PerCP-A	7	29%	4
	8	17%	4
	9	15%	8
Comp-PE CY-7-A	6	29%	3
	7	17%	3
	8	16%	5
Comp-Pacific Blue-A	6	31%	3
	7	16%	3
	8	15%	5
Comp-APC-A	6	25%	3
	7	19%	3
	8	17%	5

For example, for channel FSC-A, the relative error (absolute error/average intensity distribution value) improved significantly when the order of the polynomial increases from 8 to 9 but only marginally from 9 to 10. On the other hand, the CPU time for the polynomial fitting increased marginally when the order increases from 8 to 9 but significantly (doubled) from 9 to 10. Therefore, order 9 is selected for channel FSC-A. **Table 3** shows the selected order (highlighted) for each channel with its neighboring orders with the relative error and computation time.

2.4. Clustering

Once the features are generated and reduced for each sample, we clustered the samples based on the reduced features as well as the original features to evaluate the effect of the data reduction. We first clustered the data using features from individual channels, and then clustered the data using features concatenated from all channels. We used a set of clustering algorithms including Cobweb, EM, FarthestFirst, and k-Means, implemented in the open source data mining toolkit Weka [10].

3. Experimental Results

This section presents a set of experiments evaluating the feasibility, effectiveness, and cost of our proposed approach. Our dataset contains 18 samples collected from 3 patients at different time points (6 samples per patient) from the Protective Immunity Project at Emory University. Our main goal is to answer the question: does the regression analysis based feature reduction help with the clustering analysis with respect to quality and efficiency?

3.1. Evaluation Metrics

To evaluate the quality of the clustering result, we used a supervised measure, *Jaccard score*, and an unsupervised measure, *summed square error* [9]. We manually clustered the samples based on their clinical classification (immunization response) when available as well as the visual similarity of their cell intensity distribution curves. We then used this manual clustering result as a reference against which our computed clusters are compared. The Jaccard score is defined as,

$$J(T,S) = \frac{n_{11}}{n_{11} + n_{10} + n_{01}} \quad (1)$$

where $J(T,S)$ is the Jaccard score of a solution S against the *true* solution T, n_{11} is the number of pairs of data objects that are in the same cluster in both S and T, n_{01} the number of pairs that are in the same cluster only in S, and n_{10} the number of pairs that are in the same cluster only in T. The resulting score is in the range of [0,1] with higher score indicating a better clustering quality.

As the manual clustering result could be potentially biased towards the curve fitting approach, we also use an unsupervised measure, the summed square error defined as,

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \quad (2)$$

where E is the sum of square error for all objects in the dataset, p is a given object belonging to cluster C_i, and m_i is the mean of cluster C_i. The absolute error itself may not be meaningful but when comparing different clustering results, a lower error indicates a better clustering quality.

To evaluate the efficiency of the clustering algorithm, we measured the CPU time for the clustering process as well as the time for data preprocessing for the feature reduction.

3.2 Quality of Clustering

We first verify our hypothesis that the feature reduction will improve the clustering quality by extracting the essential features of the data. We only report the result of FarthestFirst clustering algorithm due to space restrictions and refer readers to [5] for the results of different clustering algorithms. **Figure 2** compares the Jaccard Score and summed square error of clustering based on extracted features and original features using each single channel and all channels, respectively. It can be observed that the clustering based on extracted features achieves a better Jaccard Score and a lower summed square error than the original features for most of the channels and all channels. This verified our hypothesis that polynomial fitting based feature reduction improves the quality of clustering significantly.

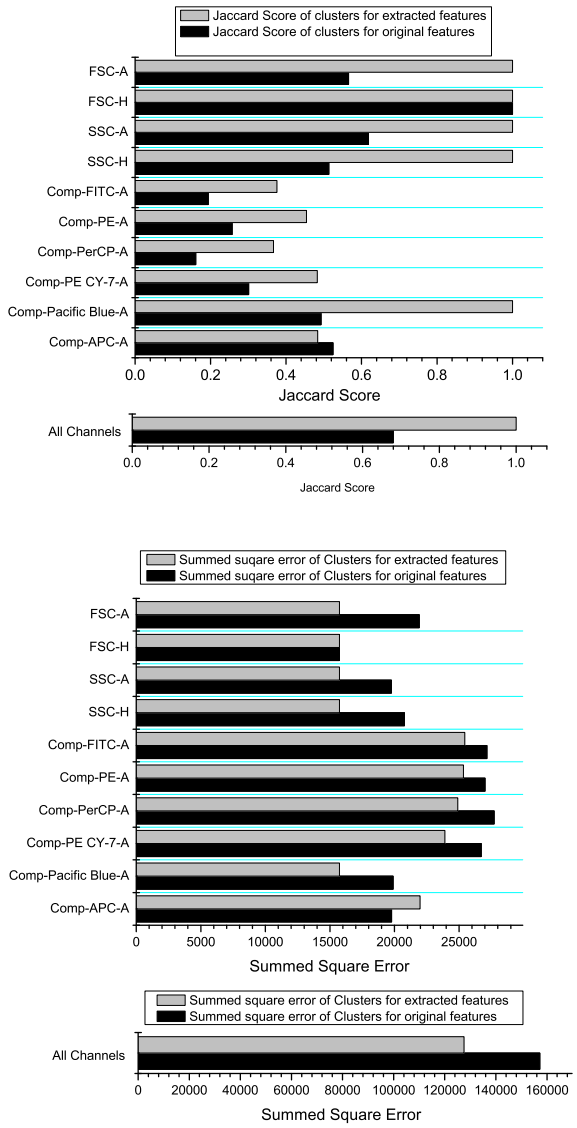

Figure 2. Clustering Quality for Single Channels and All Channels (Jaccard Score and Summed Square Error)

3.3 Efficiency of Clustering

We also evaluated the impact of feature reduction on the efficiency of clustering by measuring the CPU time. **Figure 3** presents the average CPU time for feature extraction and for clustering based on extracted features and original features using single channels and all channels. We observe that clustering based on extracted features significantly shortens the time for clustering (around 6.5 times). In addition, if we consider the overall time for the approach by summing the clustering and feature extraction time, it still represents slight improvement over the original feature based clustering.

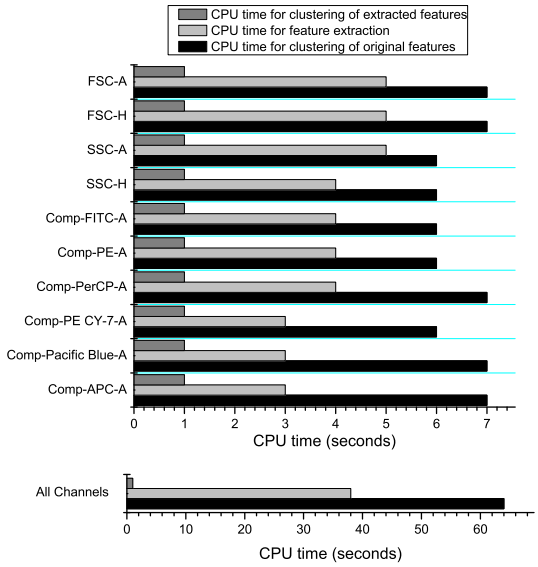

Figure 3. Clustering Efficiency for Single Channels and All Channels (CPU Time)

4. Related Work

Some clustering algorithms have been recently applied to multi-dimensional flow cytometry data for clustering cells into cell groups of a single FCS file [11, 8, 3]. There are also work focusing on visualizing flow cytometry data [6]. Our work complements these research and focuses on comparing and clustering multiple FCS files. It offers a set of features that can be potentially used in combination with the cell group based features studied in the traditional gating methods and cell clustering methods for both supervised and unsupervised learning.

Data dimension reduction has been applied in a variety of data analysis problems [2]. It is on our future research agenda to study different dimension reduction techniques and their implications on flow cytometry data clustering.

5. Conclusion

We developed and presented a framework for comparing and clustering flow cytometry data files that contain cell intensity values for different channels. We experimentally show that our system produces meaningful results with good efficiency. While our work is a convincing proof-of-concept, there are several aspects of our system that will be further explored. First, we would like to investigate other parametric models such as wavelet or Fourier transforms as well as other dimension reduction techniques. Second, we are interested in studying the correlations among the channels as well as incorporating the cell-based clustering into the sample clustering process. In addition, we are planning to explore temporal data analysis techniques to learn the variances and evolving trend of samples along different time points. Finally, we are integrating the FCM data with clinical datasets and possibly gene expression datasets to perform supervised learning in order to predict patients' immune status.

References

[1] T. Donker. FloCK: Flow cytometry clustering by k-means, 2007. http://theory.bio.uu.nl/tjibbe/flock/.

[2] J. Han and M. Kamber. *Data Mining: Concepts and Techniques, 2nd ed.* Morgan Kaufmann Publishers, 2006.

[3] J. Lakoumentas, J. Drakos, M. Karakantza, N. Zoumbos, G. Nikiforidis, and G. Sakellaropoulos. The probabilities mixture model for clustering flow-cytometric data: An application to gating lymphocytes in peripheral blood. *Biological and Medical Data Analysis*, 4345/2006, 2006. Springer.

[4] J. F. Leary, J. Smith, P. Szaniszlo, and L. M. Reece. Comparison of multidimensional flow cytometric data by a novel data mining technique. In *Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues V., Proceedings of the SPIE*, 2007.

[5] L. Liu, L. Xiong, J. J. Lu, K. M. Gernert, and V. Hertzberg. Sample clustering of flow cytometry data. Technical Report TR-2008-003, Emory University, 2008.

[6] D. Sarkar, N. L. Meur, and R. Gentleman. Using flowviz to visualize flow cytometry data. *Bioinformatics*, 24, 2008.

[7] H. Shapiro. *Practical Flow Cytometry, 4th ed.* John Wiley & Sons, Inc., 2003.

[8] U. Simon, H.-J. Mucha, and R. Bruggemann. Model-based cluster analysis applied to flow cytometry data. *Innovations in Classification, Data Science, and Information Systems*, 2006. Springer.

[9] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[10] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann, 2005.

[11] 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. *J. of Biomedical Informatics*, 40(3), 2007.