Classification and Clustering of Single Cell RNA-seq Gene Expression Data Using Deep Learning Features

Chieh Lin
Andrew ID: chiehl1
Machine Learning Department
Carnegie Mellon University
chiehl1@cs.cmu.edu

1 Introduction

To understand the regulations of cell functions in complex biological systems, it is very important to identify various cell types in the systems. Take nervous systems as example, the cell type diversity is still poorly understood so that the most of the brain functions remain a mystery. Also, even the cells with the same cell type might play different roles in different phases. Therefore, classification of different phases of cells is also important. For now, several of the cell type classification works are based on principal component analysis (PCA) with manual curations. This project aims to apply deep learning methods to extract non-linear features from the single cell RNA-seq data and use the feature in other classification or clustering methods. The ultimate goal of this project is using deep learning to generate deep features to help classify/cluster unknown cell types in order to discover new cell types.

2 Materials and Methods

Previous work
From literature survey, papers have been found to use denoising autoencoders to extract deep features and analyze the parameters or do clusterings with the features. Our first step is also to select Autoencoder model because its simplicity, good performance in many papers and it can generate compact representations of high dimentional inputs. The concept of autoencoder is to reconstruct the original input with smaller hidden layer so that the hidden layer can be regarded as the compressed code for the input.

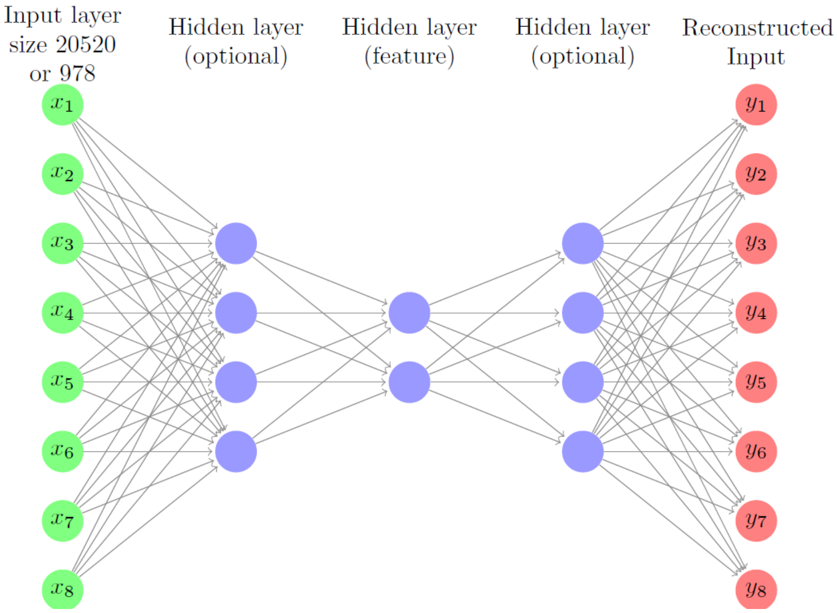
Tools selected
Keras tool, which uses Theano as backend, is adopted to perform the deep learning part because it it easy and fast to start.

Datasets preparation
4 mouse RNA-seq datasets with 3007 samples and 15 cell types in total are adopted. The 15 cell types includes bone-marrow-derived dendritic cells (BMDCs) and embryonic cells from tissues. Among the datasets, 335 samples have original labels and will be treated as testing set. The labels of remaining samples are all expression-based so they are not included in our testing set. In the integrated dataset, 20520 genes are the intersection genes. A subset of genes is also selected as input dimension since 20520 genes might take too much time to train the deep network. LINCS landmark genes is selected because it has only 978 genes and can capture about 80% information of original set of genes. The datasets are all converted to transcript per million (TPM) format and are processed with 0/1 normalization before integration.

Network architectures

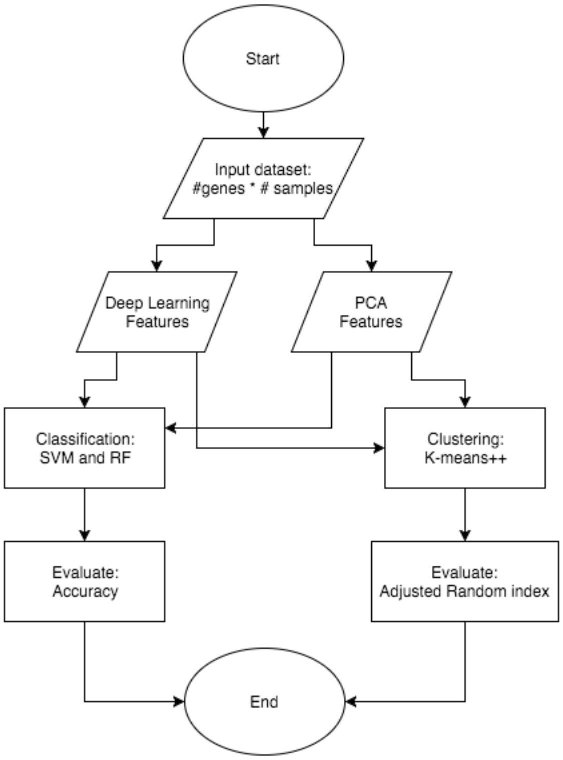
Two architectures of neural networks (NN) have been tried: one hidden layer of size 100 and three hidden layers with the smallest layer of size 100 and size $\sqrt{input\_dimension/100} * 100$ intermediate layers. This size of intermediate layers is choosed because we want to preserve the same compress ratio between each layers. Take figure 1 as an example, the input dimension is 8, the code layer is 2, so the intermediate layer is 4. Also, since there are two values of input dimension, four models in total have been trained. The training batch size is set to 32 and the network is minimized by stochastic gradient descent method with learning rate 0.1.

Figure 1: The neural network structure



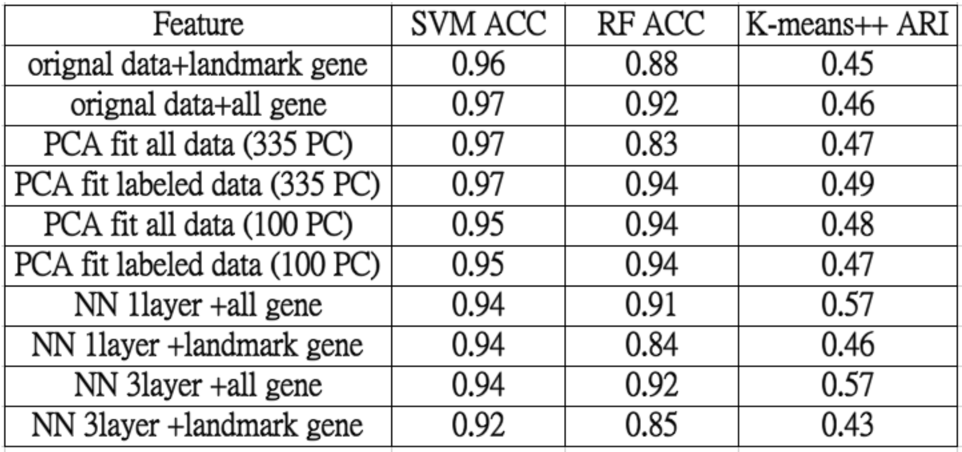
The flowchart of this project is shown in figure 2. To evaluate the performance of deep features, PCA features are selected for comparison and classification and clustering are performed. For classification, two popular methods, Random Forest (RF) and Support Vector Machine (SVM), with four sets of manually chosen parameters are performed on each feature. For clustering, K-means++ is applied to the same set of features. The number of clusters is set to 15, which is the number of cell types in the labeled dataset. 100 random initializations of K-means++ are performed and the best configuration is selected to compute adjusted random index (ARI) as the performance measurement.

Figure 2: The flowchart of methods of this project



3 Result

Figure 3: The classification and clustering results



Feature	SVM ACC	RF ACC	K-means++ ARI
orignal data+landmark gene	0.96	0.88	0.45
orignal data+all gene	0.97	0.92	0.46
PCA fit all data (335 PC)	0.97	0.83	0.47
PCA fit labeled data (335 PC)	0.97	0.94	0.49
PCA fit all data (100 PC)	0.95	0.94	0.48
PCA fit labeled data (100 PC)	0.95	0.94	0.47
NN 1layer +all gene	0.94	0.91	0.57
NN 1layer +landmark gene	0.94	0.84	0.46
NN 3layer +all gene	0.94	0.92	0.57
NN 3layer +landmark gene	0.92	0.85	0.43

Figure 3 shows the experiment results of classification and clustrering of PCA and deep features. ACC denotes accuracy. We have choose the PCA with two different setting, fitting all the data and fitting only labeled data. Also, different number of principle components (PC) coordinates have been selected for PCA features, where 335 is the size of the labeled dataset and 100 is the size of the code layer of neural network. The best performance PCA on SVM is 0.97, which is slightly greater than the best NN performance 0.94. Besides, the best performance PCA on RF is 0.94, which is also slightly greater than the best NN performance 0.92. Therefore, it could be claimed that PCA classification performance are slightly better than NN. Besides, the best performance of PCA is 0.49

where the best performance of NN is 0.57. Clearly, NN is much better than PCA in k-means++ clustering. When compared to the original dataset, classification performance of PCA and NN are no better than original data in SVM, while PCA is better in RF and NN has generally the same performance. In clustering, both PCA and NN features are better than original dataset in general.

4 Conclusion

From the results, it can be observed that PCA is slightly better than NN in classification but NN is much better than PCA in clustering. Since our ultimate goal is discovering new cell types, the result of clustering is more important than classification. We assume that NN can somehow learn non-linear biological information from the input and help clustering performance. Also, the expected result of landmark genes is no better than the set of all genes, and the result corresponds to our assumption. This result shows that if we use only landmark genes to speed up the model training time, we could lose important biological information that can help distinguish new cell types.

5 Future Work

Since neural network has so many parameters to learn, one possible future plan is to integrate more dataset to increase training set size and include more cell types. About the network architecture, it is a good idea to try denoising autoencoder with corrupted input as the deep feature extraction model to prevent overfitting and obtain the denoised codes. Besides, biological information such as protein-protein interaction network and transcription factors can be used to constructed network structures to reduce the number of parameters to train to reduce training time and prevent overfitting.

6 Reference

Gupta, Aman, Haohan Wang, and Madhavi Ganapathiraju. ”Learning structure in gene expression data using deep architectures, with an application to gene clustering.” Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on. IEEE, 2015.

Tan, Jie, et al. ”ADAGE-Based Integration of Publicly Available Pseudomonas aeruginosa Gene Expression Data with Denoising Autoencoders Illuminates Microbe-Host Interactions.” mSystems 1.1 (2016): e00025-15.

F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley and Y. Bengio. Theano: new features and speed improvements. NIPS 2012 deep learning workshop. (BibTex)

J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio. Theano: A CPU and GPU Math Expression Compiler. Proceedings of the Python for Scientific Computing Conference (SciPy) 2010. June 30 - July 3, Austin, TX (BibTeX)

keras tool website:
http://keras.io/

LINCS landmark genes website:
http://support.lincscloud.org/hc/en-us/articles/202092616-The-Landmark-Genes

Deng, Qiaolin, et al. ”Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells.” Science 343.6167 (2014): 193-196.

Tasic, Bosiljka, et al. ”Adult mouse cortical cell taxonomy revealed by single cell transcriptomics.” Nature neuroscience (2016).

Usoskin, Dmitry, et al. ”Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing.” Nature neuroscience 18.1 (2015): 145-153.

Shalek, Alex K., et al. ”Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells.” Nature 498.7453 (2013): 236-240.