Dimensionality reduction in Single cell sequencing using Autoencoder

Ashwani Kumar

University of Texas at Dallas, TX (AXK200017)

University of Texas at Southwestern Medical center, Dallas, TX

***Abstract*— Single cell sequencing technology refers to the sequencing of genome or transcriptome at the single cell resolution, so as to obtain the cell population difference. Single-cell technologies have the advantage of detecting heterogeneity among individual cells compared to other traditional sequencing methods where we get the average of overall cells and less to no resolution of the cellular heterogeneity. Although this technology has emerged as one of the most sought method for biologist, the bioinformatics of single cell data presents a challenge. This is due to high dimensional nature of data and dropout events during the sequencing.**

Keywords—Single cell sequencing, auto encoder, PCA, TSNE, UMAP

I. Introduction

Single cell RNA sequencing technology enables the capture of expression of genes of the individual cells thus giving better resolution of cellular heterogeneity. Traditional RNA sequencing methods gives an average of gene expression in many cells but loose the cellular heterogeneity [1]. An important steps in single cell sequencing analysis is the clustering of cells into groups (known and novel cell type) on the basis of expression levels of signature genes. However, clustering step has been limited by dimensionality of the single cell data. This is because clustering methods in high dimensional data could lead to misleading results as the distance between most of the pair of genes is similar. Other properties of single cell data such as sparsity, dropout, and batch effect adds further complexity. Therefore, finding the accurate low dimensional representation of the data becomes more crucial than downstream analysis [2]

Earlier works have applied various dimension reduction methods to the single cell dataset. PCA is used for an initial dimension reduction on the basis of highly variable genes. Another method called t-SNE (t-distributed stochastic neighbor embedding) is a nonlinear dimension reduction technique that preserves the local structure in the data. The points which are closer to one another in the high dimension data set will tend to be close to one another in the low dimension. The t-SNE [3] algorithm works by modeling the probability distribution of the neighbors around each point. In the high dimensional space this is modeled as Gaussian distribution, whereas in low dimensional space probability distribution are modeled as t-distribution. The algorithms works by finding a mapping onto 2-dimension space that minimizes the difference between two distributions between all points. However t-SNE suffers from limitation such as loss of information on inter cluster relationship, slow computation time. Another method which has recently been widely used in the single cell sequencing is UMAP [4] (uniform Manifold Approximation and Projection). UMPA offers advantages over t-SNE such as increase in speed and better preservation of the global structure.

The above mentioned dimensionality reduction approach achieves good performance. However, robust approaches are needed to account and adjust for nature of single cell sequencing dataset. In this project I propose one such approach; the autoencoders.

II. Autoencoders for Dimensionality Reduction

An autoencoder is an unsupervised learning technique that uses neural network to reconstruct its input to its output. We want to use the existing structure (example correlation between genes in single cell) in learning process, consequently generating a compressed knowledge representation of input data. Architecturally autoencoders consists of encoder layer, a bottle neck layer and decoder layer (Fig.1)

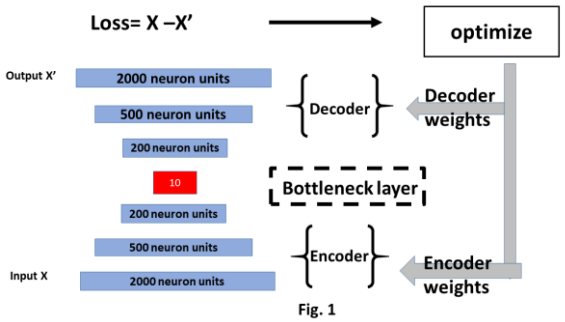


Fig. 1

Fig.1: Architecture of autoencoder

Encoder layer convert the inputs to latent representation and the decoder layer converts the latent representation to the outputs. The autoencoders are capable of learning dense representation of input data know as latent representations or coding in an unsupervised manner. These representation have lower dimension than the input layer, therefore making the autoencoder useful for dimensionality reduction and feature extraction.

There are various types of autoencoders. Under complete autoencoder uses smaller coding/representation layer dimension. This enables it to learn the most salient features of the training data. In this type of encode the model is trained according to the reconstruction loss. Therefore only way of ensuring that model is not memorizing the input data is to reduce the number of dimension in representation/bottleneck layer. Examples of other types of encoder are sparse, denoising,contractive, variantional, GAN. In this project I have focused on simple architecture of undercomplete autoencoders.

III. METHODS AND ALGORITHM

Algorithm: Dimesnion reduction using autoencoder

Data : X, an N x M expression matrix (N = number of cells/samples, M = Number of genes/Features)

1. Log transform (X)
2. X(pca_dim) = PCA(X) // Apply PCA on X; Get the principal components
3. Train autoencdor on X_TRAIN (pca_dim)
4. for each epoch
   FORWARD PASS;
   RECONSTRUCTION ERROR;
   BACKPROPOGATION;
   UPDATE WEIGHTS USING GRADINET DESCENT
5. Test encoder on X_TEST
   FORWARD PASS UPTO BOTTLENECK LAYER

6 END

DATASET

I used the human peripheral blood mononuclear cells (PBMCs). This dataset is publically available and can be download from the 10X genomics website [9]. This dataset has 2700 samples (cells).

The image files of sequencing data first should be converted to gene expression count matrix. For this I followed the Seurat guideline (https://satijalab.org/seurat/articles/pbmc3k_tutorial.html)

STEPS

1) The single cell sequenicng gene expression matrix was used as the input file for the autoencoder model. The matrix is of form N * M, where N is the number of cells or samples ; M is the number of gene (features) (Fig 2).

2) The input matrix is first projected into the PCA space to get the PCA residulas. These residuals reatain the ndim principal components of form n*p. Here n is the number of cells and p is the reduced principal components.

3. The selected number of principal components is then visulaized for distinct separtion of cells into cluster by tSNE.
4. The same selected number of principal components is then used as an input for the model training and testing.
5. Our aim here is the features dimension reduction to facilitate better distincition of cells into different clusters. Training constsist of making an autoencoder layer consisting of encoder and decder. The forward pass is made through each layer. The loss is calculated at the output resonstruced layer; then back propogated and weights updated by gradinet descent in each epoch
6. For testing, I used the split dataset for predicting the reduced features in the bottle neck layer. The output from the bottle neck layer wasn then visualized by tSNE.

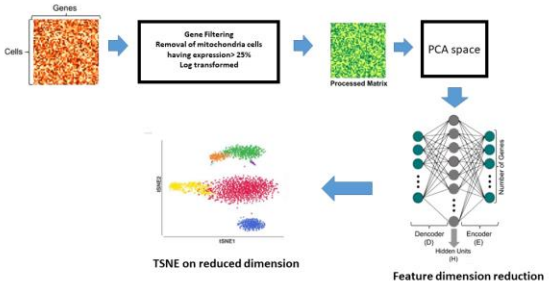


Fig 2. Steps applied in this project for preprocessing and dimensiion reduction

IV. RESULTS

The model was trained and best performance was recorded. (Table 1; Supplementary data).

Here I describe the combination of parameters which performed better. I used sigmoid activation function for the autoencoder model. The most important hyper parameters which were varied was number of layers and learning rate.

Although, principal component plots showed separation of cell but the separation was not distinct as shown in figure 3. The first 2 dimensions which explains the most variations has been shown. However when the reduced dimensions of PCA was visualized via tSNE, I see more distinct separation of cells.

This is shown in the figure 4. I am using TSNE as the visualization technique.

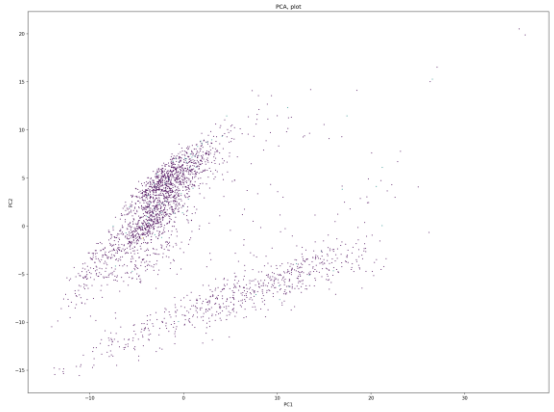


Fig 3: PCA plot. Two dimensions are shown.

Each dot represents a cell/sample

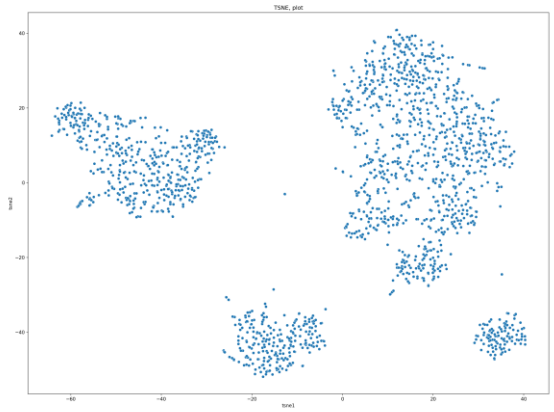


Fig 4: TSNE plot of PCA reduced dimensions

Each dot represents a cell/sample

When the model was trained using sigmoid activation function, learning rate- 0.1, number of layers - 5 and epoch size as 500, the model performed better with a decreasing loss. The loss curve is presented in Fig 5.

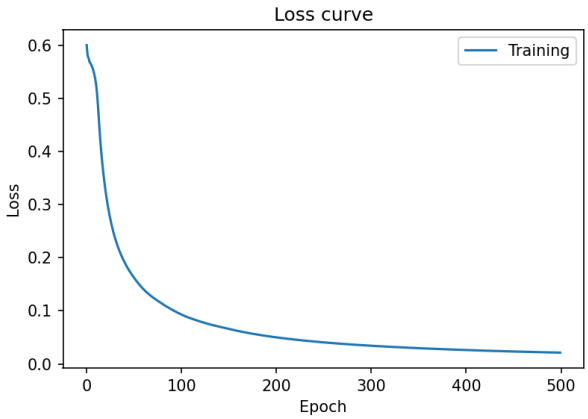


Fig 5: Mean square error loss curve

The reconstruction is diagnosed by the loss curve. To see how the model has accounted for reduced dimension, I plotted the dimension of bottle neck layer as scatter plot. This is shown in Fig 6.

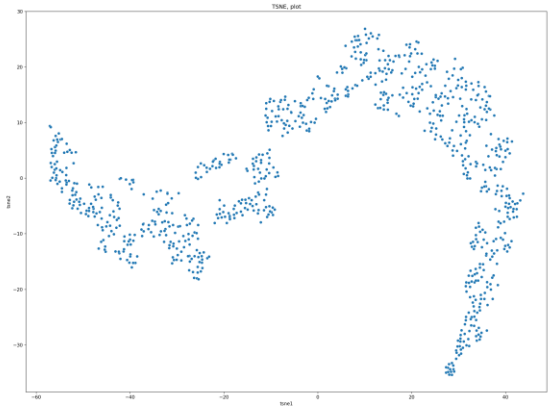


Fig 6: TSNE plot on autoencoder model (Table1; Experiment 1)

Experiment	Parameters	Results
1	Neural net: Number of layers :5 Neurons= (10,5,2,5,10) Encoder neurons = (10,5) Decoder neurons = (5,10) Bottleneck layer =2 Error func = MSE Learning rate 0.1	Train/Test split- 60:40 Loss curve on train : **Fig 5** Reduced dim on test data: **Fig 6**

2.	Neural Net: Number of layers :5 Neurons= (10,5,2,5,10) Encoder neurons = (10,5) Decoder neurons = (5,10) Bottleneck layer =2 Error func = MSE Learning rate 0.01	Train/Test split-60:40 Loss curve : **Supp. Fig 1** TSNE on test data : **Supp. Fig 1**
3	Neural Net: Number of layers :7 Neurons= (15,10,5,2,5,10,15) Encoder neurons = (15,10,5) Decoder neurons = (5,10,15) Bottleneck layer =2 Error func = MSE Learning rate 0.1	Train/Test split-60:40 Loss curve : **Supp. Fig 2** TSNE on test data : **Supp. Fig 2**
4	Neural Net: Number of layers :7 Neurons= (15,10,5,2,5,10,15) Encoder neurons = (15,10,5) Decoder neurons = (5,10,15) Bottleneck layer =2 Error func = MSE Learning rate 0.01	Train/Test split-60:40 Loss curve : **Supp. Fig 3** TSNE on test data : **Supp. Fig 3**

Table 1: Logs of experiment

V. Discussion

Selection of activation function: For this project I have selected the sigmod activation function. The selection of activation function is important because of the vanishing and exploding gradient problem. Alongside, the selection of weight initialization scheme is also important. I have randomly initialized the low weights to neuron connections. An important future direction of this project would be to train the model with different weight initialization scheme and activation functions.

Number of layers: This is one of the most important hyperparameter. The experiment was done taking the number of layer as 5, 7 and 9. The first and last layer are the input and output layer respectively. The bottle neck layer is the coding layer with the most reduced dimension. In all the experiments number of bottle neck layer was kept as 2. The number of layers in encoder and decoder was also varied. Please see Table 1 for the experiment details. It is advised that if the encoder and decoder are allowed too much capacity, the autoencoder model simply copies the input, without performing the feature extraction.

Selection of learning rate: I used learning rate of 0.1 and 0.01 for each of the experiment.

Overall the learning rate with 0.1 and number of layer as 5 (neuron units = 10, 5, 2, 5, 10) performed better than the other experimental values. Please see the Table 1 and supplementary figures for all the results.

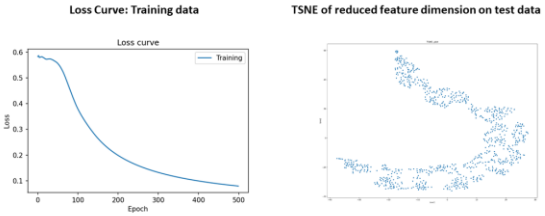
The performance also depends on the type of datasets. For example, the single cell datasets are usually prone to dropout (non-detection of genes because of experimental protocols in lab), missing of gene expression values etc. Therefore an important future direction would be fine tune parameters in more detail. Another interesting experiment would be to use different activation function for layers to see the reconstruction output.

In autoencoder models the performance is measure by the reconstruction loss and visualization of the reduced features. Figure 5 shows the TSNE visualization of test dataset predicted by autoencoder. We would expect the model to separate cells (each dot represents a cell/sample) into distinct clusters, similar to that of Figure 4. Although we see separation, we must note that training and testing data was split into 60:40. For gene expression datasets, performing dimension reduction on split test data could results in increase of error.

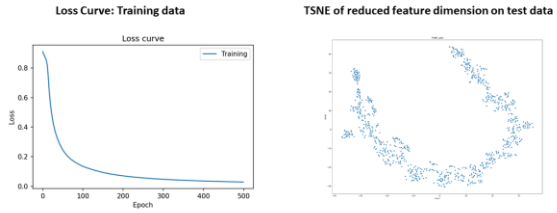
VI. Conclusions

The aim of this project was to study the dimension reduction of single cell dataset using autoencoder model. For dimension reduction there are many methods which works well with the dataset, but they have their own limitations of scalability and data dependency. Autoencoders are an alternative to both feature extraction and reduction of the dimension. This projects highlights the applicability of autoencoder model in single cell datasets, thus presenting with more application scope in other genomics fields such as discovery of candidate gene markers for diseases, co-expression analysis between genes in cluster etc.

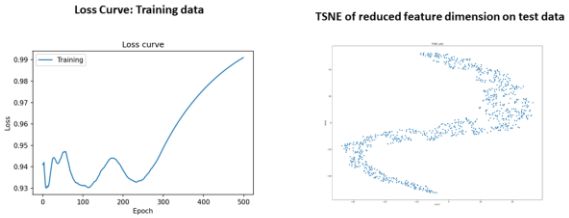
Supplementary Data



Supp. Fig 1



Supp. Fig 2



Supp. Fig 3

REFERENCES

[1] Tang, X., Huang, Y., Lei, J. et al. The single-cell sequencing: new developments and medical applications. Cell Biosci 9, 53 (2019). https://doi.org/10.1186/s13578-019-0314-y

[2] Tangherloni, A., Ricciuti, F., Besozzi, D. et al. Analysis of single-cell RNA sequencing data based on autoencoders. BMC Bioinformatics 22, 309 (2021). https://doi.org/10.1186/s12859-021-04150-3

[3] Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. J Mach Learn Res. 9(Nov), 2579–2605 (2008)

[4] Becht, E., McInnes, L., Healy, J. et al. Dimensionality reduction for visualizing single-cell data using UMAP. Nat Biotechnol 37, 38–44 (2019). https://doi.org/10.1038/nbt.4314

[5] Wang D, Gu J. VASC: dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. Genom Proteom Bioinf. 2018;16(5):320–31.

[6] Lin E, Mukherjee S, Kannan S. A deep adversarial variational autoencoder model for dimensionality reduction in single-cell RNA sequencing analysis. BMC Bioinformatics. 2020;21(1):1–11.

[7] Geddes TA, Kim T, Nan L, Burchfield JG, Yang JY, Tao D, Yang P. Autoencoder-based cluster ensembles for single-cell RNA-seq data analysis. BMC Bioinformatics. 2019;20(19):660.

[8] **https://satijalab.org/seurat/index.html**

[9] **https://www.10xgenomics.com/resources/datasets?query=&page=1&configure%5Bfacets%5D%5B0%5D=chemistryVersionAndThroughput&configure%5Bfacets%5D%5B1%5D=pipeline.version&configure%5BhitsPerPage%5D=500**