Data Integration and Cell Type Classification of scRNA-seq Pancreas Dataset

Xintong Zhai, Kexin Li, Bulun Te

2023-12-15

Introduction

Background

There are about 372 trillion cells inside the body of an adult human being. Different cells share the same DNA, but due to the transcription of RNA, cells show totally different functions[1]. For example, some become skin cells, and others become blood cells. scRNA-seq, one of the main technologies used to get RNA transcriptome, is attained by high-throughput sequencing technology at the single-cell level[1], which has an increasingly wider application, including building reference atlas, identifying new cell types and biomarker genes.

Among which, human cell atlas(HCA) is one of the biggest project to build reference cell atlas. Its downstream analysis contains diagnosing, monitoring and treating disease[2].

For the purpose of realizing such a big project, however, here is a huge challenge remains to be handle, namely batch effect correction. Most scRNA-seq are obtained from different studies, different health states, different organs and even different specials. As a result, gene expression of the same cell in different batches could be biased[1].

There is an urgent need to propose a method that can utilize the cell labeling information of known cell types and directly realize constructing atlas. The ideal performance of scRNA-seq data integration is that, different batches are mixed, and different cell types are separated clearly.

In our study, we focuses on constructing the pancreas reference atlas. Three different methods are used to solve this problem, and evaluation includes umap visualization, ARI and NMI index.

Dataset and Computational Challenges

Pancreas dataset with dimension 6321*34363, which contains more than 200-million umi counts. Datasets are collected from 4 batches(celseq1, celseq2, fluidigmc1, smartseq2)[3].

1. Pancreas dataset is a high dimensional matrix, which made it difficult to select features.
2. The count matrix contains 200-million umi counts data, which is difficult to deal with such a big matrix directly.
3. Most of the datasets are stored in a specific form, each row contains more information about the cell, and each column contains more information about the gene.

Approaches

Baed on the experiment reulsts in Nature Methods, written by Malte D. Luccken et al[4], the proposed methods could be categorized into three types, methods designed for high-dimensional data, MNN and its improvements, and deep learning related ones.

In this study, we attempt to use these three methods, namely unsupervised clustering MNN+kmeans, GLM with Lasso penalty and deep learning. The first two methods are written in R and use Rcpp to accelerate calculation speed. The third method used keras in Python to construct a neural network.

Data Preprocessing

This study filtered out cells with gene expression less than 200 and genes with less than 30 cells. Then, normalization, log transformation and z-score transformation are used to rescale the dataset.

Models

a) GLM with Lasso penalty

Softmax regression is generalization of two level logistic regression has a defination $\hat{y}_i = argmax_{j=1}^{k}(\frac{exp(x_i^T\theta_j)}{\sum_{j=1}^{k} exp(x_i^T\theta_j)})$. In this senario, target loss function with L1 regularization is $L(\theta) = f(\theta) + \lambda g(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} I\{y_i = j\}ln(\frac{e^{x_i^T\theta_j}}{\sum_{l=1}^{k} x_i^T\theta_l}) + \lambda\sum_{i=1}^{k}||\theta_i||_1$. The proximal gradient according to the paper proposing the fista algorithm[5] has the form $\theta^{k+1} = p_{\lambda,\frac{1}{L}}(\theta^k) = (|u| - \frac{\lambda}{L})_+ sgn(u)$ where $u = \theta^k - \frac{1}{L}\nabla f(\theta^k)$ solving from the equation[5] $\nabla f(\theta^k) + L(\theta^{k+1} - \theta^k) + \gamma(\theta^k) = 0, \quad \gamma \in \partial g(\theta^{k+1})$.

Due to the unbalanced data, some of the cell types are merged into one class called combined class. After finetuning model with λ ranges from 0 to 1, with 0.001 step size, we choose to set $\lambda = 0.033$ and according to the F1 score and balanced accuracy. The iteration plots are as follows, and also the result of the compressed coefficients are plotted in the following figure.

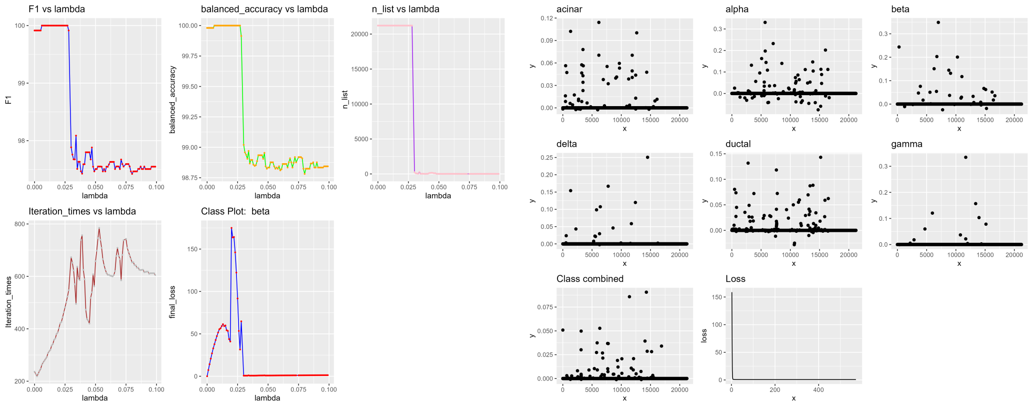


Figure 1: key results over lambda and non-zero coefficients

After deciding the λ, we find the first five genes influential for classifying specific cell types in terms of the absolute value of parameters. The results are as follows:

acinar	GSTA2	BCAT1	RNASE1	CTRB2	GSTA1
****	0.114	0.102	0.1	0.078	0.07
alpha	GCG	IRX2	TTR	GC	FAP
****	0.33	0.232	0.202	0.197	0.162
beta	INS	ADCYAP1	IAPP	NPTX2	HADH
****	0.349	0.244	0.203	0.201	0.151
delta	SST	LEPR	BCHE	RBP4	HHEX
****	0.251	0.167	0.154	0.12	0.107
ductal	TINAGL1	CFTR	KRT19	SLC4A4	SERPING1
****	0.143	0.132	0.118	0.088	0.087
gamma	PPY	SERTM1	GPC5-AS1	SLITRK6	THSD7A
****	0.334	0.157	0.121	0.103	0.078
remaining	SPARC	PMP22	HCLS1	A2M	COL4A1
****	0.091	0.086	0.053	0.051	0.05

The UMAP result is as follows:

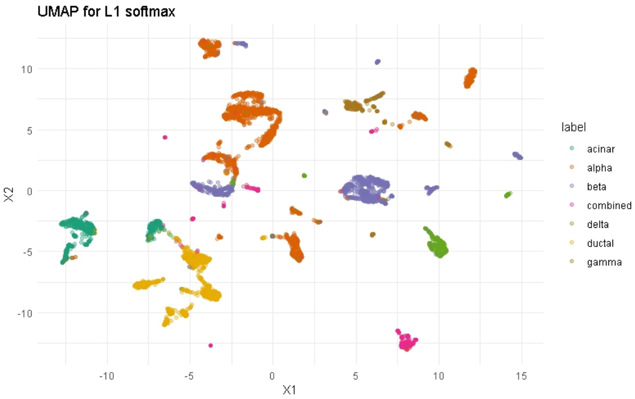


Figure 2: umap visualization of Softmax Classification Model with L1 Regularization

b) Unsupervised learning clustering: Mutual Nearest Neighbors + kmeans

In our exploration of scRNA data, we encounter challenges stemming from the generation of data in separate batches, leading to batch effects. To address this, we utilize the Mutual Nearest Neighbors (MNN) method. Considering two batches of cells, we calculate the proximity of each cell to those in the other batch. The top k nearest neighbors for each cell are then selected. Cells that appear in each other's top k nearest neighbors are termed "mutual nearest neighbors."

In our approach, we designate Batch 1 as the reference. We project the new data onto this reference, subsequently computing the weighted means of differences among these neighbors to capture batch differences $\vec{u}(x) = \sum_m \vec{v}^{ml} W(x,m) / \sum_m W(x,m)$. By subtracting this difference from the new group, we obtain new features for these cells $\vec{x} = \vec{x} - \vec{u}(x)$. This new data is then combined with the original reference group, creating a revised reference for subsequent steps. [6]

Utilizing the Pancreas dataset after PCA, with dimensions of 6321*50, and choosing the number of clusters as 13, we follow a two-step approach:

1. Employ MNN with k = 20 to identify the k nearest neighbors for each cell, effectively removing batch effects and generating new features.

2. Apply K-means clustering to the transformed data, achieving an average accuracy of 0.63.

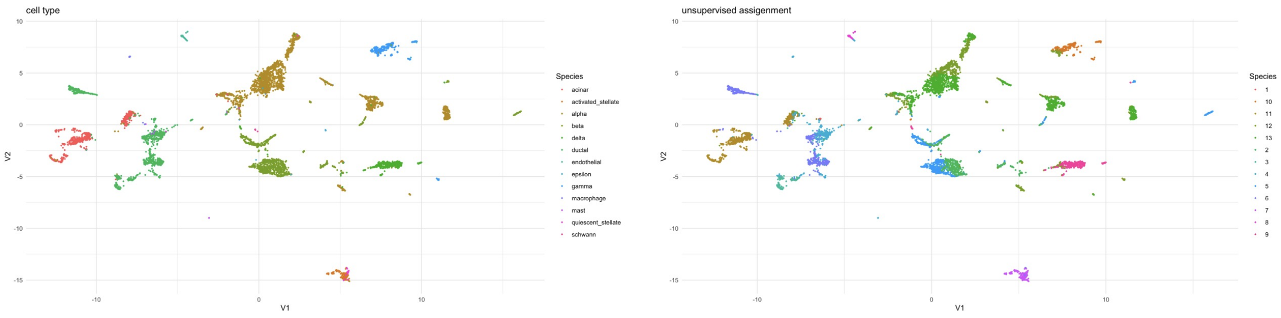


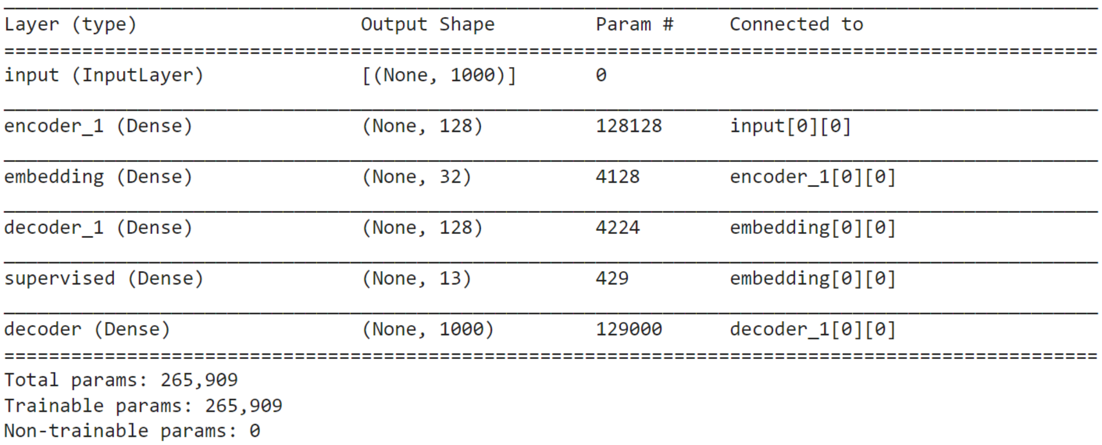
Figure 3: umap visualization of MNN + kmeans

Visualization of the original cell types using UMAP and the clustering results using UMAP indicates promising outcomes. Despite these promising results, MNN has limitations. It requires the reference group to cover all cell types, assumes linear relationships between different batches (which may not hold true in many cases), and its performance may degrade if batch differences vary significantly.

c) Transfer Learning and Multi-task Neural Network

An autoencoder neural network is built and trained at first, and then expanded this model to two tasks, reconstruction and cell type identification.

The structure of the multitask model is shown as below:



```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input (InputLayer)              [(None, 1000)]       0
__________________________________________________________________________________________________
encoder_1 (Dense)               (None, 128)          128128      input[0][0]
__________________________________________________________________________________________________
embedding (Dense)               (None, 32)           4128        encoder_1[0][0]
__________________________________________________________________________________________________
decoder_1 (Dense)               (None, 128)          4224        embedding[0][0]
__________________________________________________________________________________________________
supervised (Dense)              (None, 13)           429         embedding[0][0]
__________________________________________________________________________________________________
decoder (Dense)                 (None, 1000)         129000      decoder_1[0][0]
==================================================================================================
Total params: 265,909
Trainable params: 265,909
Non-trainable params: 0
```

Figure 4: Model Structure

Total loss of the model is the weighted sum of two parts, which is shown as below:

$$TotalLoss = MSE + \beta H(p, q),$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

$$H(p, q) = \sum_{i} p(i) log(\frac{1}{q(i)}),$$

where p is the probability of the true distribution.

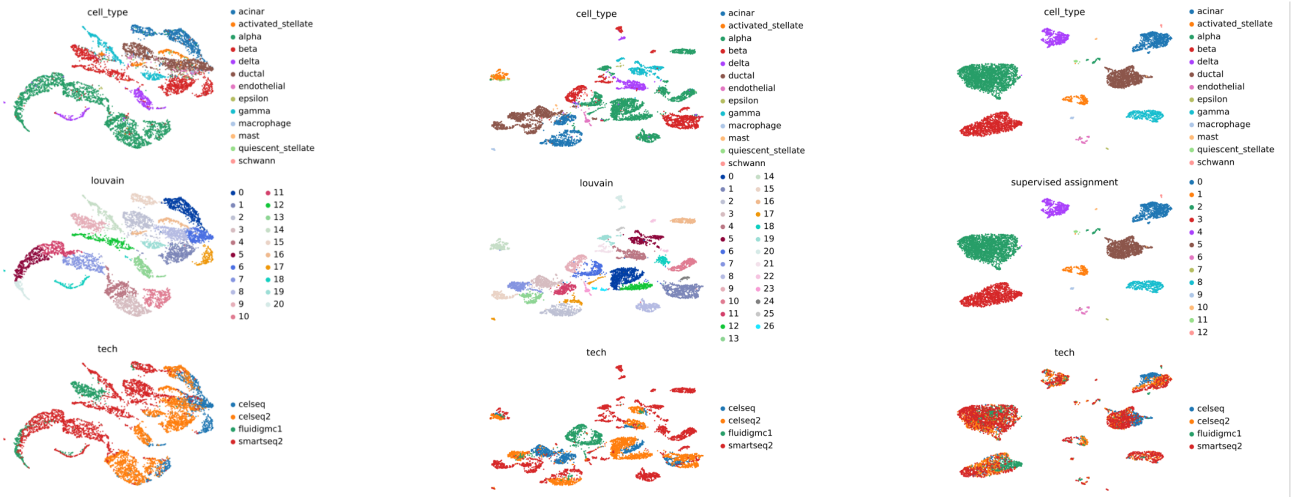


Figure 5: Comparison of umap results

The first column of the figure above is the distribution of cell type, clustering result and batch effect of the origin dataset, the second column is that of the dataset after preprocessing, and the third column is that of the embedding layer.

The last column shows that, the multitask neural network clearly separate cells from different types, give them the correct labels, and mix cells from four batches well.

Comparison among Models

In experiments, the model accuracy are recorded in order to carry out a comparison of different approaches. We will use Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) for model assessment. ARI measures how well the predicted clusters align with the true clusters, considering chance, providing an adjusted accuracy assessment. NMI evaluates clustering quality by measuring shared information between true and predicted clusters, normalized by the uncertainty in individual clusters.

Model	ARI	NMI
Unsupervised Clustering MNN + kmeans	0.5717	0.6139
GLM with Lasso penalty	0.9502	0.9146
Transfer Learning and Multi-task Neural Network	1.0000	1.0000

While the unsupervised learning method provided a baseline performance, the Softmax with L1 Regularization and Neural Network approaches significantly outperformed it. It is noteworthy that the Neural Network achieved perfect scores in both the ARI and NMI metrics. This implies that even with some categories having very few instances, the Neural Network successfully classified all categories correctly.

Conclusion

In this cell type classification project, various models were employed and assessed. We leveraged three distinct methods: MNN + kmeans, GLM with Lasso penalty, and Multi-task Neural Network. MNN + kmeans offered an unsupervised learning approach that considers batch effects. The GLM with Lasso demonstrated commendable classification results with interpretability, shedding light on which genes play a crucial role in determining cell categories. Notably, the Multi-task Neural Network exhibited exceptional classification performance, achieving metrics at a perfect 100%. These findings showcase the effectiveness of different approaches in addressing cell type classification challenges and highlight the promising results achieved by the Multi-task Neural Network.

Project Github: https://github.com/kexiin/CellTypeClassification

Contribution: Introduction and data processing was contributed by Xintong. GLM with Lasso penalty was contributed by Bulun, MNN + kmeans was contributed by Kexin, Multi-task Neural Network was contributed by Xintong. The final report was collaboratively drafted by all team members.

Reference

[1] Single Cell Course. (n.d.). Analysis of single-cell RNA-seq data. https://www.singlecellcourse.org/introduction-to-single-cell-rna-seq.html

[2] Human Cell Atlas. (n.d.). https://www.humancellatlas.org/

[3] Grün, D., Muraro, M. J., Boisset, J.-C., et al. (2016). Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature*, 529(7584), 321–327.

[4] Luecken, M. D., Büttner, M., Chaichoompu, K., et al. (2022). Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19, 41–50. https://doi.org/10.1038/s41592-021-01206-x

[5] Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1), 183–202.

[6] Haghverdi, L., Lun, A., Morgan, M., et al. (2018). Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology*, 36(5), 421–427. https://doi.org/10.1038/nbt.4091