**Artificial Intelligence-Driven Automation of Flow Cytometry Gating**

Gabriella Rivera

Applied Data Science Master’s Program

Shiley Marcos School of Engineering / University of San Diego [gabriellarivera@sandiego.edu](mailto:gabriellarivera@sandiego.edu)

John Vincent Deniega

Applied Data Science Master’s Program  
Shiley Marcos School of Engineering / University of San Diego   
 [jdeniega@sandiego.edu](mailto:jdeniega@sandiego.edu)

**ABSTRACT**

Flow cytometry is a biochemical process that measures the physical and chemical characteristic of cells in a liquid suspension. This method enables the identification and classification of various cellular populations, such as lymphocytes, monocytes, and granulocytes – from Peripheral Blood Mononuclear Cell (PBMC) samples. Clustering algorithms aim to objectively simplify the classification accuracy of a given population's cellular type to further immunological and clinical diagnostic purposes. By exploring algorithms like K-means, Agglomerative, and Gaussian Mixture Modeling, the dataset containing metrics on front and side scatter areas of cell scans, fluorescence on markers, and other dimensions can potentially lead to automatic gating and bring forth insights on cell population characteristics, resulting in increased analytical throughput and increased rate of medicinal and pharmacological breakthroughs.

1 Introduction

According to Brestoff and Frater (2022), flow cytometry is a cost-prohibitive cellular identification and pharmacological discovery process that both requires exceptional capital investment in infrastructure and is inherently difficult in adopting and implementing the latest techniques and technologies in the field. Open-source and low-cost options may serve as a viable bridge technology between capital-intensive investment cycles for laboratories to be able to continue evolving and increasing their analytical throughput without relying on the next hardware upgrade. By using existing mathematically-driven principles inherent to artificial intelligence, biochemists may be able to reduce the analytical inputs required to perform routine cellular classification and clustering of PBMCs. This labor-intensive process ultimately serves to evaluate the efficacy of experimental groups related to clinical trials that are required in the development of new life-saving medicines.

2 Background

Flow cytometry can be a capital-intensive process that requires significant investments in laboratory-grade biomedical equipment, dedicated graphics processing and tensor processing units, expansive random-access memory, and proprietary analytical software licenses. With Flow Cytometry Standard (FCS) files readily available on public repositories and by leveraging open-source and permissive license packages such as Scikit-Learn and Matplotlib to perform computational transformations to FCS data, we aim to discover cost-effective alternatives to expensive enterprise software licenses that perform flow cytometry analysis, which may result in significant reduction in the barriers to entry in biochemical flow cytometry.

Because cellular populations related to these FCS files number in the millions of records across multiple laboratory readings, this project will place heavy emphasis on dimensionality reduction in order to meet the constraints of being both cost-effective and hardware resource-efficient. Accomplishing such a feat would result in independent biochemical scientists to perform analyses without relying on exceptionally powerful computing hardware resources or costly proprietary enterprise-level software licenses.

2.1 Problem Identification and Motivation

As of this publication, flow cytometry gating is a manual process that involves a highly-trained biochemist to process and analyze the results of optical scans of cellular assays that may be further augmented by fluorescent substrates. Because of the complex and highly-dimensional nature of the data, these scientists rely on a best-practices approach based on their own respective processes and frameworks. Because of the potential variability of these processes and frameworks, the resulting findings from interpreting scan results is dependent on both the breadth and depth of methods of a given supervising scientist, thus resulting in both an increase in cost of analysis due to human error and omission as well as a reduction in consistency of results.

2.2 Definition of Objectives

The research team aims to utilize open-source and publicly-available resources from well-known algorithms known in data science to include principal component analysis, t-distributed stochastic neighbor embedding, and unsupervised clustering machine learning methods as well as FCS data hosted by FlowRepository (2020). Once data is cleaned for noise from scan data, the team aims to train models or machine-learning applications that have potential for value-added analysis relative to that of a typical human biochemist. Upon evaluation, success is generally defined when automated analysis reaches parity with a human analyst of at least 90% classification accuracy of PBMCs toward their respective dendritic cellular type on an unseen test set containing FCS scan data. In the event that this evaluation criterion is not met, further justification would have to be provided whether the measured degree of accuracy is acceptable relative to the speed of analyses.

3 Literature Review (related works)

Since 2016, a number of academic threads have been studied involving the advancement in flow cytometry, the iteration of methodologies when incorporating machine learning applications on FCS data, as well as different strategies in how to potentially automate the classification of cellular groups. By 2024, Ng et al. (2024) demonstrate maturity over an eight-year period that transitions the focus of academic research from the “what” normally seen in earlier works into the “how” with respect to interdisciplinary guidelines as well as quality control and assurance of future deployment of artificial intelligence in flow cytometry (p. 228).

3.1  FlowAI: Automatic and interactive anomaly discerning tools for flow cytometry data

FlowAI is a software package for the statistical computing language R, which Monaco et al. (2016) developed as a means to both clean FCS files from anomalies and to assess the resulting quality of the cleaned data normalized by the flow rate of a given reading. When flow rate abruptly changes during a scan, the readings may exhibit data inconsistencies. These data inconsistencies are considered anomalous and are discarded from the dataset. Using time-series analysis, the resulting dataset is broken into trend and cyclical components before being normalized by penalization function measuring absolute deviation of a data point from the median. Monaco et al. (2016) place an emphasis on data quality and anomaly handling, which are crucial considerations to flow cytometry, however they do not address the next step in automatic gating of cellular types, which is the focus of our research.

3.2  An open-source solution for advanced imaging flow cytometry data analysis using machine learning

Hennig et al. (2017) identify the challenges associated with the manual and subjective nature of flow cytometry, resulting in inconsistent in analysis. The given solution is to utilize open-source software, CellProfiler, to use raw image files to identify cell types from a flow cytometer image. Our research shares the open-source idea of being able to leverage existing machine learning algorithms to automatically classify these cell types. Contrasting the team of Hennig et al. (2017) on classification differs greatly in their use of visual image data as the basis for classification rather the numerical scan data from fluorescent biological marker excitation that is central to our approach (p. 202).

3.3  Comprehensive phenotyping of human dendritic cells and monocytes

Mair and Liechti (2020) identify the potential benefits in using biological markers to identify the phenotypes specific to dendritic cells and monocytes for cellular classification. This particular research focuses on a potentially more significant subset of biological markers and lineages that aim to more precisely identify different cellular categories as a result of their fluorescence excitation scan data. This work serves as the source data of our project which uses Python-based machine learning packages for automatic gating. A similar methodology was employed by Hennig et al. (2017) who instead synthesized with visual imagery data with the open-source software, CellProfiler.

3.4  Application of machine learning for cytometry data

Hu et al. (2022) acknowledge the complex challenge of highly-dimensional flow cytometry data and the potential for existing machine learning software packages to perform analysis on this type of data. This particular team first focuses on dimensionality reduction by means including Principal Component Analysis and stochastic methods, unsupervised and supervised machine learning methods to predict resulting clinical outcomes such as healthy populations versus diseased populations (p. 2). Our project aims to build on this research with greater training and tuning toward existing biological knowledge cross-validated across different FCS file scan results.

3.5  Recommendations for using artificial intelligence in clinical flow cytometry

Most recently, Ng et al. (2024) focuses on a more interdisciplinary approach to using artificial intelligence in flow cytometry with unique considerations for clinical risk management, quality control and assurance, and computational efficiency. This requires extensive consideration as to the narrative annotations required for clinical implementation. Though the article is comprehensive across multiple sectors related to flow cytometry and the technical and regulatory nuances required when applying artificial intelligence, it only provides general recommendations and guidance for future scientist who wish to leverage this new technology. Relative to our existing work, our research team aims to apply these general recommendations and implement them in an open-source and demonstrable product for flow cytometry automatic gating.

4 Methodology

Our platform approach is organized into several key subtopics, starting with data extraction. The FCS files were read using FlowCal, which were then transformed into NumPy and Pandas objects to facilitate compatibility with Interactive Development Environments (IDEs), specifically with Jupyter Notebook and Google Colab. Exploratory Data Analysis (EDA) was then employed to generate data visualizations and detect outliers, facilitating the data cleaning process. Following this, the data was split into training, validation, and testing sets. Then dimensionality reduction was applied to simplify the complexity of the data before feeding the sets into the classification models. The following GitHub link contains the code written from the beginning to the final launch of the completed flow analysis product: <https://github.com/vanguardfox/ADS599>

4.1 Data Extraction and Format Conversion

The flow cytometry dataset was acquired from FlowRepository, a public database for flow cytometry peer-reviewed experiments. It contains a staining panel from Mair and Leichti’s (2020) article that aims to refine traditional and recently described markers for phenotyping dendritic cells and monocytes. The panel is composed of 23 fluorochrome markers including the time of collection as well as the forward scatter and side scatter measurements. There were 28 fluorescence channels in total, five of the wavelengths are unlabeled as a consequence of continuous data acquisition. About 2 million cells were collected per sample, which is reflected in the file sizes between 267 to 405 megabytes for one PBMC FCS file. Compensation FCS files are also included in the dataset package.

The data extraction process starts by defining the file paths to the FCS data. Then employing the FlowCal.io.FCSData function to read and load the data into a structured NumPy array. The available attributes from the FCS meta data are inspected to retrieve the channel marker labels using the channel\_labels() method. The first three features for forward scatter area (FSC-A) and side scatter area measurements (SSC-A) are updated to "FSC-A", "FSC-H", and "SSC-A". The "Time" label is adjusted as well. The FCS data is then converted into a Pandas DataFrame and the column marker labels retrieved are applied. The "Time" column is scaled while the rest of the numeric columns are converted to little-endian float32 from object datatype. Finally, the formatted DataFrame is saved as a CSV file for further analysis.

4.2 Data Feature Selection

For feature selection, columns with missing marker labels were removed from the dataset. Although 23 fluorescence markers were used to identify specific cell surface proteins, a total of 28 channels were acquired. The five unused channels with missing marker labels were subsequently discarded. To focus on the cellular pathways relevant for dendritic cell phenotyping (Figure 1), several monocyte markers were also excluded. Specifically, markers such as CD45RA, CD3, CD19, CD14, CD20, HLA-DR, CD123, CD11c, and Live Dead UV Blue were retained, along with Time and scattering measurements. This reduced the feature set to 13.

**Figure 1**

*Maecker et al. (2012) Dendritic Cell Lineage*

A diagram of cells and cells

Description automatically generated

4.3 Exploratory Data Analysis

Two-dimensional visualizations were plotted to help identify general areas and priorities for cleaning the dataset. A SSC vs. Time scatter plot (see Figure 2) was created to identify inconsistencies during data acquisition as cells pass through the inflection point against the detection probe. This plot ensures that only cells collected during the stable portion of the sample run are included in the analysis. Figure 1 further illustrates the gating boundaries, which capture consistent readings across time and help exclude artifacts or outliers caused by fluctuations in the data acquisition process.

**Figure 2**

*Acquisition Plot*

A diagram of a graph

Description automatically generated with medium confidence

A frequency plot of the forward scatter area measurement (FSC-A) is shown in Figure 3. FSC-A is used to estimate cell size in a given sample mixture. In this case, the plot reveals three distinct peak populations, corresponding to the expected cell types in PBMC as lymphocytes, monocytes, and granulocytes (left to right). The red gate is applied to exclude cellular debris, which typically appears as smaller events at the lower end of the FSC-A distribution.

**Figure 3**

*Cellular Debris Plot*

A blue graph with numbers and a red line

Description automatically generated

Figure 4 illustrates the removal of doublets or cell aggregates from the dataset, which are typically identified by an inconsistent ratio between FSC-A and FSC-H. Doublets tend to exhibit a higher FSC-H relative to FSC-A as they are larger due to the presence of two cells but still emit a "tall" scatter signal. In this case, the plot reveals a relatively small population of cell aggregates, identified by the gate on the y-axis. The x-axis limit is set further out to avoid truncating the monocyte population in the SSC-A vs. FSC-A plot, ensuring that monocytes are not excluded while still removing the majority of doublets.

**Figure 4**

*Single Cell Plot*

A diagram of a single cell

Description automatically generated

Finally, the Live/Dead™ UV Blue marker was used to exclude dead cells from the dataset. The dye binds to free amines present on the surface and interior of dead cells, producing an intense fluorescent signal. In contrast, live cells emit a much weaker signal. As shown in Figure 5, live cells are gated to the left of the Live/Dead marker at 700 in both the original and log-transformed Live/Dead channels. This gating strategy reduced the dataset to 1.3 million viable cells.

**Figure 5**

*Live Dead Cells: Original vs Log(Live Dead)*

A graph of a number of people

Description automatically generatedA blue graph with red line

Description automatically generated

4.4 Data Cleaning

As alluded to during exploratory data analysis (EDA), data cleaning involved several steps to ensure the quality of the dataset before passing it through the model classification development. These steps included checking for missing values using the Klib package – No missing values were detected – followed by removing irrelevant cell populations, identifying cellular debris, excluding doublets, and filtering out dead cells. Instances where the "Time" variable was outside the desired range were excluded by gating on values greater than 3 and less than 215. Additionally, cell populations with abnormal FSC and SSC characteristics were removed to eliminate debris and doublets. For FSC, cells were selected by gating for values greater than 20,000, less than 550,000 for FSC-A and less than 200,000 for FSC-H. For SSC, a similar gating strategy was applied, where cells were retained only if SSC-A values were between 110 and 20,000. Finally, dead cells were excluded by applying a threshold on the Live/Dead UV Blue marker which keeps only those with values below 700. After these steps, the dataset was refined to include quality viable cells suitable for data splitting and model development. The cleaned dataset profile is shown in Figure 6.

**Figure 6**

*Clean Dataset Light Scatter Plot*

A diagram of a diagram

Description automatically generated with medium confidence

To prepare the data for model training and evaluation, the dataset was split into separate subsets: training, validation, and test sets. Initially, 20% of the data was reserved for testing using the train\_test\_split function, resulting in the train\_val set containing 80% of the original data. This train\_val set was then further split into training and validation sets, with 25% of the train\_val data used for validation, which is equivalent to 20% of the original dataset. This process resulted in 60% training, 20% validation, and 20% testing distribution sets. These splits ensure that the model can be trained, tuned, and evaluated on distinct data subsets, minimizing overfitting and ensuring robust performance assessments.

4.5 Dimension Reduction

We applied dimensionality reduction techniques to simplify the complexity of the datasets before inputting them into the classification models. First, we use PCA to reduce the dimensionality and capture the most significant variance in the data. Then, we further refine the dataset by applying t-SNE or UMAP. t-SNE is used to preserve local structures and reveal clusters in the data, while UMAP provides a more scalable approach that retains both local and global patterns. This multi-step process helps to reduce noise, improve model efficiency, and enhance interpretability by focusing on the most relevant features in the data.

4.5.1 Principal Component Analysis (PCA)

After selecting features specific to dendritic cell markers and reducing the dataset to 12 columns, computationally expensive pairwise comparisons pose a challenge if the data is used directly. To address this hardware limitation, PCA is applied to reduce the data’s dimensionality. Using the elbow method to determine the optimal number of components, the cumulative explained variance plot (Figure 7) shows that PCA1 captures less than 95% of the variance, while PCA2 accounts for 97%. This indicates that the first two components explain the majority of the variance. PCA3 is also included to enable 3D visualization (Figure 8), which provides an additional perspective on the data's structure and relationships, helping to better distinguish patterns that may not be as apparent in lower-dimensional representations. By transforming the 12 marker columns into three principal components the variance is maximized while simplifying the data which then improves the efficiency of downstream clustering and modeling steps.

**Figure 7**

*PCA Cumulative Explained Variance Plot*

A graph of a number of components

Description automatically generated

Figure 8

*3D PCA Plot of Training Dataset*

A graph of different colors

Description automatically generated with medium confidence

4.5.2 T-SNE

Using t-SNE helps analysts capture local structures in high-dimensional data. Using three resulting components from PCA, t-SNE aids in being able to visualize different clusters while still being memory-efficient, which is a key consideration for our cost-effective solution. Because about .99 of the variance is still captured with only three components, the loss of one-percent of the data for significant memory efficiency directly addresses expected hardware limitations with a computationally intensive algorithm such as t-SNE. Further, t-SNE directly addresses the subjectivity issue that lends to analysts potentially being inconsistent across multiple scatter plots. As such, this method provides clearer and more objective population boundaries for the purposes of gating where different clusters may be isolated for further downstream analysis as shown in Figure 8.

**Figure 8**

*Working t-SNE Gating of CD19 vs. CD3 Markers*

A diagram of a cluster of dots

Description automatically generated with medium confidence

4.5.3 UMAP

UMAP preserves both local and global data structure, which is important for downstream lineage-based gating procedures, as clustering and categorization relies not only on the point analysis of a dot plot, but the preceding markers that led to the reading. In other words, UMAP enables analysts to perform cellular population identification using lineage data that would be preserved with this method.

Further, along with the benefits of dimensionality reduction and preservation of global data between clusters, UMAP utilizes greater memory efficiency than t-SNE since the former is deterministic and does not require a probabilistic distribution of outcomes with respect to which neighbors belong to which cluster. This, however, requires us to perform sufficient cross-validation of hyperparameters in order to find the optimal number of neighbors and minimum distance required for UMAP when transforming the original data.

**References**

Beckman Coulter. (2022). *Automatic gating - Beckman Coulter*. Flow Cytometry. <https://www.beckman.com/flow-cytometry/software/cytobank-premium/learning-center/automatic-gating>

Brestoff, J. R., & Frater, J. L. (2022). Contemporary challenges in clinical flow cytometry: Small samples, big data, little time. *Journal of Applied Laboratory Medicine*, *7*(22), 931-944. <https://doi.org/10.1093/jalm/jfab176>

Cossarizza, A., Chang, H. D., Radbruch, A., Acs, A., Adam, D., Adam-Klages, S., Agace, W. W., Aghaeepour, N., Akdis, M., Allez, M., Almeida, L. N., Alvisi, G., Anderson, G., Andrä, I., Annunziato, F., Anselmo, A., Bacher, P., Baldari, C. T., Bari, S., Barnaba, V., … Zychlinsky, A. (2019). Guidelines for the use of flow cytometry and cell sorting in immunological studies (second edition). *European journal of immunology*, *49*(10), 1457–1973. <https://doi.org/10.1002/eji.201970107>

FlowRepository. (2020). FlowRepository ID FR-FCM-Z32U. Retrieved from <http://flowrepository.org/experiments/3166/download_ziped_files>

Hennig, H., Rees, P., Blasi, T., Kamentsky, L., Hung, J., Dao, D., Carpenter, A. E., & Filby, A. (2016). An open-source solution for advanced imaging flow cytometry data analysis using machine learning. *Methods*, 112(2017), 201–210. <http://dx.doi.org/10.1016/j.ymeth.2016.08.018>

Hu, Z., Bhattacharya, S., & Butte, A. J. (2022). Application of machine learning for cytometry data. *Frontiers in immunology,* 12, 787574. <https://doi.org/10.3389/fimmu.2021.787574>

Lee, J. A., Spidlen, J., Boyce, K., Cai, J., Crosbie, N., Dalphin, M., Furlong, J., Gasparetto, M., Goldberg, M., Goralczyk, E. M., Hyun, B., Jansen, K., Kollmann, T., Kong, M., Leif, R., McWeeney, S., Moloshok, T. D., Moore, W., Nolan, G., Nolan, J., … Brinkman, R. R. (2008). MIFlowCyt: the minimum information about a Flow Cytometry Experiment. *Cytometry. Part A: the journal of the International Society for Analytical Cytology*, 73(10), 926–930. <https://doi.org/10.1002/cyto.a.20623>

Maecker, H. T., McCoy, J. P., & Nussenblatt, R. (2012). Standardizing immunophenotyping for the Human Immunology Project. *Nature Reviews Immunology,* 12(3), 191–200. <https://doi.org/10.1038/nri3158>

Mair, F., & Leichti, T. (2020). Comprehensive Phenotyping of Human Dendritic Cells and Monocytes. *Journal of Quantitative Cell Science*, 99(3), 231–242. <https://doi.org/10.1002/cyto.a.24269>

Monaco, G., Chen, H., Poidinger, M., Chen, J., Magalhaes, J., & Larbi, A. (2016). FlowAI: Automatic and interactive anomaly discerning tools for flow cytometry data. *Bioinformatics*, 32(16), 2473–2480. <https://doi.org/10.1093/bioinformatics/btw191>

Ng, D. P., Simonson, P. D., Tarnok, A., Lucas, F., Kern, W., Rolf, N., Bogdanoski, G., Green, C., Brinkman, R. R., & Czechowska, K. (2024). Recommendations for using artificial intelligence in clinical flow cytometry. *Cytometry part b: Clinical cytometry*, 106(4), 228–238. <https://doi.org/10.1002/cyto.b.22166>

Spidlen J, Breuer K, Rosenberg C, Kotecha N and Brinkman RR. (2012). FlowRepository - A Resource of Annotated Flow Cytometry Datasets Associated with Peer-reviewed Publications. *Cytometry A*. 81(9), 727-31. <https://doi.org/10.1002/cyto.a.22106>