Isolating single cell types from co-culture flow cytometry experiments using automated *n*-dimensional gating for CAR T-based cancer immunotherapy

Victor Tieu[1] (vtieu@stanford.edu)
[1]Department of Bioengineering, Stanford University, Stanford, CA

Category: Life Sciences

Introduction

Flow cytometry is a method of single cell analysis where cells are encased in individual microfluidic droplets and run through a series of lasers. The lasers excite either fluorescent proteins within the cell or antibody probes bound to cell markers, and the scattering and intensity of the emitted light is recorded—each cell has a distinctive cell signature comprised of how the cell scatters light (which represents the size of the cell, granularity of the cell surface, and overall cell viability) and the intensity of different colors of emitted light (which represent the relative level of labeled proteins of interest that are present). Filters specific to a range of wavelengths collect light into discrete "channels" with photomultiplier tubes that convert light intensity into voltage readouts, which can be stored by the computer.

Flow cytometry experiments are ubiquitous within the field of cancer immunology, since researchers frequently probe the effect of certain stimuli on gene expression levels, protein expression levels, surface marker expression, etc. of single immune cells, since these properties often determine their downstream activity and function. To analyze flow cytometry data, researchers plot single cells on a 1- or 2-dimensional plot (histogram/dot plot, respectively) and "gate" on a specific cell population of interest by manually drawing linear/polygonal boundaries. (Importantly, these populations usually follow a Gaussian-like distribution.) However, these gates are often arbitrarily drawn, which can lead to inconsistencies in data analysis if researchers are not fully aware of which populations they are selecting, and how they are selecting them. For example, a positive gate that is too stringent might lead to many false-negatives, while a positive gate that is too inclusive might lead to many false-positives.

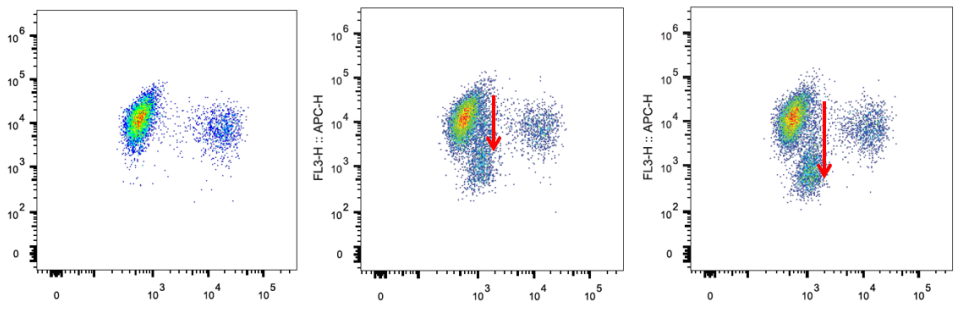

Figure 1. 2D dot plot of flow cytometry data. Each dot represents a single cell. The two axes represent light intensity of a particular color/channel (log10 scale). The overlap in cell populations shown in the middle and right plots makes traditional manual gating using polygonal boundaries impossible, despite clear, visually-distinct populations. (From Victor Tieu, Qi Lab, Stanford University, unpublished data).

Furthermore, while the use of flow cytometry is straightforward for populations of a single cell type, researchers are often interested in collecting single cell data from mixtures of cell types in order to investigate cell-cell interactions. However, gating flow data with mixed cells is challenging, since unique cell "markers" must be present in order to differentiate cell types. In co-culture functional/killing assays that mix CAR T cells (a specific type of immune cell that is artificially engineered to target and kill tumor cells) with tumor cells, the levels of these unique markers often change due to cell-cell interactions (i.e. cells no longer "look" like the unmixed control populations). As a result, if two cell types are not distinctive enough in their fluorescent readout for a given channel, their populations may overlap significantly, making traditional gating impossible (**Fig 1**).

In summary, there is a need for a more reliable gating strategy in flow cytometry data analysis that does not involve manually drawing boundaries in plottable dimensions to isolate populations of interest, which is both limited in its effectiveness at separating semi-distinct cell types, and problematic in its arbitrary nature. A machine learning approach, while not entirely immune to researcher bias, provides a more rigorous and consistent definition of gating cell populations of interest. It follows that an unsupervised learning algorithm that utilizes data across all n channels is appropriate to identify clusters within flow data that represent a single cell type where no reliable ground truth exists. Previously, researchers have trained unsupervised models to identify distinct, potentially novel cell subpopulations within image[1,3] and flow[2,3,4] data. However, there remains an unmet need to "unmix" already-known cell types in co-culture experiments for single-cell analysis after cell-cell interactions have already changed their signature from that of the ground truth. In this project, I evaluated the ability of k-means clustering and Gaussian mixture models, in combination with PCA, to isolate these single cell types, and trained a semi-supervised EM algorithm to do so.

Dataset and features

To train the model, I utilized an unpublished flow cytometry dataset that I collected in lab from a functional assay with mixed primary human CAR T cells and K562 leukemia cells (target cells). I recorded 10000 events (single cell data points) per condition and ran each condition in triplicate (30000 total events per condition). I partitioned the dataset to have 10000 examples (1/3 replicates) for training the unsupervised learning algorithm and tuning hyperparameters, and 20000 (2/3 replicates) for testing. In addition to the co-culture conditions, I also took samples of unmixed human primary T cells and tumor cells to serve as controls for the ground truth labels. All cells were stained using fluorescent antibodies in two colors (anti-CD19 APC and anti-CD8 AF405), and the CAR T cells express mCherry and EGFP through a lentiviral vector. With the forward scatter (FSC-H height, FSC-A area) and side scatter (SSC-A area) lasers, each single cell is represented by a feature vector with seven channels of interest ($n = 7$).

Raw FCS files were converted to CSV using an open-source script provided by GenePattern[5] and the Broad Institute at MIT. The CSV files were then read and manipulated as dataframe objects using the Python pandas library. Each row in the design matrix represents a single cell (example), and each column represents one of seven channel features from the raw input data. Derived features include PCA projections of the input for visualization and clustering due to correlation between channels. Unmixed cells were used as the ground truth-labeled dataset. Simple data pre-processing included shifting all values by the min and taking the log10 of color channels, since the distribution of fluorescence intensity is log10-normal (the distribution for scattered light intensity is Gaussian-like on the linear scale). For PCA, the mean and variance of the dataset were set to 0 and 1, and the eigenvectors used for projection of the unlabeled dataset were also used for the labeled dataset.

Methods and implementation

In order to train the model to assign cluster labels to each cell, I first reduced the dimensionality of the labeled dataset from seven to two dimensions using principal components analysis (PCA). To calculate the covariance matrix Σ:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} x^{(i)^T}$$

Then, the projection of the design matrix to the K-dimensional PCA subspace $Y = XU^T$, where U is the $K \times n$ array of the top K eigenvectors $u_1, u_2, \ldots, u_K$ of Σ sorted by the corresponding eigenvalues. Following PCA, I qualitatively evaluated two unsupervised learning algorithms (k-means clustering and the unsupervised/semi-supervised EM algorithm for Gaussian mixture models) on the ability to cluster single cell types in comparison to the manual gating strategy implemented by hand. For k-means clustering, I initialized centroids μ_j at random training examples, shifted the centroid to the average of the closest proximal training examples, assigned these new centroids as labels $c^{(i)}$ to the closest examples to draw linear separation boundaries for k clusters, and iterated until convergence:

$$\mu_j := \frac{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)} = j\}} \qquad c^{(i)} := \operatorname{argmin}_j \left\| x^{(i)} - \mu_k \right\|^2$$

For the un/semi-supervised EM algorithm, I calculated the weights (probabilities) $w_j^{(i)}$ of each training example belonging to each of k Gaussian distributions in the E-step and updated the parameters μ_j, ϕ_j, Σ_j in the M-step by maximizing the log-likelihood with respect to each of the parameters, iterating until convergence (the supervision term for the labeled dataset is weighted by α) and assigning examples to the most probable Gaussian:

$$w_j^{(i)} := \frac{\mathcal{N}(x^{(i)}; \mu_j, \Sigma_j)\phi_j}{\sum_{l=1}^{k} \mathcal{N}(x^{(i)}; \mu_l, \Sigma_l)\phi_l} \qquad \mu_j := \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)} + \alpha \sum_{i=1}^{\widetilde{m}} \mathbf{1}\{\tilde{z}^{(i)} = j\}\tilde{x}^{(i)}}{\sum_{i=1}^{m} w_j^{(i)} + \alpha \sum_{i=1}^{\widetilde{m}} \mathbf{1}\{\tilde{z}^{(i)} = j\}} \qquad \phi_j := \frac{\sum_{i=1}^{m} w_j^{(i)} + \alpha \sum_{i=1}^{\widetilde{m}} \mathbf{1}\{\tilde{z}^{(i)} = j\}}{m + \alpha\widetilde{m}}$$

$$\Sigma_j := \frac{\sum_{i=1}^{m} w_j^{(i)}(x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T + \alpha \sum_{i=1}^{\widetilde{m}} \mathbf{1}\{\tilde{z}^{(i)} = j\}(\tilde{x}^{(i)} - \mu_j)(\tilde{x}^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)} + \alpha \sum_{i=1}^{\widetilde{m}} \mathbf{1}\{\tilde{z}^{(i)} = j\}} \qquad z^{(i)} = \operatorname{argmax}_j \, w^{(i)}$$

Then, I reassigned the cluster labels in the PCA subspace to the same cells in the original 7-dimensional representation, plotting on a 2D FSC-A/SSC-A dot plot to qualitatively evaluate clustering in comparison to my manual gating strategy (this workflow is shown in **Fig 2** below).

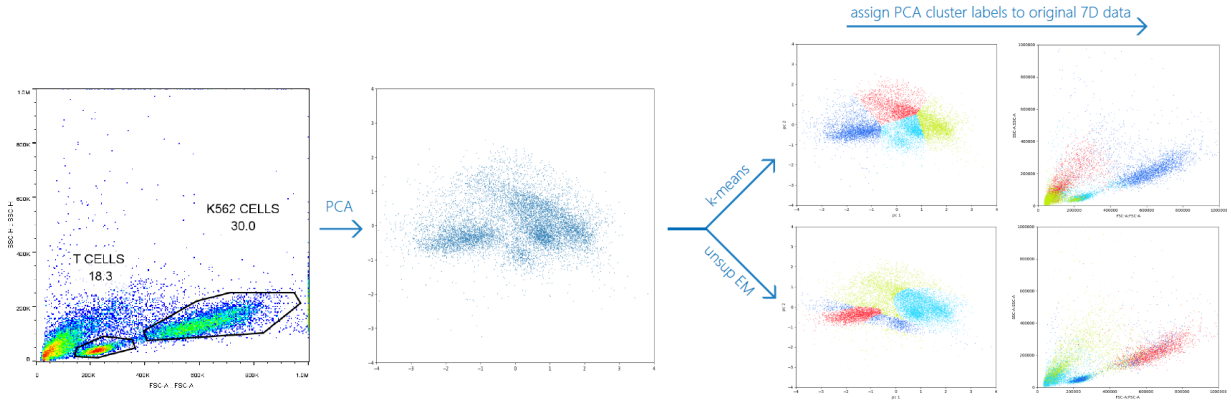

Figure 2. Schematic of workflow. Raw 7-dimensional data for the mixture of T cells and target cells (left, plotted in two-dimensions as FSC-A/SSC-A) is projected onto a two-dimensional subspace by PCA (middle) and clustered using k-means or unsupervised EM (right). Each dot represents a single cell training example. PCA labels are reassigned to original 7-D data (far right) and compared with the manual gating strategy (polygonal boundary drawn by hand) on the original dot plot.

Results

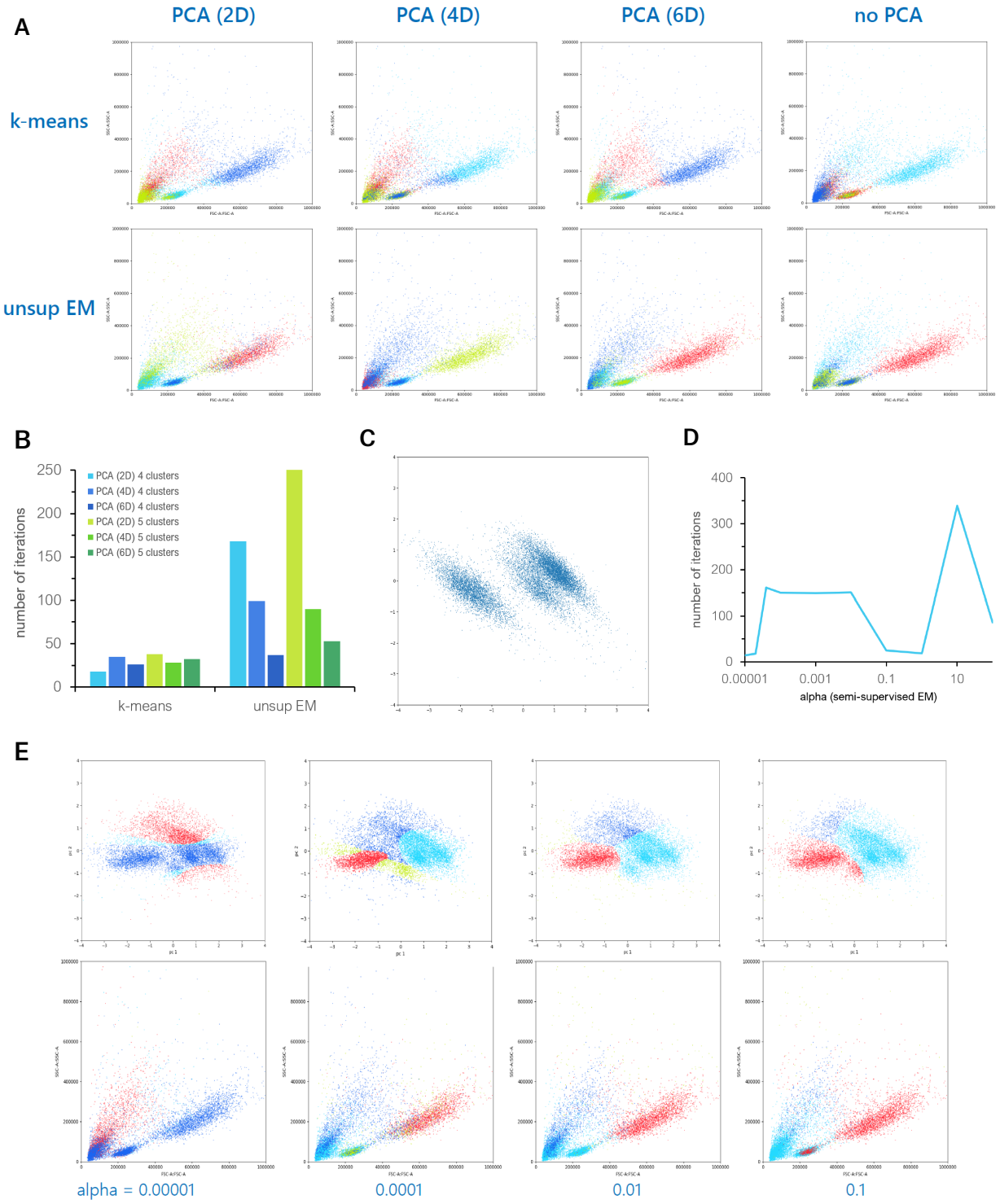

Figure 3. **A.** Clustering results for k-means and unsupervised EM with k=4 clusters and varying PCA subspace dimension. **B.** Number of iterations until convergence for both models, varying the number of PCA principal components or cluster number. **C.** 2D PCA projection of the labeled dataset using eigenvectors generated from the unlabeled set for semi-supervised EM. **D.** Number of iterations until convergence is plotted against varying alpha coefficients for the labeled ground truth examples. **E.** Clustering results for different alpha values (cluster number k=4). Top row: 2D PCA projection; bottom row: 7-D original data.

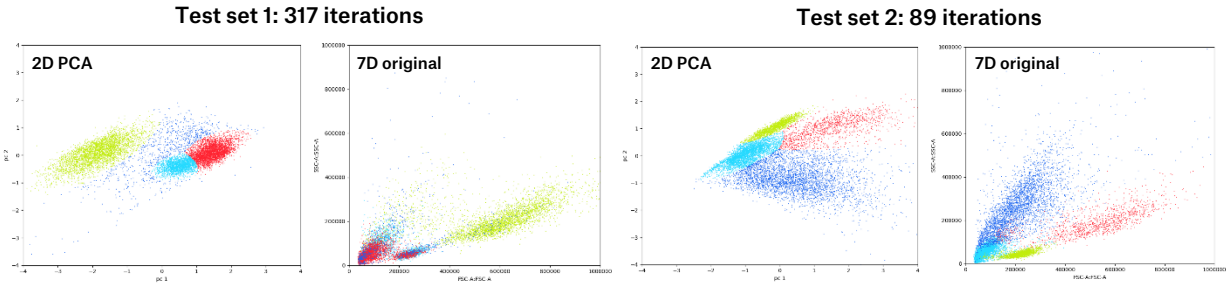

Figure 4. Clustering on the test set. Result of semi-supervised EM clustering on two test datasets (10000 examples each) using the hyperparameters set by the training set ($\alpha = \ 0.00004, K = 2, k = 4$).

Discussion

Qualitatively, unsupervised EM outperforms k-means clustering in comparison to the manual gates set for T cells and K562 cells—this is unsurprising, since (like hand-drawn gates) linear boundaries do not separate cell clusters as well as multivariate Gaussian fits. Quantitatively, the k-means algorithm converges much quicker and with little correlation to principal component number or cluster number, whereas for unsupervised EM, smaller PCA subspace dimension and greater cluster number seem to slow algorithm convergence, though this performance trade-off results in improved clustering quality (**Fig 3A, B**). To improve convergence rate and stability, I introduced a supervised term into the EM algorithm using unmixed, ground-truth labeled cells. However, semi-supervision does not work as well in this model, since interactions between mixed cell populations change how each cell compares to the ground truth in the PCA subspace (note differences in PCA plots in **Fig 2, 3C**). Trying out a range of alpha values resulted in varied clustering quality and convergence rate (**Fig 3D, E**). Interestingly, $\alpha = \ 0.00004$ resulted in the same clusters as the unsupervised case, which suggests that giving the algorithm a very small hint about what the unmixed cells should look like in the PCA subspace helps improve stability and convergence rate, when similar outcomes derived from an unsupervised run may be more susceptible to noise and randomness between runs on identical replicates (high variance). On the other hand, since unmixed cells do not greatly resemble mixed cells, too large of a weighting on the supervised term results in poor quality clustering (**Fig 3E**). Indeed, despite the improved stability and convergence rate of the semi-supervised EM algorithm, clustering on the test sets showed noticeable variance—algorithm performance on test set 2 was remarkably good, though somewhat average on test set 1 (**Fig 4**). The hyperparameters chosen during training (principal component number $K = 2$, cluster number $k = 4$, supervision coefficient $\alpha = 0.00004$) which resulted in overall best performance when evaluated on clustering quality and convergence rate may slightly overfit to the training dataset, however, it is also worth noting that the eigenvectors generated by the initial PCA projection are inherently high variance and are most likely responsible for much of the variation observed between runs on replicates of identical experimental conditions.

Conclusion and future work

I proposed a machine learning approach to automated gating of single cell types. Though mixed cells show large deviations from their unmixed counterparts, through empirical testing I found that the inclusion of a very small weighted supervised term in the EM algorithm successfully improved training performance based on clustering quality and convergence rate. The clustering results on two test sets were highly variable—one converged quickly with impressive clustering, while the other was more lackluster, most likely a result of PCA instability. Performance might be improved by kernelizing, reducing dimensionality with tSNE, or including a skewing factor within the GMM, since many of these shifted distributions are skew-normal[5]. In the long term, it would be useful to train/test the model with mixtures of different cell types to see whether clustering is generalizable beyond a single experimental condition.

References

Source code and project files uploaded to: https://stanford.box.com/s/t1km0vb0ak5v3gkfktuks9j8eyafvl1y

1. Nitta et al. "Intelligent Image-Activated Cell Sorting." *Cell*. 2018 175, 266–276.
2. Lee et al. "Transfer Learning for Auto-gating of Flow Cytometry Data." *JMLR: Workshop and Conference Proceedings*. 2012 27:155–166.
3. Doan et al. "Diagnostic Potential of Imaging Flow Cytometry." *Trends in Biotechnology*. 2018 36, 649-652.
4. Sugár et al. "Misty Mountain clustering: application to fast unsupervised flow cytometry gating." *BMC Bioinformatics*. 11:502.
5. Reich M et al. "GenePattern 2.0." *Nature Genetics*. 2006 38:5, 500-501.
6. Watson et al. "How does flow cytometry express Gaussian distributed biological information?" *Journal of Immunological Methods*. 1985 77, 321-330.

Python packages/libraries

Matplotlib
John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95 (2007).
Numpy
Travis E, Oliphant. A guide to NumPy, USA: Trelgol Publishing, (2006).
Pandas
Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)