Carter Allen

December 4, 2020

BMI 8050.01

Final Project Report

Deep Learning Methods for Dimension Reduction and Clustering in Single-Cell Spatial Transcriptomics Data

**Abstract**

In single-cell RNA sequencing (scRNA-seq) data analysis, a fundamental goal is to identify clusters of cells that exhibit similar gene expression patterns. These cell clusters may then be mapped to know reference data to annotate each cell cluster with a biological cell type (e.g., T-cells, natural killer cells, etc.). Recently, spatial transcriptomics technologies such as 10X Genomics Visium have made it possible to simultaneously measure the 2-dimensional spatial location and gene expression profiles of individual cells or small groups (i.e., 3 to 5) cells. This spatially resolved scRNA-seq data provides the opportunity to improve cell clustering, and as a consequence, may allow for more accurate cell type identification, since we expect fundamentally that cells of the same type are on average likely to be close to one another in a tissue sample in addition to featuring similar gene expression profiles. In this project, we investigate using a deep learning framework, namely autoencoders, for improved dimension reduction and feature extraction of scRNA-seq data. We then present an extension of standard modularity-based network clustering methods for integration of spatial information into the cell clustering analysis phase. We evaluate the cluster recovery accuracy on simulated spatial transcriptomics data with known ground truth parameters and compare the autoencoder dimension reduction framework to a collection of standard methods. We found that the use of autoencoders along with spatial information allows for highest accuracy of cell type identification.

**Introduction**

Single-cell RNA sequencing (scRNA-seq) data analysis often centers on identification of cell types within a tissue sample. These cell types help determine the ecological dynamics of a tumor microenvironment and have important implications for determining treatment effects, among other things. As such, accurate identification of cell types is crucial for the validity of a scRNA-seq data analysis.

Typically, the cell-by-gene expression matrix must be first fed through a feature extraction and dimension reduction workflow to reduce noise and allow for efficient construction of cell-cell similarity scores in a low dimensional space. For example, the popular software Seurat utilizes the linear dimension reduction algorithm Principal Components Analysis (PCA) to generate “meta-features” to be used in cell clustering (Stuart et al., 2019). These meta-features can be regarded as orthogonal summarizations of various sources of variability in the data. While classic dimension reduction algorithms such as PCA may be natural choices for snRNA-seq data analysis workflows due to their ubiquity in the computational community, the critical nature of dimension reduction in determining the quality of scRNA-seq analysis justifies further investigation of the best possible methods for generating features from a gene expression matrix.

Autoencoders are unsupervised neural networks suited to finding low-dimensional feature representations from unlabeled data. Generally speaking, an autoencoder neural network is comprised of three components, namely an encoder layer(s) to project the high dimensional space into a lower dimensional space, a bottleneck layer to learn the low dimensional features, and output layer(s) for predicting the original features. Autoencoders are non-recurrent, meaning there is no implicit temporal or other ordered structure to the input data. While autoencoders are a flexible deep learning framework, their flexibility requires careful tuning of model parameters. For example, the number and dimension of the encoder/decoder layers can improve performance but also lead to overfitting. We discuss these issues and more below in **Methodology**.

**Methodology**

1. *Dimension Reduction.* We implemented a variety of gene expression dimension reduction schemes to compare standard workflows with cutting edge frameworks. Specifically, we considered (i) a straightforward PCA dimension reduction to 50 principal components, (ii) a UMAP dimension reduction to 50 features, (iii) a combination of PCA reduction to 50 features followed by a further UMAP reduction to 2 features, (iv) a combination of PCA reduction to 50 features followed by a further reduction to 2 features using a 3-layer autoencoder framework, (v) a combination of PCA reduction to 50 features followed by a further reduction to 2 features using a 5-layer autoencoder framework, and (vi) a combination of PCA reduction to 50 features followed by a further reduction to 2 features using a 7-layer autoencoder framework. In each autoencoder implementation, the exponential linear unit (ELU) activation function was used for the encoder and decoder layers, while a linear activation function was used for the bottleneck layer. In future work, we will consider further combinations of these methods in addition to other commonly used dimension reduction techniques such as t-Distributed Stochastic Neighbor Embedding (t-SNE) and Hierarchical Stochastic Neighbor Embedding (HSNE).
2. *Nearest Neighbors Network Construction.* After the single-cell level gene expression data is fed through each of the considered dimension reduction approaches, we construct cell-cell K-nearest neighbor (KNN) networks to be used for community detection and cell type inference. We adopted the widely used heuristic to choose the number of neighbors (Duda et al., 2000), where is the number of cells in the data. Cell-cell similarity is defined as Pearson’s correlation between cell-level gene expression features given by the dimension reduction strategies outlined previously. We encode the cell-cell nearest neighbor network by an adjacency matrix , with elements equal to if cells and are neighbors, and otherwise.
3. *Network Clustering.* For each of the aforementioned dimension reduction techniques, we implemented two modularity-based community detection algorithms to cluster cells in the cell-cell KNN network. The first community detection algorithm that we considered uses the concept of network modularity to determine the optimal cluster labeling. For a given network encoded by adjacency matrix and a given cluster labeling for each node encoded by the vector with , the modularity is defined as

*,*

where is the total number of edges in the network, is a resolution parameter that controls the number of clusters detected, is the degree of node , and is an indicator function equal to if and otherwise. The modularity function only considers gene expression information in the clustering of cells.

In contrast, we define spatial modularity as

+ ,

where is the spatial distance between cells and , and , as visualized below, is a distance function chosen to reward clustering of spatially close cells in the same cluster. Since the goal is to maximize modularity, larger values of lead to higher modularity and thus better clustering fit.

A picture containing line chart

Description automatically generated

*Figure 1. Spatial distance function*

For both the classic modularity and spatial modularity approaches, we adopted a belief propagation approach to find the cluster labeling that maximized the modularity. In this approach, each node (cell) is first assigned to a unique cluster. Then, the cluster label of each node is changed to the cluster label of one of its neighboring nodes such that the modularity gain is maximized. This process is repeated until no further increase in modularity is possible. Figure 2 below depicts an animated visualization of this algorithm through the first three iterations.

Chart, scatter chart

Description automatically generated

*Figure 2. Animation of belief propagation modularity optimization algorithm.*

1. *Simulation Data*. To compare the performance of each dimension reduction and clustering method, we generated a simulated spatial transcriptomics data set with known ground truth parameters. The primary goals of our simulation study were to (i) generate gene expression counts that resemble features found in real-world single-cell data such as zero-inflation due to gene drop out and over-dispersion, and (ii) simulate cell-level spatial locations such that cells of the same cell type, on average, tend to be spatially closer than cells of two different cell types. For this project, we simulated cells of different cell types. Each cell was assigned gene expression counts for genes, of which genes were assigned to be markers for one of the cell types. The markers were distributed randomly and uniformly into each of the cell types, thus each cell type was defined by approximately marker genes. The values of gene expression counts were sampled from one of two zero-inflated negative binomial (ZINB) distributions. For the gene expression values of genes and cell combinations where the gene is considered a marker for the cell type of the cell, the expression counts were drawn from a “signal” ZINB distribution with a mean of 50 and dispersion parameter of 0.5 (dispersion parameters below 1 indicate over-dispersion). All other expression counts were sampled from a “noise” ZINB distribution with mean 25 and dispersion parameter 0.5. The zero-inflation factor was 0.02 in the signal distribution and 0.05 in the noise distribution. For the spatial locations of cells, we constructed a rectangular and evenly spaced mean grid for each of the cell types and sampled locations of each cell from an independent bivariate normal distribution with mean dictated by the mean grid and common variance of 1. Figure 3 below shows visualizations of the simulated data.

A picture containing text

Description automatically generated*Chart, scatter chart

Description automatically generated*

*Figure 3. (Left) Heatmap of gene expression counts. Rows correspond to cells and columns correspond to genes. The diagonal signal pattern in interspersed with zero counts due to gene dropout and surrounded by random noise. (Right) 2-dimension spatial plot of cell locations, where each cell is colored according to cell type.*

**Results**

Table 1 displays cluster recovery accuracy from each combination of dimension reduction and clustering approach. In each case, the inclusion of spatial information improved the clustering accuracy. This indicates that spatial information can be used to clarify ambiguities in gene expression data introduced by experimental and biological noise. With regards to the dimension reduction approaches, the combination of PCA and 5-layer autoencoder achieved the highest clustering accuracy. This suggests that, despite the low-dimension data projection produced by autoencoders, this framework was able to preserve the gene expression signal patterns needed to discern cell types.

|  |  |  |
| --- | --- | --- |
| Dimension Reduction | Non-spatial modularity | Spatial modularity |
| PCA (50) | 0.8150 | 0.8293 |
| UMAP (50) | 0.8974 | 0.9725 |
| PCA (50) + UMAP (2) | 0.8326 | 0.8671 |
| PCA (50) +  Autoencoder (3 layer) | 0.8991 | 0.9413 |
| PCA (50) + Autoencoder (5 layer) | 0.9401 | 0.9739 |
| PCA (50) +  Autoencoder (7 layer) | 0.9398 | 0.9684 |

*Table 1. Cluster recovery accuracy from each dimension reduction and clustering method. Integrating spatial information improved cluster recovery accuracy in all cases.*

For each autoencoder, we observed little to no evidence of overfitting, as the training and validation loss curves do not clearly diverge over the course of all model iterations (Figure A1). In Figure A2, we show the dimension reduction of all cells using the preferred 5-layer autoencoder, where each cell is colored by the ground truth cell type given in the simulated data. This figure shows an apparent separation between cell types, indicating an ability of the autoencoder to recognize the latent cell types given the gene expression PCA features.

**Discussion & Future Work**

We found two overarching conclusions from this project. First, we found that, using simulated spatial transcriptomics data, the inclusion of spatial information into cell clustering improves cluster recovery accuracy. In future work, we will investigate other more complex spatial structures to see if this framework is capable of detecting spatial signal in non-regular distributions. Additionally, we found that the use of autoencoders for dimension reduction improved cluster accuracy performance relative to PCA. However, further investigation is required. In future work, we hope to develop a robust framework for choosing autoencoder parameters. We will also make more comprehensive comparisons to existing methods.

**References**

Duda, R. O. *et al. Pattern Classification (2nd Edition)*. (Wiley-Interscience, 2000)

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck III, W. M., ... & Satija, R. (2019). Comprehensive integration of single-cell data. *Cell*, *177*(7), 1888-1902.

**Appendix**

Chart

Description automatically generatedChart

Description automatically generatedChart, line chart

Description automatically generated

*Figure A1. Autoencoder loss curves for training and validation data. We used a 75/25 training/test split to train each autoencoder. (Left) shows the 3-layer autoencoder (1-1-1), (middle) shows the 5-layer autoencoder (2-1-2), and right shows the 7-layer autoencoder (3-1-3). Loss curves for the training and validation sets do not show substantial signs of divergence, indicating that overfitting is likely not a major concern for these models.*

*Chart, scatter chart

Description automatically generated*

*Figure A2. Autoencoder dimension reduction of the cell by gene expression matrix with cells colored by true simulated cell types.*