SWIFT: SCALABLE WEIGHTED ITERATIVE SAMPLING FOR FLOW CYTOMETRY CLUSTERING

Iftekhar Naim[1], *Suprakash Datta*[4], *Gaurav Sharma*[1,2], *James S. Cavenaugh*[3], *and Tim R. Mosmann*[3]

[1]Dept. of Electrical and Computer Engineering, [2]Dept. of Biostatistics and Computational Biology, [3]Center for Vaccine Biology and Immunology,
University of Rochester, Rochester, NY
[4]Dept. of Computer Science and Engineering, York University, Toronto

ABSTRACT

Flow cytometry (FC) is a powerful technology for rapid multivariate analysis and functional discrimination of cells. Current FC platforms generate large, high-dimensional datasets which pose a significant challenge for traditional manual bivariate analysis. Automated multivariate clustering, though highly desirable, is also stymied by the critical requirement of identifying rare populations that form rather small clusters, in addition to the computational challenges posed by the large size and dimensionality of the datasets. In this paper, we address these twin challenges by developing a two-stage scalable multivariate parametric clustering algorithm. In the first stage, we model the data as a mixture of Gaussians and use an iterative weighted sampling technique to estimate the mixture components successively in order of decreasing size. In the second stage, we apply a graph-based hierarchical merging technique to combine Gaussian components with significant overlaps into the final number of desired clusters. The resulting algorithm offers a reduction in complexity over conventional mixture modeling while simultaneously allowing for better detection of small populations. We demonstrate the effectiveness of our method both on simulated data and actual flow cytometry datasets.

Index Terms— Flow cytometry, clustering, Gaussian mixture model, sampling, expectation-maximization

1. INTRODUCTION

Flow cytometry (FC) has recently emerged as a high throughput technology with a wide variety of biological applications. We focus here on immunophenotyping where the presence of antigens in blood cells is detected by fluorescently labeled antigen-specific antibodies. The fluorophores bound to each cell are recorded using laser excited fluorescence with detectors matched to the wavelengths emitted by the individual fluorophores. Thus the technology enables measurement of the relative amount of each antigen within a cell. FC has proven useful for the diagnosis and monitoring of different types of acute leukemias, chronic lymphoproliferative disease, HIV infections, and malignant lymphomas [1].

The goal of the FC data analysis is to identify populations that express similar behaviors in the measured variables. Traditional analysis is done by manually drawing gates or regions of interest on bivariate plots and sequentially filtering the data until homogeneous cell populations are identified. This approach, known as bivariate gating, is subjective, labor intensive, and scales poorly with increasing number of dimensions. Moreover, many high dimensional features may not be perceptible in lower dimensional (bivariate) projections. So it is highly desirable to use automated multivariate clustering instead of bivariate manual gating. However automated multivariate clustering comes with many challenges. The FC datasets can be very large ($\sim$ 1 million cells) with high dimensionality ($\sim$ 20 dimensions). For many clustering algorithms, this size and dimensionality are prohibitively expensive both in terms of computation and memory. Also there exist overlapping, non-spherical clusters that are heterogeneous in size, shape and orientation. Moreover, FC datasets tend to have few sparse and rare populations of interest (few hundred cells or less). Finding these rare populations out of millions of cells is extremely difficult for the existing clustering methods. Recently several methods were suggested for clustering FC data [2–4] and all of them are based on mixture model [5] clustering and the Expectation Maximization (EM) algorithm [6]. Thus far, however, these methods have only been applied to small datasets and not scalable to large datasets and tend to miss small sparse populations in the presence of other large clusters.

In our work, we address two main issues with the existing methods, the issue of scalability and the identification of rare populations. We propose an iterative sampling framework that is based on mixture model fitting to random samples drawn from the dataset and probabilistically "damp" the well explained larger clusters iteratively. This method can yield higher performance and scalability and at the same time increase the probability of finding the smaller clusters which may not be correctly estimated in the presence of the larger clusters. Our experiments show that, the proposed method can find some very small populations that the standard EM algorithm (working on full datasets) often fails to find in the presence of larger partially overlapping populations. Previously sampling based methods have been proposed to scale EM algorithm to large datasets [7–9]. Our method is different in that, it is driven by the goal of finding the rare small populations which are highly significant in FC. We propose a novel posterior-probability based iterative weighted resampling technique to "damp" the already well-explained larger clusters. We focus mainly on Gaussian Mixture model based clustering because of its analytic closed form and computational efficiency. However, our framework can easily be extended to accommodate mixture of t or skewed t-distributions. Finally, we propose a hierarchical merging method to merge overlapping Gaussian components that may represent a single non-Gaussian cluster.

2. PROPOSED METHOD

We propose a two-stage method for automated multivariate clustering of FC data. The presence of overlapping non-spherical clusters, background noise and near Gaussian distribution of different clusters in FC data motivated the use of Gaussian mixture model clustering. In the first stage, we fit a k-component Gaussian mixture model to

This work was supported in part by a CEIS (NYSTAR) award, by a NSERC Discovery grant, and by a NIH Grant R24 AI054953.

the data. Many clusters in flow cytometry data are distinctly skewed and are not well-approximated by Gaussian distributions. We address this issue by representing these with two or more Gaussian components and adding a second stage for combining components that overlap to form the final set of clusters.

The EM algorithm is typically used for fitting Gaussian mixture models to data. Since the EM algorithm is inefficient on large data sets and we are particularly interested in small sparse clusters, we use a modified EM algorithm that uses weighted iterative sampling. This is described in Section 2.1. Since the input to our algorithm is the number of clusters and not the number of Gaussian components, we determine the appropriate number of Gaussian components to fit to the data using the Bayesian Information criterion (BIC) [5] which has been previously reported to provide good estimates [10].

In the second stage, we apply a graph-based hierarchical merging technique to combine overlapping Gaussian components that may represent a single non-Gaussian cluster. This is described in Section 2.2.

Let $\mathbf{X} = \{\mathbf{X}^{(i)}\}_{i=1}^{N}$ be a set of N d-dimensional vectors describing N cells in terms of the d FC measurements per cell. The probability density function of a k-component Gaussian mixture distribution can be written as:

$$p(\mathbf{X}^{(i)}|\theta) = \sum_{j=1}^{k} \pi_j \, \mathcal{N}(\mathbf{X}^{(i)}|\mu_j, \Sigma_j) \tag{1}$$

where $\mathcal{N}(.)$ denotes the normal distribution, and π_j, μ_j and Σ_j are respectively the mixing coefficient (fraction of points belonging to each cluster), mean and covariance of the j-th component. These are known as the parameters (θ) of the Gaussian Mixture model. The goal of Gaussian mixture model clustering is to estimate the parameters θ that maximize the log-likelihood of the given data $\mathbf{X}$. Once the parameters (θ) are estimated, the cluster assignment is performed using the posterior probabilities $\gamma_j^{(i)}$, the probability of the i-th datavector ($\mathbf{X}^{(i)})$) belonging to the j-th cluster where $i = 1, \ldots, N$ and $j = 1, \ldots, k$. Specifically,

$$\gamma_j^{(i)} \quad = \quad \frac{\pi_j \mathcal{N}(\mathbf{X}^{(i)}|\mu_j, \Sigma_j)}{\sum_{l=1}^{k} \pi_l \mathcal{N}(\mathbf{X}^{(i)}|\mu_l, \Sigma_l)} \tag{2}$$

and each datavector is associated with the component for which the posterior probability is the largest.

2.1. Iterative weighted sampling for complexity reduction

Our algorithm, summarized as Algorithm 1, is designed to iteratively identify large dense clusters and perform weighted resampling from the dataset that will select the remaining datapoints (not belonging to those large cluster) with higher probability. We improve both efficiency and the ability to find small, sparse clusters by working with the weighted random samples taken from the data set. The basic intuition is that a random sample represents the large, dense populations with reasonable fidelity, but may miss the sparse, small populations. Thus, the large clusters detected in a random sample are likely to be found in the original data set. In each iteration of our algorithm (steps 5-12), we fix the parameters of the p most populous clusters and perform the weighted resampling that will select the points belonging to the remaining smaller clusters with higher probablity. In the M-step (step 9) we re-estimate all parameters except those that are already fixed. We continue this process iteratively until all the cluster parameters are fixed. After explaining the larger populations and reducing their weights while resampling, the probability of discovering smaller population increases.

The weighted resampling is not straightforward in the presence of overlapping clusters. After fixing the parameters of the largest clusters in each iteration, we resample points from the dataset based on their posterior probabilities of not belonging to those fixed clusters. Let $\mathbf{F}$ be the set of Gaussian components whose parameters have already been fixed. In the next iteration, we resample according to a weighted distribution where the probability of selecting each point $\mathbf{X}^{(i)}$ is as follows:

$$p(\mathbf{X}^{(i)} \; is \; selected) = 1 - \sum_{l \in \mathbf{F}} \gamma_l^{(i)} \tag{3}$$

This resampling technique helps us to reject points belonging to the largest fixed cluster without distorting the distributions of other overlapping smaller clusters. For example, points near the centroid of the largest cluster have a posterior probability of nearly 1.0 to belong to this cluster and are therefore almost certainly rejected in our next resampling. On the other hand, points near a Gaussian tail that overlaps with another Gaussian have lower posterior probabilities and are less likely to be rejected. This posterior probability based resampling technique is a novel aspect of our algorithm. Previously a threshold based approach was proposed that excludes only the high confidence regions of the fitted Gaussian using a threshold on probability density or outcomes of a statistical test [8]. However, that approach leaves behind the tails of the excluded Gaussian and thus introduces inaccuracy in the subsequent iterations.

```
   Input: $\mathbf{X}, k, n, p$
   $\mathbf{X}$: sequence of $N$ data vectors $\{\mathbf{X}^{(i)}\}_{i=1}^{N}$
   $k$: Number of Gaussian mixture components
   $n$: Sample size
   $p$: Number of components to fix at a time
   Output: $\theta$: Parameters of Gaussian mixture model
1  Obtain set $\mathbf{S}$ of $n$ random samples drawn from $\mathbf{X}$.
2  Estimate parameters $\theta_S$ using EM on $S$
3  Estimate posterior probabilities $\gamma_j^{(i)}$ via an E-step on $\mathbf{X}$
   using parameters $\theta_S$
4  Let $\mathbf{F}$ be the set of Gaussian components whose
   parameters have been fixed. Initialize $\mathbf{F} \leftarrow \emptyset$
5  repeat
6      Determine $\mathbf{F}_1 = \{$ The $p$ most populous Gaussian
       components $\notin \mathbf{F}\}$ for the current model $\theta_S$
7      Fix the parameters of components $\in \mathbf{F}_1$. Set
       $\mathbf{F} \leftarrow \mathbf{F} \cup \mathbf{F}_1$
8      Resample a set of $n$ points $\mathbf{S}$ from $\mathbf{X}$ with a weighted
       distribution where each point is selected with
       probability $(1 - \sum_{l \in \mathbf{F}} \gamma_l^{(i)})$
9      Apply modified EM algorithm on $\mathbf{S}$ that does not
       update the parameters of already fixed components.
       In the M step, update only components $\notin \mathbf{F}$
10     Normalize the mixing probabilities $\pi_j, j \notin \mathbf{F}$,
       computed in the M step to $(1 - \sum_{l \notin \mathbf{F}} \pi_l)$
11     Perform a single E-step on $\mathbf{X}$ to recalculate the
       posteriors $\gamma_j^{(i)}$
12 until all the components are fixed
13 $\theta \leftarrow$ parameters of all the components $\in \mathbf{F}$
```

Algorithm 1: Iterative sampling based EM

As expected, the computational complexity is significantly reduced by the iterative sampling approach. For N data points with dimensionality d, the computational cost of finding k clusters using the traditional EM algorithm is $\mathcal{O}(Nkd^2)$ per iteration. On the other hand, for sample size n, the cost of each EM iteration of the proposed method is $\mathcal{O}(nkd^2)$.

2.2. Hierarchical merging of clusters

As mentioned, the proposed method merges highly overlapping Gaussian components to get clusters. For deciding which components to merge, we use the symmetric KL divergence as the distance measure between two components [11].

$$\mathcal{D}_s\left[p, q\right] = \mathcal{D}_{KL}(p, q) + \mathcal{D}_{KL}(q, p) \tag{4}$$

For Gaussian distribution,

$$\mathcal{D}_s\left[\mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i), \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)\right] = \frac{1}{2}Tr\left[\Sigma_i^{-1}\Sigma_j + \Sigma_j^{-1}\Sigma_i\right] + \frac{1}{2}(\mu_i - \mu_j)^T\left[\Sigma_i^{-1} + \Sigma_j^{-1}\right](\mu_i - \mu_j) \tag{5}$$

Our merging algorithm, summarized as Algorithm 2, is graph-based, and similar in spirit to the 'multiclustering' algorithm by Ashlock et al [12] with two key differences, 1) we use EM based mixture modeling instead of K-means for identification of the clusters and 2) we use (symmetric) KL divergence instead of the cluster coherence metric as the distance between clusters as in [12].

The algorithm runs in time $\mathcal{O}(k^4)$ since the number of edges $|E|$ is $\Theta(k^2)$ and for the number of connected components can be found using a depth-first search, which runs in time $\mathcal{O}(k^2)$.

```
Input: $\theta$, k, m
$\theta$: The estimated Gaussian Mixture parameters
k: Number of Gaussian mixture components
m: The final number of clusters after merging
Output: $mergedClusterList$
Represent each of the k components as a vertex of a Graph.
Initially assume the graph is fully connected. Let the weight
of each edges between two vertices be equal to the symmetric
KL divergence between the associated componets.
repeat
    Remove the next maximum weighted edge from the
    current graph.
    Compute the connected components in the graph.
until Number of connected components is m
```

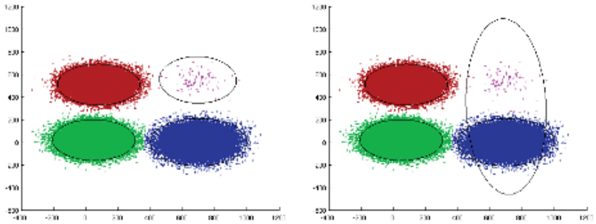
Algorithm 2: Hierarchical cluster merging

3. EXPERIMENTAL RESULTS

In this section, we demonstrate that the proposed method works well on both synthetic data and FC data. Synthetic data allows us to compare against the ground truth and against other algorithms. We note that these comparisons are with smaller datasets since memory and speed limitations do not allow these existing algorithms to be used on typical full size FC datasets. For the actual FC data, we compare our automated clustering results against the manual gating process that is typically employed by biologists. Since this process does not identify an exhaustive set of clusters and because objective criteria for clustering of FC data are difficult to define, our comparison explores the agreement between collections of clusters and populations of interest identified in the manual gating process.

3.1. Experiments on synthetic data

The goal of this experiment is to compare the proposed method with the traditional EM algorithm with respect to speed and the ability to find the smaller clusters. We created synthetic data from a mixture of four bivariate Gaussians with wide variations in their population size, viz., 150,000, 100,000, 50,000 and 150 points respectively. Note that the smallest cluster is 1000 times smaller compared to the largest cluster. We applied the traditional EM and the proposed algorithm on the same dataset. While the proposed sampling based method correctly estimated the parameters of the smallest cluster (see Figure 1), the traditional EM algorithm (running on the full dataset) fails to estimate the parameters correctly. Table 1 presents quantitative comparison results for these algorithms (average over the 20 independent runs) in terms of the average KL divergence between the true and estimated parameters for all clusters and the critical smallest cluster.



(a) Proposed Method

(b) Traditional EM (on full dataset)

Fig. 1. Clusters found by the proposed method and the EM algorithm

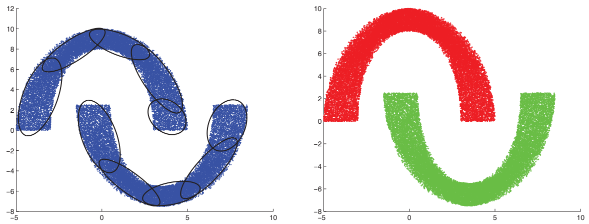
Table 1. Comparison with traditional EM algorithm.

	Proposed Method	Traditional EM
Avg Runtime	41.18 sec	155.08 sec
Avg KL divergence	0.1194	0.3699
Avg KL divergence (smallest cluster)	0.3584	1.3065

Next, we verified that the proposed method works well when clusters are non-Gaussian. Due to space constraints, we only present results on a synthetic data set that has two clusters that are are strongly non-Gaussian and non-convex (Figure 2). After fitting 10 Gaussian components and then merging them down to two clusters using the hierarchical merging method of Algorithm 2, the two clusters can be detected accurately.

3.2. Experiments on flow cytometry data

For our first evaluation on FC data we used the publicly available FICCS test dataset 'FACSAria' (available at `http://www.ficcs.org`). This dataset contains 1 million cells with 20 recorded



(a) After fitting 10 Gaussians

(b) After merging down to 2 clusters

Fig. 2. Clustering horse-shoe dataset with two non-convex clusters

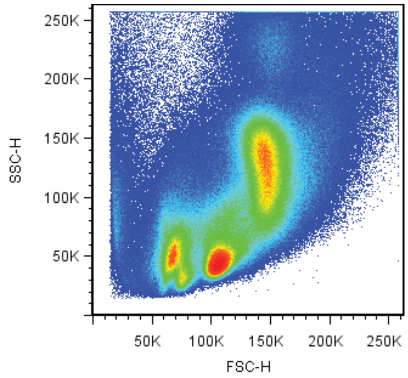
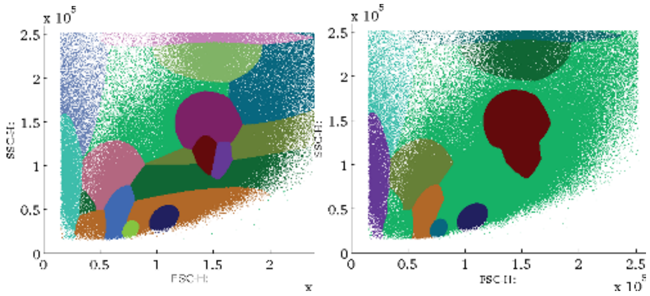


Fig. 3. Density plot for two variables, FSC-H and SSC-H



(a) Fitted 16 Gaussians

(b) After merging

Fig. 4. Clusters produced by the proposed method

variables per cell. Due to space limitations and challenges with high dimensional data visualization, we present the results of the proposed method in a 2D view with respect to two size variables, viz., forward scatter (FSC-H) and side scatter (SSC-H). Figure 3 shows the density of data in these two variables.

Using BIC, we chose to fit 16 Gaussian components to the data. Note the presence of relatively sparse cluster on the left and a very small cluster in the bottom part of the density plot (Figure 3). Even though the method correctly estimated most of the dense elliptic clusters (Figure 4(a)), the uniform background and a non-Gaussian cluster in the middle are divided into multiple clusters. After applying the hierarchical merging (down to 10 clusters), most of the overlapping clusters are merged (Figure 4(b)) and we can see strong visual correspondences between the clusters and the dense regions in Figure 3.

Next we validate the proposed method against manual FC gating performed by a biologist. We have tested with 3 PBMC blood sample datasets among which, one is unstimulated and the other two are stimulated with SEB and the Flu vaccine respectively. Each of these datsets contain 23 dimensions and on average 450,000 cells descriptions. For all the three datasets, the biologist successively gated the FC data using 3 bivariate axes views (of the 23-D data) that corresponded to, 1) The live cells that express negative response to the stain Live-Dead, 2) The CD3+ CD14- cells and 3) The CD4+ CD8- cells. In order to assess the consistency between the manual gating and the proposed automated clustering, the population, selected by each gate or sequence of gates ($\mathcal{G}$) is compared against a union of clusters ($\mathcal{C}$) that have significant (50$\%$ or higher) overlaps with the corresponding gated population. To quantify the agreement between the manual gating process and automated clustering, we use two metrics, $\mathcal{C}_a = \mid \mathcal{C} \cap \mathcal{G} \mid / \mid \mathcal{C} \mid$ and $\mathcal{G}_a = \mid \mathcal{C} \cap \mathcal{G} \mid / \mid \mathcal{G} \mid$ that corresponds roughly to the sensitivity and specificity, respectively. The results are shown in Table 2 and validate that the clustering is in good agreement with the manual selection process.

Table 2. Comparison of proposed method with manual gating on PBMC blood samples with different stimulations.

Stimulation	Live-Dead-		CD3+ CD14-		CD4+ CD8-	
	$\mathcal{C}_a\%$	$\mathcal{G}_a\%$	$\mathcal{C}_a\%$	$\mathcal{G}_a\%$	$\mathcal{C}_a\%$	$\mathcal{G}_a\%$
None	92.80	93.02	87.24	89.42	88.52	91.14
SEB	95.8	97.03	93.65	97.94	90.93	89.79
Flu Vaccine	93.64	93.73	90.78	91.11	87.14	84.13

4. CONCLUSION

In this paper, we propose a scalable algorithm for clustering flow cytometry data. Unlike most existing methods, our algorithm scales to the large data sets typical in Flow cytometry applications. Using both simulated and real data, we demonstrated that the proposed algorithm is able to identify small, sparse clusters in the presence of heterogeneity of cluster sizes, shapes and densities - a situation that is commonly encountered in typical FC applications.

5. ACKNOWLEDGEMENT

High performance computing support from the Center for Research Computing (CRC) at University of Rochester is gratefully acknowledged. The authors also wish to thank Jonathan Rebhahn, Sally Quataert, and Ernest Wang for useful comments and discussions.

6. REFERENCES

[1] M. Brown and C. Wittwer, “Flow cytometry: principles and clinical applications in hematology,” *Clinical chemistry*, vol. 46, no. 8, pp. 1221–1229, 2000.

[2] C. Chan, F. Feng, J. Ottinger, D. Foster, M. West, and T. Kepler, “Statistical mixture modeling for cell subtype identification in flow cytometry,” *Cytometry Part A*, no. 8, 2008.

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

[4] S. Pyne *et al.*, “Automated high-dimensional flow cytometric data analysis,” *PNAS*, vol. 106, no. 21, p. 8519, 2009.

[5] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley Interscience, 2000.

[6] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[7] P. Bradley, U. Fayyad, and C. Reina, “Scaling EM (expectation-maximization) clustering to large databases,” *Microsoft Research Report, MSR-TR-98-35*, 1998.

[8] R. Maitra, “Clustering Massive Datasets With Application in Software Metrics and Tomography,” *Technometrics*, vol. 43, no. 3, pp. 336–346, 2001.

[9] C. Fraley, A. Raftery, and R. Wehrens, “Incremental model-based clustering for large datasets with small clusters,” *Journal of Computational and Graphical Statistics*, vol. 14, no. 3, pp. 529–546, 2005.

[10] J. Baudry, E. Raftery, G. Celeux, K. Lo, and R. Gottardo, “Combining Mixture Components for Clustering,” Univ of Washington, Tech. Rep., 2008.

[11] M. Figueiredo, J. Leitão, and A. Jain, “On fitting mixture models,” *Lecture notes in computer science*, vol. 1654, pp. 54–69, 1999.

[12] D. Ashlock, E. Kim, and L. Guo, “Multi-clustering: avoiding the natural shape of underlying metrics,” *Smart Engineering System Design: Neural Networks, Evolutionary Programming, and Artificial Life*, pp. 453–461, 2005.