2D Visualization of High-Dimensional Molecular Data from Single-Cell Mass Cytometry

Yishun Dong, Diana Wan
Email: {ydong2, jdwan}@ stanford.edu
CS 229 Final Project

I. Introduction

A. Background

Emerging single cell technologies have provided rich, high dimensional datasets. Analysis of single cell data has shed light on various different cellular processes. [1] These data have proved to be particularly useful in many important medical applications. One of the most prominent applications is to detect cancer tumors. These datasets from single-cell mass cytometry often come in the form of large matrices, where each column represents measurements of some parameter of interest, such as the expression level of certain protein. We can treat these measurements as realization of some random variables (up to 40 variables). Therefore, these matrices represent data from some very high dimensional space. While these datasets are of great interest to the cancer community, the challenge comes to developing efficient tools/algorithms that can represent these high dimensional data into lower dimensional subspace (2 or 3 dimensions) where human can digest and visualize effectively. Traditionally, these single-cell datasets have been examined in two dimensions at a time in the form of a scatter plot and inspected/gated by human expert. However, as the number of parameters increases, the number of pairs becomes overwhelming. In addition, a pairwise viewpoint often misses the biological meaningful high dimensional relationships among these datasets. [1]

Various algorithms have been proposed to address this problem. Examples of such algorithms include t-distributed stochastic neighboring embedding (t-SNE) [1] [5], spanning-tree progression analysis of density-normalized events (SPADE) [3] [4], elastic embedding (EE), etc. These stochastic neighbor embedding (SNE) and many other related non-linear manifold learning algorithms have achieved reasonable quality low-dimensional representations of these data by optimally preserving the pairwise Euclidean distance in their original high-dimensional space. [2] In this project, we will examine these algorithms and compare them with the dimensionality reduction algorithm we learned in class - PCA. Furthermore, we explore several meaningful extensions base on the rich single-cell data we have at hand.

B. Goal and Outline

The goal of this project is to apply existing low dimensional visualization algorithms to the rich single-cell mass cytometry datasets obtained from a healthy donor. The data was made publicly available by Bendall et al [1] (see next section for more description on the structure of the datasets). We will both use code provided by Max Vladymyrov (UC Merced) and written by ourselves to transform the high dimensional data into two dimensions in the form of a scatter plot. The resulting two dimensional scatter plots will be shown and compared visually to see how well cells from different populations (cell types) are separated. At the same time, we will develop our own error metric to quantitatively compare how well separated are the data in two dimensions. The algorithms that will be used include the generic Principal Component Analysis (PCA), elastic embedding (EE), t-distributed stochastic neighboring embedding (t-SNE) and symmetric stochastic neighboring embedding (s-SNE).

Due to the richness of the datasets provided by [1], there are many interesting extensions possible for this project. We will explore a few. In particular, the 30 to 40 or so parameters in the original dataset can be roughly divided into two classes (for the purpose of this project). One class of variables represents the protein expression levels of some surface markers, whereas the other class of variables represents the expression level of some intracellular markers. One way to refine the dimensionality reduction algorithm and potentially getting better classification results is to use (1) only the surface markers, (2) only the intracellular markers, (3) both surface and intracellular markers, and compare the results from these three sets. In addition to having data from different cell types available to us, we also have the data with different conditioning. In this project we will explore the basal (nothing added) condition versus the PVO4 condition. From a biological point of view the PVO4 condition will not have any effect to the parameters corresponding to the surface markers, but for intracellular markers PVO4 condition is expected to produce some effect to the data. So we will see how PVO4 condition affects our classification results.

C. Data

The data for this project were downloaded from the publicly available Cytobank website, https://www.cytobank.org/cytobank/experiments/6033/illustrations/54573. When downloading data files, there were various settings we needed to select. First of all, we can select conditions to be applied to the cell populations. In this project, we selected the ”Basal” condition and the ”PVO4” condition, as their effects will be compared later. We use cell data coming from the same individual subject in this project. We selected a set of 20 cell populations to be studied in this project. Due to the speed of some embedding algorithms, we will only train on a subset of P populations among these 20 populations to obtain a common 2-dimensional subspace. Furthermore, when we make the scatter plot and examine classification results, we will only scatter plot a subset of these P populations to see separation and how well classification is done. For a fixed individual, condition, and cell population, there is a single corresponding .fcs file, which consists of a data matrix of the form:

$$A_{c,p} = \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix} \tag{1}$$

where $c \in \{\text{Basal}, \text{PVO4}\}$, $p \in \{1, 2, ..., 20\}$ is the population identifier. Each column $a_j \in \mathbb{R}^{m_{c,p}}$, where $m_{c,p}$ is the number of samples for that particular cell type and condition (usually about 10^2-10^4), n is the number of parameters in the dataset (about 40). The parameters are also selectable when downloading the data. In short, $A_{c,p}$ contains the high dimensional data we would like to reduce and visualize.

II. Results and Discussion

A. Preprocessing

In the "Data" section, we described the form of the data matrix. In this section, we will outline the preprocessing we performed on the data matrices that are used in the subsequent sections. These preprocessing are very important in order to obtain meaningful results.

(1). *Extract relevant columns* - each data matrix downloaded contains 41 columns, but not all of them contain relevant information. In particular, there are 10 columns containing either information irrelevant to classifying cell types (such as time, event number, etc) or duplicated information (redundant data that are summarized by other columns). So these 10 columns are completely ignored. For the rest 31 columns, they can be divided into two classes. One class consists of 13 columns containing measurements from **surface markers**, while the other class consists of 18 columns containing measurements from **intracellular markers**. For example, in the subsequent sections, when we perform training algorithm on surface markers only, we mean that we only use the 13 columns of the data corresponding to the surface markers.

(2). *Data subsampling* - due to the difference in the abundance of different cell types in the subject, distinct cell types have vastly different sample sizes, i.e. $m_{c,p}$. In order not to let certain cell type dominate the training, it makes sense to fix an m, such that we take m samples (randomly) from each cell type. One drawback of this preprocessing is that the sample size m is limited by $m \leq \min\limits_{c,p} m_{c,p}$. As it turns out, even the smallest $m_{c,p}$ is on the order of hundreds, so we still have a total of a few thousands of data to train, which seems to be enough for this project.

(3). *arcsinh transform data* - after extracting relevant columns and a subset of rows, the data first go through an arcsinh transformation as follows:

$$x^{(i)} = \text{arcsinh}\left(\frac{x^{(i)}}{5}\right) \tag{2}$$

this transformation is very standard when researchers deal with data from flow cytometry. The main reason for this transformation is to reduce the distortion caused by outliers while emphasizing the distances around origin. In short, this transformation makes it easier to compare distances.

(4). *Normalize the data* - since we are using dimensionality reduction algorithms here, as pointed out in the course lecture note on PCA, we need to first normalize the data to have zero mean and unit variance. For the other algorithms, we perform similar normalization so the data are on the same scale. This allows us to see the intrinsic variability among different cell types rather than simply look at the noise.

B. Error Metric

In this project, we are mainly dealing with supervised learning, since we know the class label (cell type) ahead of the time. After the high dimensional data is reduced to two dimensions, in addition to scatter plotting the data and visually seeing how well separated different types of cells are, we establish the following metric for computing the fraction of misclassified cells.

Step 1: compute the centroid of each cluster

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} z_j^{(i)} \tag{3}$$

where j is the index for the cell type being clustered. $z_j^{(i)} \in \mathbb{R}^2$ is the low dimensional representation of the original high dimensional data $x_j^{(i)} \in \mathbb{R}^n$.

Step 2: relabel each cell to the nearest centroid

$$y_j^{(i)} = \underset{j'}{\text{argmin}}\, ||z_j^{(i)} - \mu_{j'}||_2 \tag{4}$$

here j is the original label, $y_j^{(i)}$ is the new label.

Step 3: count the fraction of cells being mislabeled

$$\epsilon_j = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\left\{y_j^{(i)} \neq j\right\} \tag{5}$$

Intuitively, the higher fraction the cells being misclassified, the poorer are the data being separated. However, this error metric does not capture all the information about how well separated the data are. Therefore, we still show most of the scatter plots for visual comparison.

C. Linear algorithm - PCA

The first algorithm comes to mind for dimensionality reduction is the principal component analysis (PCA) we studied in the class. Matlab has built-in function **princomp** for PCA, which is what we will use in this project. As a starting point, PCA provides a simple first pass to address this problem, and will give and confirm intuition on the appropriate parameter sets to use. Figure 1 - 3 shows the resulting low dimensional scatter plots of the high dimensional preprocessed datasets using only the 13 surface markers, only the 18 intracellular markers, and all 31 surface and intracellular markers, respectively. As expected [1], the intracellular parameters do not provide useful information about different cell populations and it is the surface markers that really distinguish different cell types. Visually, Figure 1 shows that using just the surface markers, the five populations separate reasonably well under PCA, whereas after adding the intracellular markers (which seem to be purely noise here), the resulting datasets do not separate nearly as well (see Figure 3). This is further quantitatively confirmed in Table I using the error metric we established earlier. We see that the misclassification error increased for all five cell populations after intracellular markers were added.

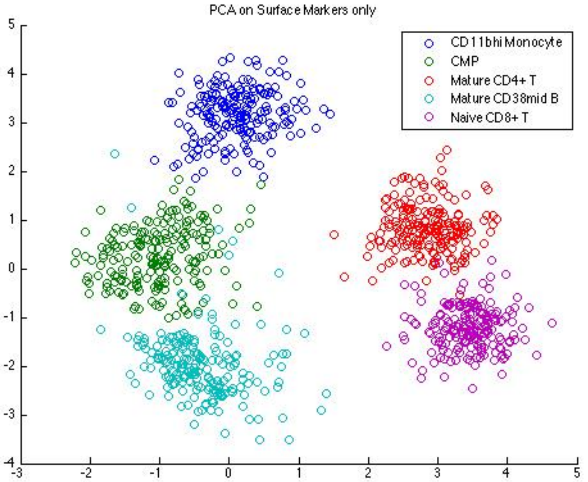

Fig. 1. Use principal component analysis (PCA) to reduce dimensionality of cell data using parameters from surface markers only.

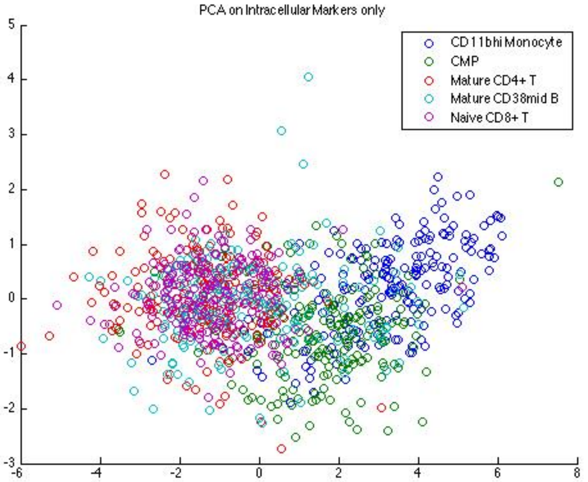

Fig. 2. Use principal component analysis (PCA) to reduce dimensionality of cell data using parameters from intracellular markers only.

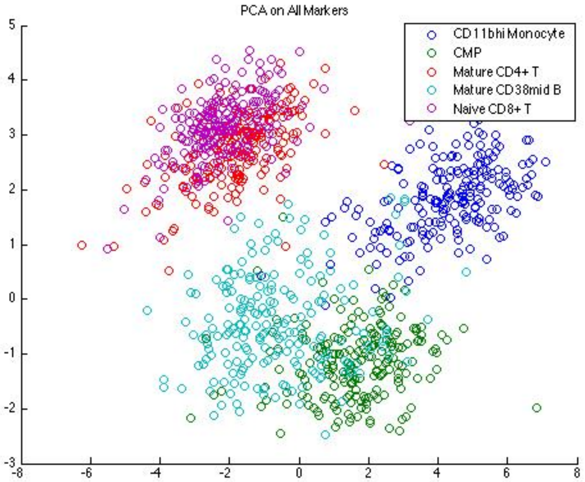

Fig. 3. Use principal component analysis (PCA) to reduce dimensionality of cell data using parameters from both surface and intracellular markers.

TABLE I
CLASSIFICATION ERROR USING PCA

Cell Type	Surface Markers Only	All Markers
CD11bhi Monocyte	0.0%	7.5%
CMP	4.5%	16.0%
Mature CD4+ T	0.5%	33.0%
Mature CD38mid B	4.0%	23.5%
Naive CD8+ T	2.0%	28.5%

D. Other algorithms - t-SNE, s-SNE, EE

It can be seen that PCA provides reasonable separation and is a reasonably effective algorithm for reducing the dimension of high dimensional single-cell cytometry data. However, it has been pointed out that for high dimensional flow cytometry datasets, linear methods such as PCA fail to capture the high dimensional relationship among the cell datasets. [2] Therefore, many non-linear methods have been developed for this purpose, and we shall explore them in this section.

We discovered that data from intracellular markers are pretty much just noise from the PCA results. They cause worse separation when they are added. Therefore, in studying the effectiveness of the other three algorithms, we show only the scatter plots of low dimensional representation of data from surface markers only (Figure 4 - 6). It can be seen visually that all three algorithms give very good separations. Numerically, Table II shows that they all give smaller (in fact, close to zero) misclassification error than PCA. Visually, it looks like the most cutting-edge ([1][5]) tSNE is the best among all four algorithms; followed by sSNE and EE giving roughly the same performance; whereas PCA is slightly worse than the three algorithms in this section.

E. Conditions - Basal vs. PVO4

In this section, we explore how the four algorithms perform when different conditions are applied. In particular, we examine

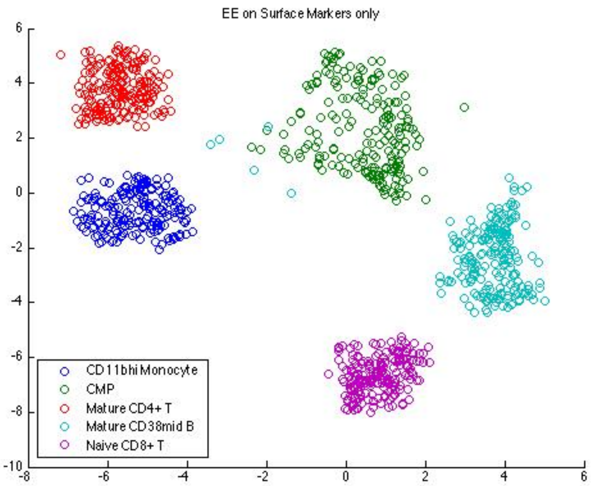

Fig. 4. Use elastic embedding (EE) to reduce dimensionality of cell data using parameters from surface markers only.

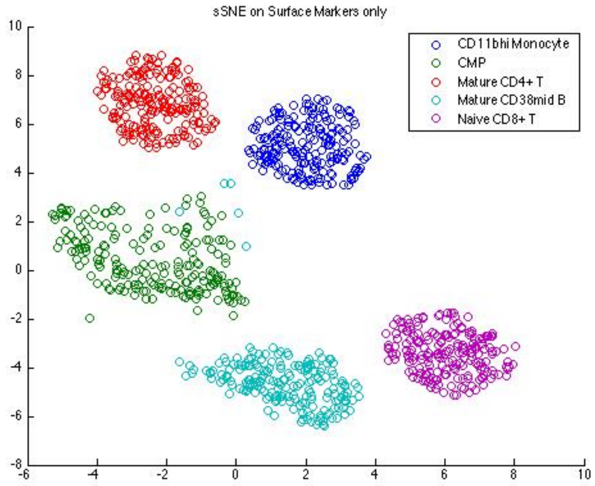

Fig. 5. Use symmetric stochastic neighbor embedding (sSNE) to reduce dimensionality of cell data using parameters from surface markers only.

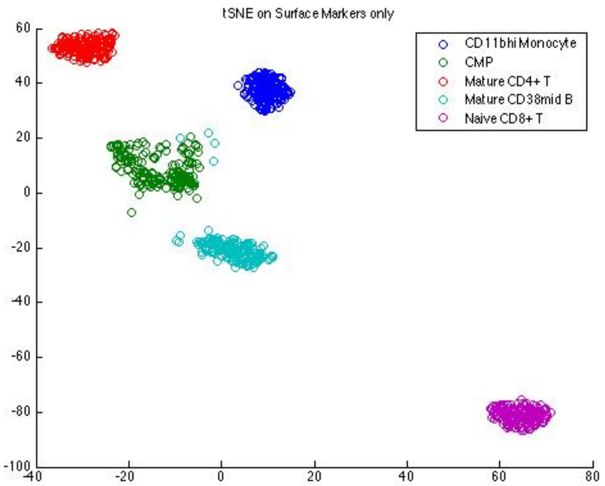

Fig. 6. Use t-distributed stochastic neighbor embedding (tSNE) to reduce dimensionality of cell data using parameters from surface markers only.

TABLE II
CLASSIFICATION ERROR USING FOUR DIFFERENT ALGORITHMS ON SURFACE MARKERS ONLY

Cell Type	PCA	EE	sSNE	tSNE
CD11bhi Monocyte	0.0%	0.0%	0.0%	0.0%
CMP	4.5%	0.5%	0.5%	0.0%
Mature CD4+ T	0.5%	0.0%	0.0%	0.0%
Mature CD38mid B	4.0%	2.5%	2.5%	2.5%
Naive CD8+ T	2.0%	0.0%	0.0%	0.0%

what happens to the cell cytometry data when Basal (no condition) and PVO4 condition are applied to the cell being measured. From a biological point of view, the measurements corresponding to the surface markers should not change in a fundamental way, i.e. the random variable remains the same, when different conditions are applied. But the measurements corresponding to the intracellular markers will change fundamentally depending on which condition is applied. Therefore, if we only look at the scatter plots corresponding to surface markers (as we did in the previous section), we will not get meaningful results. Since we will just be looking at two different sets of realizations of the same random variables. On the other hand, based on the PCA results, we expect the two dimensional scatter plot of just the intracellular markers to be a complete mess that only resembles some random noise. Therefore, it makes sense to examine and compare the scatter plots when the algorithms are applied to all markers. (If we still want to think of intracellular data as some kind of "noise", then we can think of different conditions will cause some systematic shift in the "noise" such that we expect them to affect the resulting scatter plots of the 2D representation of data from all markers.)

Figure 7 shows the resulting scatter plots for basal condition, and Figure 8 shows the resulting scatter plots for PVO4 condition. Interestingly, we see the shapes of the clusters formed by different cell populations are indeed quite different for Basal (round) and PVO4 condition (banana-shape). Numerically from Table III, it looks like PVO4 condition gives uniformly worse average classification error. But we believe that is just due to the nature of the shape of the clusters, as round shape clusters are naturally less likely to be misclassified compared to longer and more-stretched clusters as in the PVO4 case. Visually, we see that the three algorithms (EE, sSNE and tSNE) still give quite good separation even using data from all markers. In contrast, the result for PCA shows much poorer separation (the errors are in the range of $20\% - 30\%$). This shows these modern nonlinear methods are more robust against error in reducing cell cytometry data compared to the generic PCA.

TABLE III
CLASSIFICATION ERROR USING FOUR DIFFERENT ALGORITHMS ON ALL MARKERS WITH BASAL CONDITION AND PVO4 CONDITION

	PCA		EE		sSNE		tSNE	
Cell Type	Basal	PVO4	Basal	PVO4	Basal	PVO4	Basal	PVO4
CD11bhi Monocyte	7.5 %	21.0%	0.0%	8.5%	0.0%	9.0%	0.0%	2.5%
CMP	16.0%	38.0%	2.5%	10.5%	1.5%	11.5%	1.0%	8.0%
Mature CD4+ T	33.0%	36.5%	0.0%	2.5%	0.5%	1.0%	0.0%	0.5%
Mature CD38mid B	23.5%	26.0%	1.0%	2.0%	2.0%	2.0%	2.5%	1.5%
Naive CD8+ T	28.5%	21.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Average	**21.7 %**	**28.6 %**	**0.7 %**	**4.7 %**	**0.8 %**	**4.7 %**	**0.7 %**	**2.5 %**

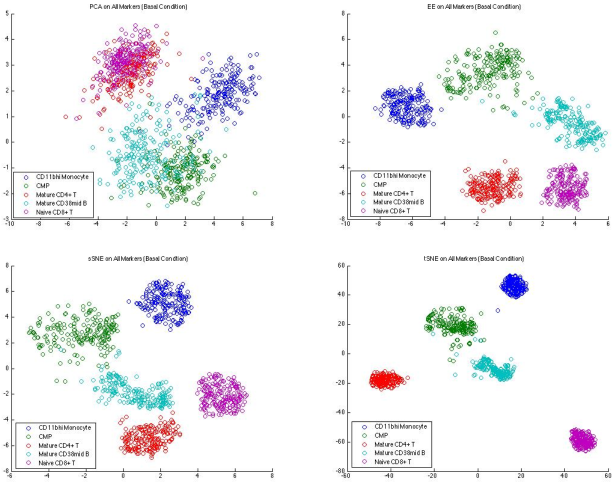

Fig. 7. Dimensionality reduction on all markers with Basal condition using all four algorithms: PCA, EE, sSNE, tSNE.

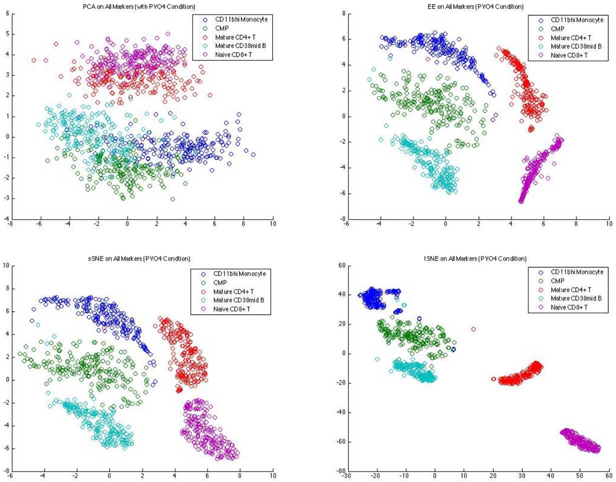

Fig. 8. Dimensionality reduction on all markers with PVO4 condition using all four algorithms: PCA, EE, sSNE, tSNE.

III. SUMMARY AND CONCLUSION

In this project, we applied four dimensionality reduction algorithms to the high dimensional single-cell cytometry datasets so that we can visualize the data in the form of a 2D scatter plot. We observed that all four algorithms give very good separation, hence low dimensional representation, when only the data from surface markers are used. We found out that data from intracellular markers are mainly just noise in separating different cell populations. Among the four algorithms, we conclude that tSNE gives the best performance followed by EE and sSNE. The generic PCA seems to perform poorer and is less robust against noise in comparison for this application. Nonetheless, as a quick and dirty first pass to the problem, PCA performs reasonably well. Finally, we examined what happens to the cell cytometry data when different conditions are applied. We conclude that the different conditions cause systematic shifts in the data from intracellular markers. As a consequence, when 2D representation of high dimensional data from all markers were scatter plotted, the clusters from different cell populations form clusters with different shapes.

ACKNOWLEDGEMENT

We give special thanks to Karen Sachs and Andrew Gentles for giving us an interesting and approachable project to work on and providing us helpful suggestion and guidance throughout the project. We would also like to thank Max Vladymyrov (UC Merced) for sharing and allowing us to use his Matlab code for efficient non-linear methods in this project.

REFERENCE

[1]. Amir E.A., Davis K.L., Tadmor M.D. et al. “viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia.” *Nat Biotechnol.* 31, 545 - 52 (2013)

[2]. Vladymyrov, Max and Carreira-Perpinan, Miguel A. “Partial-Hessian strategies for fast learning of nonlinear embeddings.” *ICML 2012* , pp. 345-352, Edinburgh, Scotland, Jun. 26 - Jul. 1 2012.

[3]. Bendall, S.C. et al. “Single-cell mass cytometry of differential immune and drug responses across the human hematopoietic continuum.” *Science* 332, 687-696 (2011).

[4]. Qiu, P. et al. “Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE.” *Nat. Biotechnol.* 29, 886-891 (2011).

[5]. Laurens van der Maaten, Geoffrey Hinton. “Visualizing Data using t-SNE.” *Journal of Machine Learning Research* 9, 2579-2605 (2008).