Finding Structure in CyTOF Data

Or, how to visualize low dimensional embedded manifolds.

Panagiotis Achlioptas
panos@cs.stanford.edu *

General Terms

Algorithms, Experimentation, Measurement

Keywords

Manifold Learning, Dimensionality Reduction, Flow Cytometry, CyTOF

1. INTRODUCTION

High-dimensional data are notorious for the difficulties that pose to most of the known statistical procedures that aim to manipulate the data in a meaningful and enlightening way. The so called phenomenon of the 'curse of dimensionality' has a severe negative impact to our current methods for various reasons; the size of the considered space grows exponentially as the number of dimensions increases, meaning that for instance solving a density estimation problem in such a space requires exponentially many samples (if no further assumptions are made). Single cell mass cytometry (CyTOF) is a recently introduced technique [1], that is anticipated to revolutionize the field of hematology and enhance our general understanding about the cells of living organisms. A main obstacle towards these goals is going to be the analysis of the produced, high dimensional, data.
How are we going to deal with data that 'live' in an ambient space of e.g. 100 dimensions? Definitely, without any further assumptions, our task will be difficult. Hopefully, as with many other 'real-life produced' datasets we expect that CyTOF data will be governed by only a few degrees of freedom. Such a scenario is possible when for example the data lie on a (non) linear manifold of low dimensionality, embedded in a higher dimensional space. This kind of data is said to have a 'small' intrinsic dimensionality and various techniques have been produced in order to discover the essential 'forces' that shape the data (leading to various dimensionality reduction schemes).

In this article, we will make the first move by showing that data produced by CyTOF are indeed intrinsically low dimensional and we will apply to them various state of the art techniques in order to get a concrete *visual* representation of them. The remainder of this article is organized as follows. In Section 2 we begin by giving a brief description of the kind of data produced by CyTOF along with some notational conventions. Next, in Section 3 we thoroughly introduce three popular techniques for estimating the intrinsic dimension of a data set sampled from a manifold. In sections 5, 6, 7 we describe the algorithms and the strategies we used for discovering the non linear, low dimensional embedding of CyTOF data. Instead of describing the experimental component of this article in a separate section, we explain our findings along with *each* technique used. We make our final conclusions in Section 8.

2. CYTOF DATA

The CyTOF data at hand, are composed by measurements of 31 human's *cell characteristics*, varying from the diameter of a cell, to its various proteins abundances. Though 31 dimensions used for describing the state of each cell are 'few' compared with the number of dimensions produced in other instances of the problem (e.g a 64×64 pixel photograph lives in the 4094 dimensional space), still, having for example a visual representation in the 31 dimensional space, is far from easy. Also, we anticipate that in the near feature CyTOF will be able to measure up to a hundred features of a cell, giving further importance to this first exploratory analysis. On top of this, instead of having a single (static) 'snapshot' for the characteristics of the analyzed cells, CyTOF generated different 'snap-shots' over 8 different time periods. The data is organized in $X_t(n \times D)$ matrices, where n is the number of analyzed cells (up to many thousands), $D = 31$ is the ambient (high) dimension and $t \in \{1, \ldots, 8\}$ varies over the different time periods. Finally, we mention here, that for many evaluation benchmarks the data behaved very similarly over the different time points; in these cases, for reasons of space saving, we present the experimental findings only for a single point in time $t = c$. For the rest of this article, $x_i \in \mathbb{R}^D$ will represent the i-th analyzed cell and $y_i \in \mathbb{R}^d$ it's counterpart in the low d-dimensional embedding.

*Computer Science Department, Stanford University.

3. ESTIMATING INTRINSIC DIMENSION

Thresholding Principal Components

Given a set $S_n = \{X_1, \ldots, X_n\}, X_i \in X, i = 1, \ldots, n$ of data points sampled independently from a Manifold M, probably the most obvious way to estimate the intrinsic dimension, D_{intr}, is by looking at the eigenstructure of the covariance matrix C of S_n. In this approach, $\hat{D}_{pca}$ is defined as the number of eigenvalues of C that are larger than a given threshold. This technique has two basic disadvantages. First it requires a threshold parameter that determines which eigenvalues are to discard. In addition, if the manifold is highly nonlinear, $\hat{D}_{pca}$ will characterize the global (intrinsic) dimension of the data rather than the local dimension of the manifold. $\hat{D}_{pca}$ will always *overestimate* D_{intr}; the difference depends on the level of nonlinearity of the manifold.

Luckily, our data suggest a very clear threshold. As seen in Figure 1 the three largest eigenvalues capture more than 70% of the total variance, while the 'elbow' between the third and fifth eigenvalue indicates the sharp decrease of the captured variance. Thus, we can be confident that D_{intr} with high probability will be less than 5 and it seems that even a 'meaningful' 2-dimensional embedding of the data exists. In order to enhance our reasoning for the existence of a low-dimensional embedding we estimate D_{intr} with two more statistics.

Correlation dimension

The second approach to intrinsic dimension estimation that we used is based on geometric properties of the data and requires neither any explicit assumption on the underlying data model, nor input parameters to set. It is based on the *correlation dimension*, from the family of fractal dimensions and the intuition behind its definition is based on the observation that in a D-dimensional set, the number of pairs points closer to each other than r is proportional to r^D.

DEFINITION 1. *Given a finite set $S_n = \{x_1, \ldots, x_n\}$ of a metric space X, let*

$$C_n(r) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} I(||x_i - x_j|| < r)$$

where I is the indicator function. For a countable set $S = \{X_1, X_2, \ldots\} \subset X$, the correlation integral is defined as $C(r) = \lim_{n\to\infty} C_n(r)$. If the limit exists, the **correlation dimension** *of S is defined as*

$$D_{corr} = \lim_{r\to 0} \frac{\log C(r)}{\log r}.$$

Since, for a finite sample the zero limit cannot be achieved, we instead used the **scale-dependent correlation dimension** of a finite set. This is defined as

DEFINITION 2.

$$\hat{D}_{corr}(r_1, r_2) = \frac{\log C(r_2) - \log C(r_1)}{\log r_2 - \log r_1}.$$

In our experiments instead of measuring $\hat{D}_{corr}$ for some constant, small values, r_1, r_2, we defined them to be the median and the maximum distance (respectively), among the distances defined by the k nearest neighbors of *each* datapoint. Thus, for each datapoint we computed a separate $\hat{D}_{corr}$ and our final estimate was their average value. We noticed our results to be very robust under different values of k.

Unfortunately, it is known that $D_{corr} \leq D_{intr}$ and that D_{corr} approximates well the D_{intr} if the data distribution on the manifold is nearly uniform (our approach of using different r_i's is not enough to circumvent this). We deal with the known non-uniformity of the CyTOF [1] by using our last estimator, this of *packing numbers*[4]. We avoid giving a detailed analysis of this recently introduced technique. Intuitively it tries to efficiently estimate the number of minimum open balls whose union covers the space S. This technique was shown to *not* be dependent on the data distribution of the manifold.

Intrinsic dimension of CyTOF data

As seen in Figure 2 CyTOF data (X_t), for all the different time points seem to lie on a *low* dimensional manifold. The principal components and the correlation dimension bound D_{intr} between 5 and ~ 1.5. (For estimating $\hat{D}_{corr}$ we used the distances of the $k = 10$ nearest neighbors of each datapoint.) Supported by these results, we turn to our next goal which is to find a meaningful *visual representation* of the data.

4. THE GLOBALLY LINEAR CASE

If we knew that the our data lived on, or very close to, a linear subspace, a d-dimensional hyperplane embedded in the ambient space, then PCA would be the tool to use. It would 'perfectly' discover the embedded hyperplane, since this would be the hyperplane spanned by the d eigevectors of the empirical covariance matrix that would have non zero (or very close to zero) eigenvalues.

But what if the data live on a more complex 'micro-world', like a non-linear manifold? We will describe and use 3 different techniques to deal with such a scenario: LLE, Isomap and T-Sne.

5. LOCALLY LINEAR EMBEDDING

Locally Linear Embedding [2], or LLE, is based on the idea that over a small 'patch' of a smooth manifold, its surface is approximately flat. Thus, it proposes to use the expected *local* linearity of the manifold in order to find a linear weight-based representation of each point from its neighbors, characterizing in this way the local relative positioning of each neighborhood in the high dimensional space. By using this local parameterization one can look for a new set of points in a lower dimension which preserves, as closely as possible, the same relative positioning information.

The first step of LLE is to solve for the weights that best characterize the points' relationship in $\mathbb{R}^D$:

[1] Apart from the experimental results which were supporting it, we had *prior* knowledge for the existence of small, distinct cell sub-populations.

$$\tilde{W} = \arg\min_{W} \sum_{i=1}^{n} ||x_i - \sum_{j \in N(i)} W_{ij} x_j||^2 \quad (1)$$

$$\text{s.t.} \qquad \forall i \sum_{j} W_{ij} = 1 \quad (2)$$

where $N(i)$ is the neighboring points of x_i (the neighborhood of each point can be calculated with various means, e.g. based on the k-nearest neighbors or by some local sphere of radius ϵ around x_i). The size of the neighborhood is a compromise; it must be large enough to allow for a good reconstruction of the points (it must contain at least d_{intr} points), but small enough for not violating the locally linear assumption. With the normalization requirement added the produced weight based representation will be invariant to any local rotation, scaling, or translation of x_i and its neighbors.

The second step is to find the embedding $\tilde{Y}$ which best preserves the previously found local configurations:

$$\tilde{Y} = \arg\min_{Y} \sum_{i=1}^{n} ||y_i - \sum_{j \in N(i)} \tilde{W}_{ij} y_j||^2 \quad (3)$$

$$\text{s.t.} \quad Y\vec{1} = 0, \quad YY^T = I_n \quad (4)$$

Under the above constrains, the problem is well-posed and a unique globally optimal solution can be found analytically, by reformulating it as a quadratic program.

5.1 LLE in practice

We tried LLE with a varying number k for the nearest neighbors of each data point, measuring their distances under the *Euclidean* norm (a meta-parameter of LLE). We found the algorithm to perform extremely poorly when we used as its input the entire analyzed cell population of any given time period (see Figure 4a). The reason behind this failure is explained graphically in Figure 3; here we we have plotted the percentage of cells captured by the two largest connected components of the (**disconnected**) neighborhood graph, under different values of k. In other words, the raw high dimensional data even for $k > 31$ (the ambient dimension) do **not** form a strongly connected neighborhood graph.

In 4b, we used a value of $k = 16$ and the algorithm considered $\sim 98\%$ of the cell population; the cells that form the largest *connected* component. The rest 2% of the cells were completely ignored, treated as very noisy observations. A justification for this stems from the fact that this 2% was scattered through roughly 200 different unconnected components, resulting in a lots of very small (2-3 cells each) groups of unrelated data.

6. ISOMAP

Isometric feature mapping, or Isomap [5] may be viewed as an extension of Multidimensional Scaling (MDS), a classical method for embedding dissimilarity information into Euclidean space. Isomap consists of two main steps:

1. Estimate the geodesic distances (shortest *curved* distances) among the data points in $\mathbb{R}^D$.
2. Use MDS to embed the data points in a low-dimensional space that best preserves the estimated geodesic distances.

For the first part of Isomap, we appeal again to the local linearity of the manifold. If our data are sufficiently dense, then there is some local neighborhood in which the geodesic distance is well-approximated by the naive Euclidean distance. Taking this local distances as trusted, farther distances may be approximated by finding the length of the shortest path along trusted edges.

6.1 Isomap in practice

For the approximation of the geodesic distance to be accurate (among distant points) the prerequisite of dense data is essential. A simple way to achieve this would be to use the neighborhood graph resulted by some large value of k. On the other hand, such an attempt with high probability would suffer by what is know in the literature as *'short circuit distances'*, which can lead to drastically different (and incorrect) low-dimensional embeddings. $k = 19$ was found to be the golden ratio in our empirical results as it managed to give a good balance in the trade-off between the fraction of the variance in geodesic distance estimates which are *not* accounted for in the Euclidean (low dimensional) embedding; and the fraction of points not included in the largest connected component (see Figure 5).

7. T-SNE

T-distributed stochastic neighbor embedding [3], works under a different vain. It was developed to address the problem of *visualizing* the high dimensional data, aka it works for embeddings with $d \leq 3$ and it's not appropriate for arbitrary dimensionality reduction. Its basic idea is to describe the similarities between data points (e.g. their Euclidean distances) as conditional probabilities and then try to minimize the (Kullback-Leibler) divergence of these probabilities as they were captured in the low and the high dimensional space respectively. In a sketchy description, T-SNE works as follows:

1. $\forall\ x_i$ in the high dimensional space let $p_{j|i}$ to be proportional with the density of $||x_i - x_j||$ drawn from a $Gaussian(x_i, \sigma)$.
2. $\forall\ y_i$ in the low dimensional space let $q_{j|i}$ to be proportional with the density of $||y_i - y_j||$ drawn from a Student's T-distribution with 1 degree of freedom.
3. Use a gradient descent like method, to minimize the the $KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$

The success of this method, as also revealed in the CyTOF data, is that it effectively copes with the *crowding problem*[3]. This problem in simple words stems from the fact that the area in two (or three)dimensions that is available to accommodate moderately distant points (in the high dimensional space) will not be nearly large enough compared with

the area available to accommodate nearby points. Thus, if we want to model the small distances accurately in the low dimensional space, most of the points that are at moderate distance from point i will have to be placed much too far away from it. The way T-SNE achieves this in an effective manner comes from the use of the *heavy tale* T-Student distribution for modeling the distances in the *low dimensional* space. Comparing it with the probabilistic setting of SNE or LLE, where every pairwise distance can be regarded as having being generated under a Gaussian distribution (with mean equal to one of the two points), the T distribution makes it much more likely for 2 points to be embedded **far away** from each other. Thus as seen, also in our experiments (see Figures 9, 10) T-SNE produces less compact and thus 'cleaner' results. For all our experiments with T-SNE we used gradient descend for 1000 iterations and with 10 different re-initializations. In most cases we manage to converge to a local minima pretty quickly.

8. CONCLUSIONS

The most important result which stemmed out from this project is that we now have more experimental evidence that CyTOF data are likely to be governed by very few degrees of freedom. Also, regarding the methods used, we can now make some statements about their performance in this kind of data. By visual inspection,T-SNE outperformed all the other methods. In order to boost this evidence, we have measured the following type of error that each method introduced in its low dimensional embedding; For every x_i, let $p_i(m)$ be the value the i-th cell has in its m-th dimension in the high dimensional space. Figure [12] plots the sum of the squared distances between every embedded cell and its 5 nearest (embedded) neighbors, as these are formed under the dimension of the *original space* $p_i(m)$ with the maximum variance.

(The two dimensions that with maximum variance in the original space, were those measuring the CD3 and CD45 protein abundances of the cells. These 2 protein abundances were used to give color to all of our plots, by coloring every cell with an intensity proportional to its original CD3 or CD45 abundance).

In Figure 8 we see the 2-dimensional embedding resulted by PCA. Interestingly enough, it manages to separate the data (colors) to a large extent. Some results from the use of Isomap (who found to perform slightly better than LLE) are shown in [6], [7]. Finally, in Figure [11] one can see the kind of the 2-dimensional embeddings that a k-means like algorithm could produce for this data. Clearly, the manifolds learners used outperform such an approach.

9. ACKNOWLEDGEMENTS

The author would like to thank Prof. Daphne Koller who served as his rotation advisor in Stanford and gave him the opportunity to 'play around' with the CyTOF data. Also, Manfred Claasen, who was an invaluable source of inspiration and help. The materials written by Lawrence Cayton and Alexander Ihler, available online, helped him a lot in demystifying non linear manifold learners. Finally, the publicly available code of Laurens van der Maaten was a great help.

10. REFERENCES

[1] Sean C. Bendall et. al, Single-Cell Mass Cytometry of Differential Immune and Drug Responses Across a Human Hematopoietic Continuum. Science, 2011.

[2] Sam Roweis, Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290 (5500), 2323-2326, 22 December 2000.

[3] Laurens van der Maaten, Geoffrey Hinton Visualizing Data using t-SNE. Journal of Machine Learning Research, 2008.

[4] Balazs Kegl Intrinsic Dimension Estimation Using Packing Numbers NIPS, 2002.

[5] J. B. Tenenbaum, V. de Silva and J. C. Langford A Global Geometric Framework for Nonlinear Dimensionality Reduction Science, 290 (5500), 2319-2323, 22 December 2000.

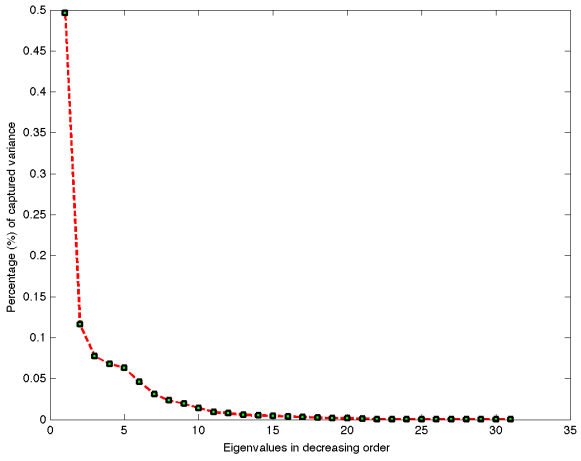


Figure 1: Percentage of variance as captured by an increasing number of (decreasingly sorted) eigenvalues, ($t = 5$).

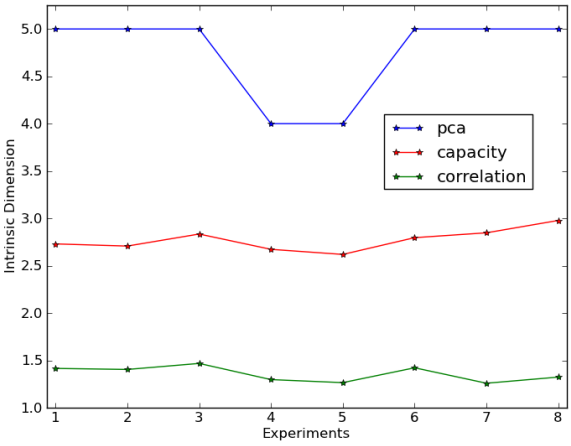


Figure 2: Three estimators of the Intrinsic Dimension for all the 8 different time periods.

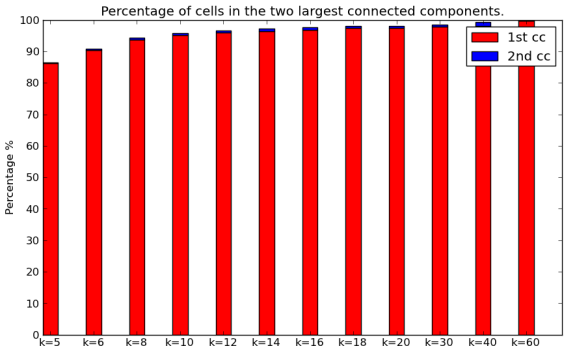
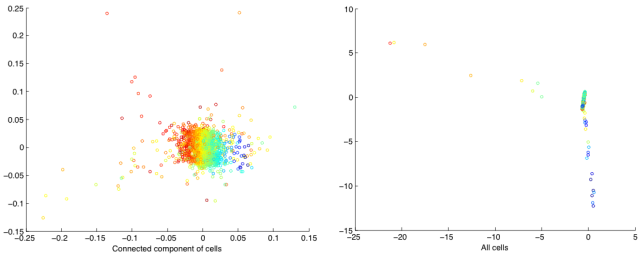


Figure 3: Percentage of cells in largest and second largest connected components of the neighborhood graph ($t = 5$).



(a) LLE Only CC (b) LLE All Cells

Figure 4: LLE 2D embedding ($k = 16$). In 4a only the cells in the largest connected component were considered by LLE. In contrast at 4b , LLE was applied to the whole cell population. The difference in the embedding is pretty dramatic. Both experiments are at ($t = 5$) and for coloring use CD45 cell abundances.

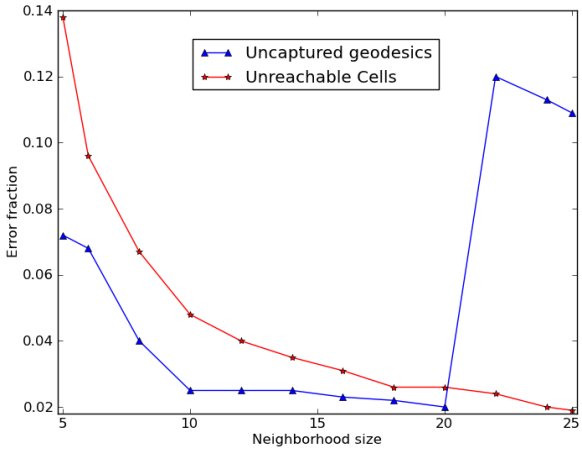


Figure 5: Trade-off between the fraction of the variance in geodesic distance estimates not accounted for in the Euclidean (2-d) embedding and the fraction of points not included in the largest connected component.

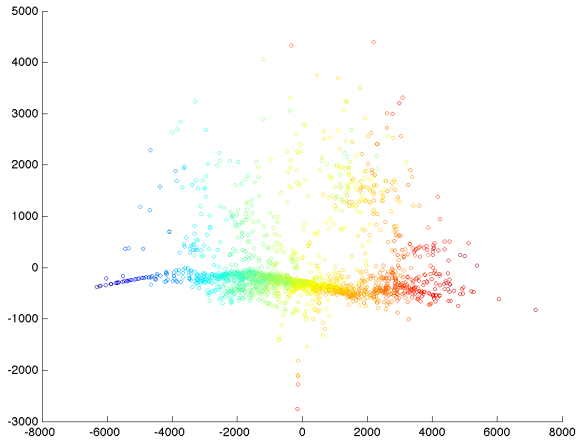


Figure 6: Isomap 2D embedding ($t = 1$). CD3 cell abundances used for coloring.

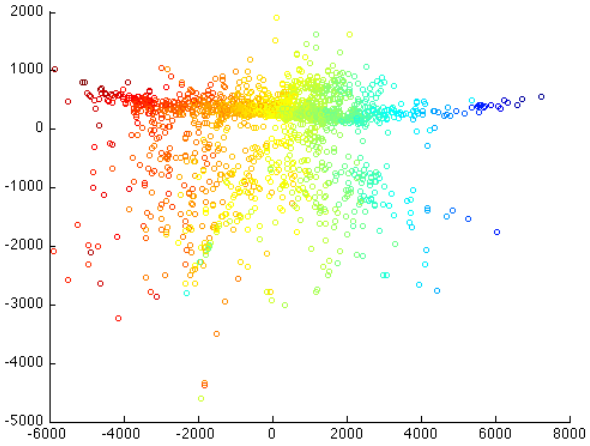
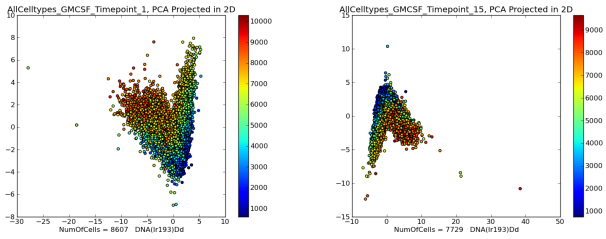


Figure 7: Isomap 2D embedding ($t = 5$). CD3 cell abundances used for coloring.



(a) $t = 1$ (b) $t = 8$

Figure 8: A 2D embedding constructed by projecting the cells on the 2 larger eigenvectors. Colors resulted by CD3 cells abundances.

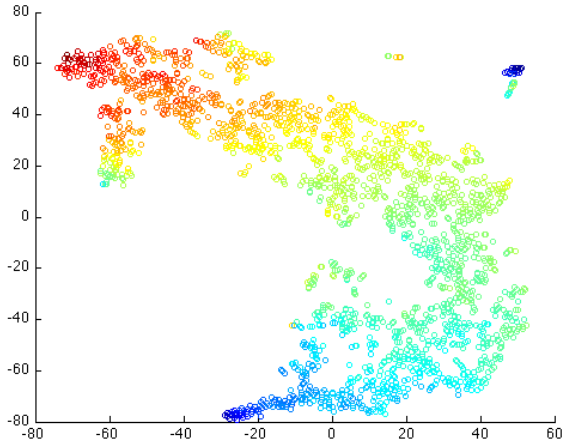
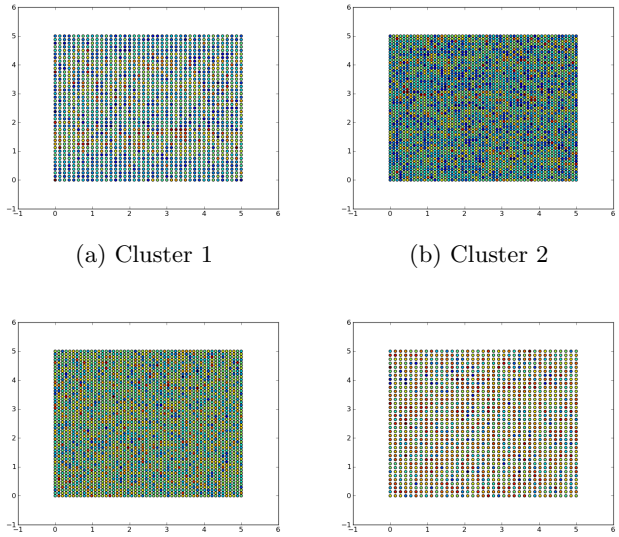


Figure 9: T-SNE 2D embedding ($t = 1$). CD3 cell abundances used for coloring.



(a) Cluster 1 (b) Cluster 2

(c) Cluster 3 (d) Cluster 4

Figure 11: A 2D embedding as considered by the K-medians algorthim, with $k = 4$. For each cluster we have plotted the colors resulted by CD45 cells abundances. The relative inner-cluster (grid) positions of the cells are random.

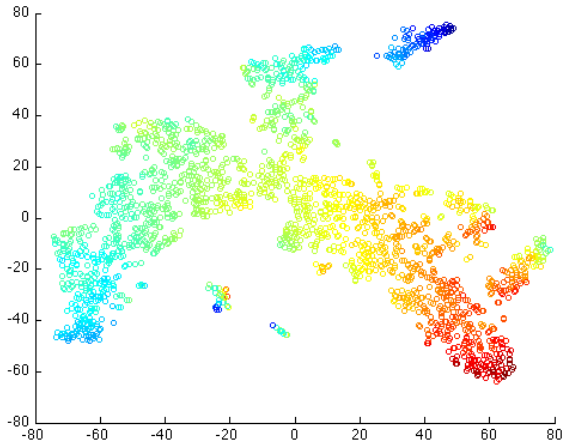


Figure 10: T-SNE 2D embedding ($t = 5$). CD3 cell abundances used for coloring.

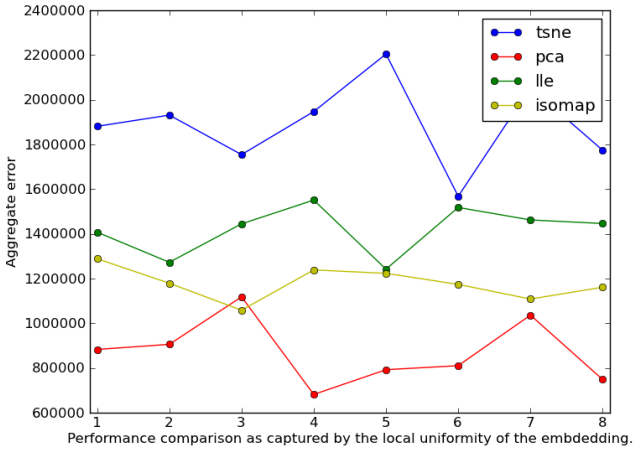


Figure 12: Comparison of methods as captured by the average difference between each embedded data point and its 5-closest neighbors under their L_1 distance in feature CD3.