2D visualization of high-dimensional cellular protein data for cancer detection

Amit Badlani
Dept. of Electrical Engineering Stanford University
abadlani@stanford.edu

Andre Esteva
Dept. of Electrical Engineering Stanford University
andre.esteva@gmail.com

Anand Sampat
Dept. of Electrical Engineering Stanford University
asampat@stanford.edu

Abstract

1 Introduction and Problem Statement

In the field of cancer immunology, scientists use the protein content of immune system cells as a way to identify a cell's corresponding type. For example, immune system cells, which are contained in bone marrow, are comprised of a variety of cell types, and to a large degree, each type is uniquely identifiable by both intracellular proteins (IP) and surface proteins (SP). Highly sophisticated methods have been developed that process cells and return information on the types and quantities of proteins expressed in those cells. This data can then be viewed by an expert in the field and categorized. The laborious process of viewing the different dimensions of protein expression and labelling a cell to be of a particular type is known as gating.

Is there order to this data? Does cell type, as a function of protein expression, have some structure in multi-dimensional protein space which can be understood and segmented? Can this data be mapped onto a different multi-dimensional space (MDS) where structure is better defined? These are questions of interest to cancer immunologists. If a well-defined structure exists which separates healthy immune system cells from other types of cells, such structure could be leveraged to identify cancer cells.

Cells evolve. Stem cells become progenitor cells, which become monocytes, etc. As they change, so do the types and quantities of proteins that they express. This leads to a shifting of their representation in some MDS, which can be tracked and understood.

The final goal of this project is two-fold:

- Develop machine learning algorithms that probabilistically tag single bone marrow cells as cancerous after being trained on the protein expression levels of healthy immune system cells.
- Understand the underlying structure in MDS of cell types as identified by their protein content.

2 Data Representation and Significance

There are two techniques which are used in the medical field to get the protein data for each of the cell types. These are flow cytometry and mass cytometry and they are described below.

Flow cytometry is a laser-based, biophysical technology employed in cell counting, cell sorting, biomarker detection and protein engineering, by suspending cells in a stream of fluid and passing them by an electronic detection apparatus. The flow cytometers are used for this purpose, which are able to analyze several thousand particles every second, in "real time," and can actively separate and isolate particles having specified properties.

Mass cytometry or CyTOF (DVS Sciences) is a single-cell multiparametric protein detection technology based on inductively coupled plasma mass spectrometry. It is an extension of flow cytometry in which antibodies are tagged with isotopically pure rare earth elements allowing simultaneous measurement of greater than 40 parameters while circumventing the issue of spectral overlap. Mass cytometry data is recorded in tables that list, for each cell, the signal detected per channel, which is proportional to the number of antibodies tagged with the corresponding channel's isotope bound to that cell.

All this data is formatted and stored in FCS files. We use an FCS reader that converts the data in these files into matrices that can be operated on.

In the interest of better labeling, biologists consider data from normal, unstimulated cells, as well as "super-stimulated" cells with overly-expressed protein levels. In our analysis, Basal cells refers to unstimulated cells whereas PVO4 refers to the stimulated ones.

Table 1 shows the different names of the cells and their respective cell types.

Table 1: Cell names and their respective cell types

PART	DESCRIPTION
HSC	Stem Cells/Progenitors
MPP	Stem Cells/Progenitors
CMP	Stem Cells/Progenitors
GMP	Stem Cells/Progenitors
MEP	Stem Cells/Progenitors
Plasma	B Cells
pre-B-I	B Cells
pre-B-II	B Cells
Immature B	B Cells
Mature CD38 low B	B Cells
Mature CD38 mid	B Cells
Mature CD4 + T	T Cells
Mature CD8 + T	T Cells
Naive CD4 + T	T Cells
Naive CD8 + T	T Cells
NK	NK
Plasmacytoid DC	pDC
CD11B - Monocyte	Monocytes
CD11B high Monocyte	Monocytes
CD11B mid Monocyte	Monocytes

The flowchart in Figure 1 shows how the different cell types evolve. The HSC are the first type of cells which divide and form the MPP cells. The MPP cells further divide and form LMP cells as well as GMP cells. The GMP cells further evolve into pDC and Monocytes, whereas the LMPs evolve produce B-cells, T-cells and NK type cells.

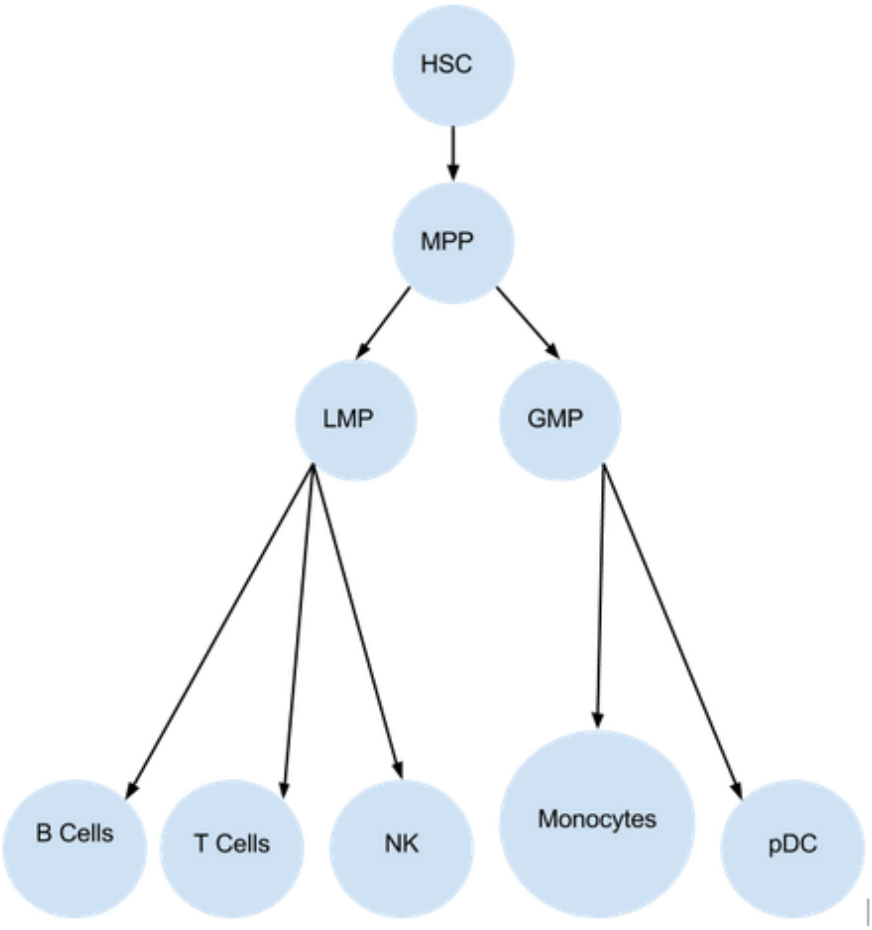

Figure 1: Cell progression

3 Machine Learning Techniques and Application

Our dataset is essentially a large matrix where entries in the matrix refer to a metric for how much a given protein is represented in a particular cell. Specifically the data matrix has dimension $n \times p$ where n is the number of cells considered and p is the number of proteins for which expression levels have been monitored across cells. Thus, each cell is represented by a point in $\mathbb{R}^p$. In order to represent this in $\mathbb{R}^2$ or $\mathbb{R}^3$ for visualization, we'll use various linear and non-linear techniques to map points in the original space $\mathbb{R}^p$ to a different space $\mathbb{R}^k$ which can be projected down to $\mathbb{R}^2$ or $\mathbb{R}^3$ while minimally compromising the structure behind the data.

In the following, we describe algorithms that we intend to use in our final analysis of the data, including principal component analysis (PCA), which has already been successfully implemented.

3.1 Methods

3.1.1 Principal Component Analysis (PCA)

In order to run this algorithm on our dataset $\{x^{(i)}; i = 1, \ldots, m\}$, we'll consider the two cases of $x^{(i)} \in \mathbb{R}^p$ and $x^{(i)} \in \mathbb{R}^s$ with $s < p$. In the first case we'll use all proteins (i.e. all columns) in the dataset, whereas in the second, we'll only consider surface proteins expressed on cell membranes, which are biologically more permament and thus better indicators of cell type.

In particular each $x_j^{(i)}$ is a measure of how much the protein j is represented in cell i. Since we cannot visualize this data of i cells in $\mathbb{R}^p$ we use PCA to capture variance data by converting these $x^{(i)}$'s into principal components (i.e. new unitless axes that contain most of the information within just a first few principal components).

We preprocess the data by subtracting the mean and normalizing the values by their standard deviation. The mean centering is key to ensure the first principal component doesn't just represent the mean and the normalizing ensures each protein is weighted equally in the algorithm (going forward we may want to weight them individually as some proteins may be more relevant in determining certain cell types).

Finally, we apply PCA, which finds the k eigenvectors $u_1, \ldots, u_k$ of the covariance matrix $\frac{1}{m}\sum_{i=1}^{m} x^{(i)}x^{(i),T}$ and then projects $x^{(i)}$ onto the $\{u_j\}$. The points on the new set of k axes (i.e. the principal components) are then defined as:

$$y^{(i)} = \begin{bmatrix} u_1^T x^{(i)} \\ \vdots \\ u_k^T x^{(i)} \end{bmatrix} \in \mathbb{R}^k$$

Visualization in 2D and 3D thus corresponds to plotting the points $u_1^T x^{(i)}$, $u_2^T x^{(i)}$, (and) $u_3^T x^{(i)}$.

3.1.2 Classical Multidimensional Scaling (CMDS)

Similar to PCA, MDS aims to replot each data point in a reduced dimension. As above, assume we want to reduce points in $\mathbb{R}^m$ to $\mathbb{R}^k$. Specifically we are considering $k = 2$ and $k = 3$. MDS then outputs an embedding of points $x_1, \ldots, x_j \in \mathbb{R}^k$ where j is the number of cells (i.e. number of samples).

For any two of the j given cells x_a and x_b in the original space $\mathbb{R}^m$we define a distance $\delta_{a,b}$ between the two points in the higher dimension. In classical CMDS, this is just the Euclidean distance between the points (e.g. $\sqrt{(x_{a,1} - x_{b,1})^2 + (x_{a,2} - x_{b,2})^2}$ for $\mathbb{R}^2$). Thus the constraint on the embedding of points $x_1, \ldots, x_j \in \mathbb{R}^k$ is that $||x_a - x_b|| \approx \delta_{a,b}$. In other words we want to solve the optimization problem with the constraint:

$$\min_{x1,\ldots,x_j} \sum_{a<b} (||x_a - x_b|| - \delta_{a,b})^2$$

3.1.3 Stochastic Neighbor Embedding (SNE)

Single cell data is characterized by various non-linear relationships as well, thus we look to non-liner methods to give us better results [2]. This is the first non-linear method we will try, however, unlike other non-linear methods, SNE, as suggested by the name, is a probabilistic approach that aims to preserve the distribution of neighbor identities. SNE, rather than just matching Euclidean distances, SNE searches for a set of k lower-dimensional vectors who's probability distributions best match the probability distributions over all potential neighbors of points in the higher dimension. [1]

More specifically, if we consider all examples $x_1, \ldots, x_j \in \mathbb{R}^D$ where D is the dimension of each of the rows (i.e. how many proteins for which we have metrics) and similar to PCA we consider a set of $y_1, \ldots, y_j \in \mathbb{R}^k$ where k is the reduced dimension of proteins, SNE iteratively finds the value of $y_1, \ldots, y_j$ that minimizes the difference in probability distribution of neighbors. In short, we initialize some lower dimensional set of vectors $y_1, \ldots, y_j$, then loop through each x_a and y_a, find the neighbors of each, calculate the pairwise probability that x_a chooses x_b as a neighbor p_{ab} and that y_a chooses y_b as a neighbor q_{ab} with a fixed variance. Finally we compare each p_{ab} to each q_{ab} with a Kullback-Leiber divergence ad create a cost function that we then minimize and who's gradient we use to update the vectors $y_1, \ldots, y_j$ using gradient descent $y^{(c+1)} = y^{(c)} - \eta^{(c)}\nabla J^{(c)}$. SNE has many variations as the different probability distributions will yield different results. In particular, we will study a variation called t-SNE which differs from SNE in two ways. One, it uses a symmetric cost function and gradient and two, it uses a student t-distribution to calculate q_{ab}. Although we don't know that this method will work better on our data, prior work has been

published that suggests it may improve our results and improve the time to convergence. Thus, it is the next logical next step [2][3].

4 Current Results

Below we present a PCA analysis of the Basal data. Since surface proteins are more standard proteins for cell type identification, we compare PCA run on surface protein data to PCA run on all protein data. Additionally, we show PCA for all cell types as well as PCA for individual subsets of cells (i.e. B Cells, T Cells, Monocytes, and Stem Cells).

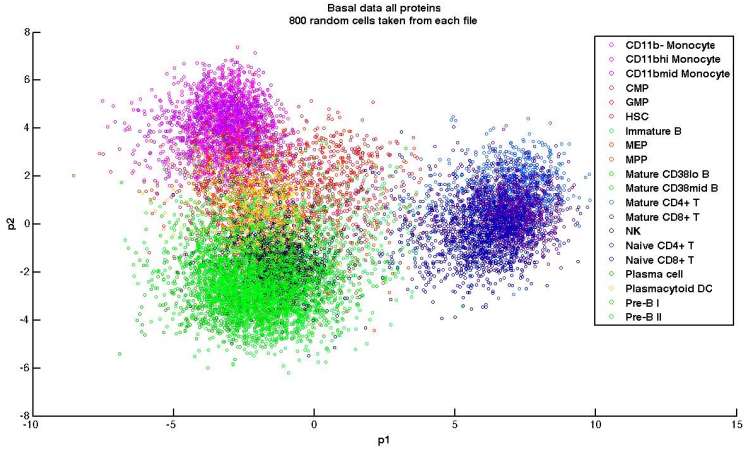

Figure 2: Basal, all proteins, all cell types

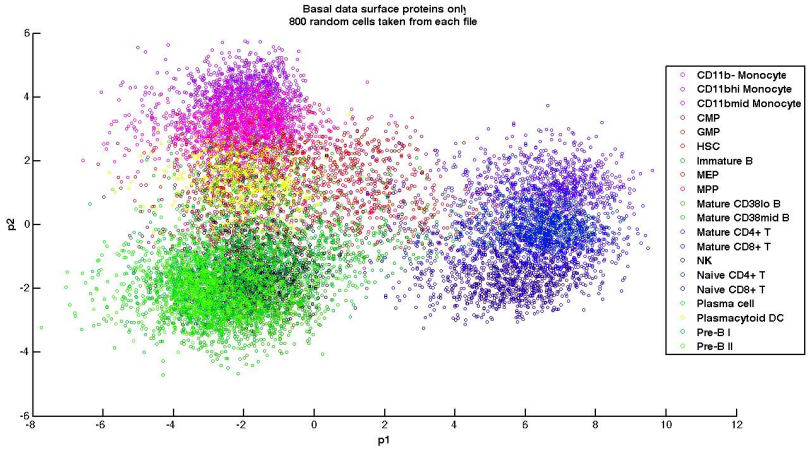

Figure 3: Basal, surface proteins, all cell types

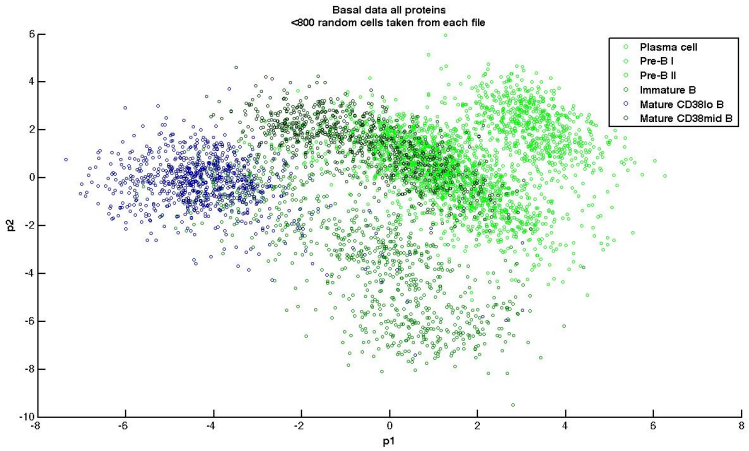

Figure 4: Basal, all proteins, B cells

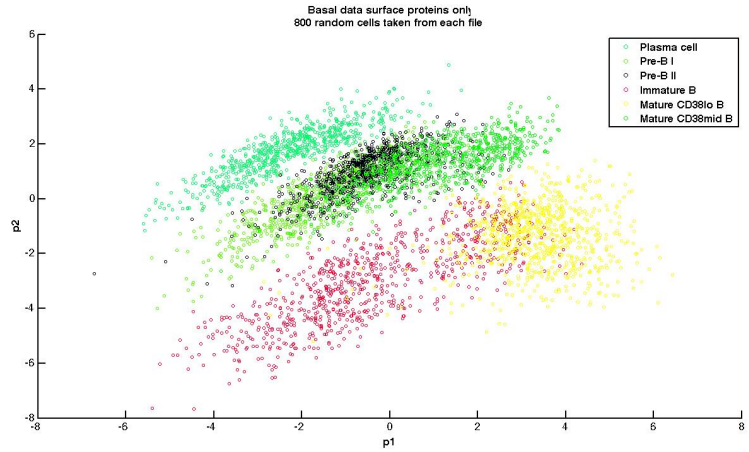

Figure 5: Basal, surface proteins, B cells

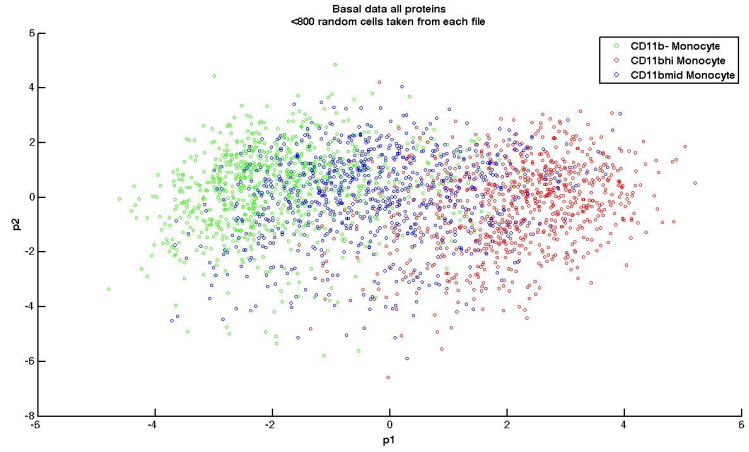

Figure 6: Basal, all proteins, Monocytes

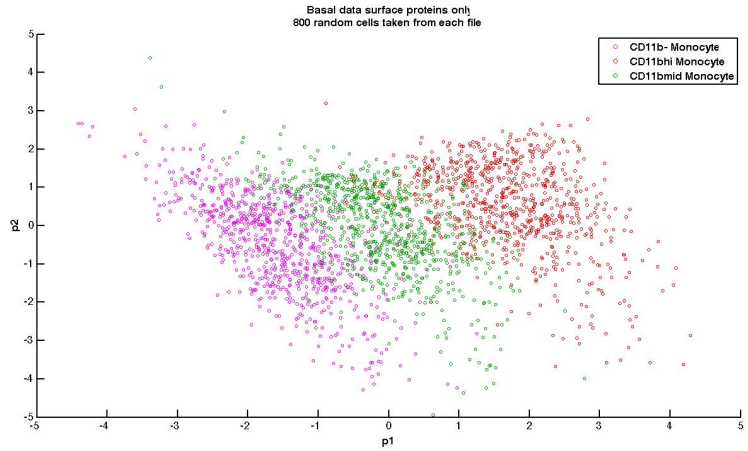

Figure 7: Basal, surface proteins, Monocytes

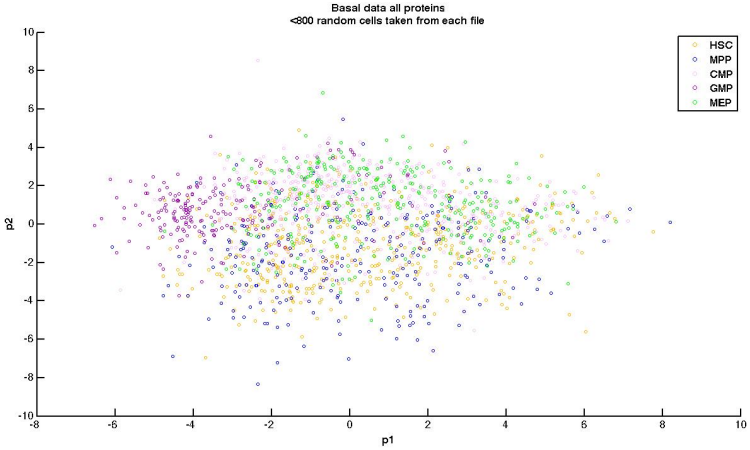

Figure 8: Basal, all proteins, Stem Cells

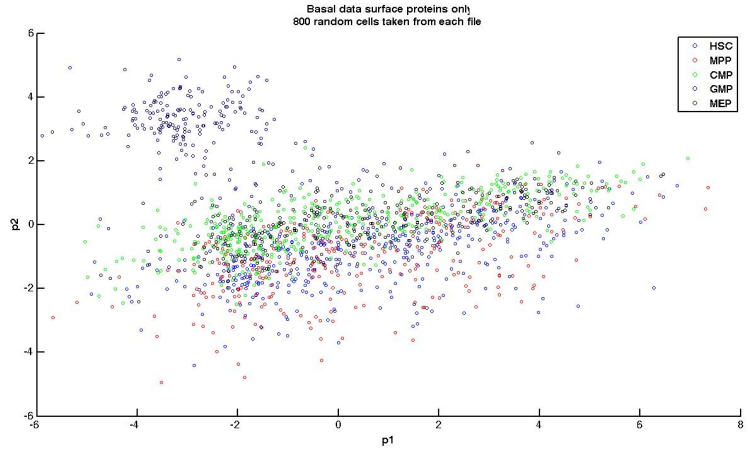

Figure 9: Basal, surface proteins, Stem Cells

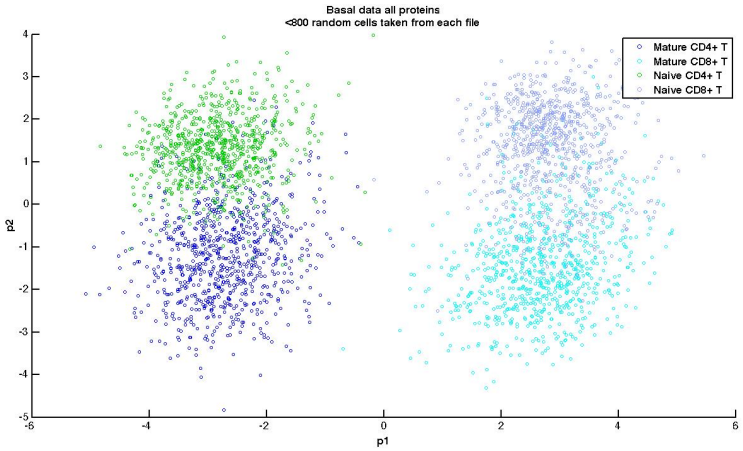

Figure 10: Basal, all proteins, T Cells

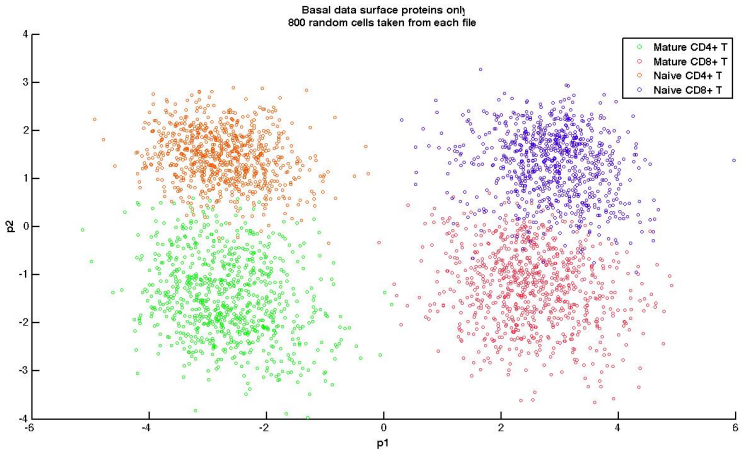

Figure 11: Basal, surface proteins, T Cells

5 Conclusion / Looking Ahead (End of Project Goals)

Currently we have observed dimensional reduction using PCA and have compared various plots to determine qualitatively how well we can differentiate between the various cells. Specifically, we have been given labelled cell data that characterizes each cell type (and sub-cell type) by its protein characteristics. Our job is to take this labelled data and ensure that we can differentiate between each cluster. That way, once we remove the labels we can run the same trained algorithm on cancer cell data to establish whether a cell is anomalous (i.e. doesn't fit within a cluster) or whether it is normal.

To ensure good classification we have two jobs ahead of us:

- determining a quantitative measure that characterizes how well our algorithm differentiates cells
- using that metric to compare other linear and non-linear algorithms (e.g. MDS, SNE, t-SNE, etc).

In doing so, we will run a similar analysis as above where we vary the number of cells and the number of proteins sampled to determine an optimum algorithm and parameter set.

Finally, we will use the data to try and map out the life cycle of a cell. Although our data is not temporal and cannot be sampled in-situ, we plan to use a community of cells and sample a different cell within the same community at different stages of their life to determine if we can create a volume in 3D space that subtends the life cycle of a healthy cell. With this we can determine whether a cancer cell is anomalous regardless of which part of the lifecycle it may be currently in.

Acknowledgments

We would like to thank Karen Sachs and Andrew Gentles for providing the data and guidance for the project.

References

[1] Nam, K., Je, H. & Choi, S. Fast stochastic neighbor embedding: a trust-region algorithm. in 1, (2004).

[2]Amir, E.-A. D. et al. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. Nat Biotechnol 31, 545552 (2013).

[3]Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. Journal of Machine Learning Research 9, 85 (2008).

[4]Bendall, S. C. et al. Single-Cell Mass Cytometry of Differential Immune and Drug Responses Across a Human Hematopoietic Continuum. Science 332, 687696 (2011).