Report on Single Cell RNA Sequence Analysis Project

Members 李钟楷 周守琛 齐红军

 Task Description

Single-cell RNA-seq analysis is a rapidly evolving field at the forefront of transcriptomic research, used in high-throughput developmental studies and rare transcript studies to examine cell heterogeneity within a populations of cells.

In this project, our task revolves around the preprocessing and clustering process of scRNA-seq, which requires the usage of a number of python packages for scientific statistics analysis, such as pandas, matplotlib, scikit-learn.

---

 Our Team's Timeline

Week 1

In week 1, we got familiar with the usage of Git, a version control system.

Week 2-3

Week 2-3 was when we embarked on handling our task for real. Through week 2-3, our task were separated into these 3 following steps:

(1) Preprocessing. We read the data into Jupyter Notebook and filtered the raw data, so as to provide a high-quality dataset for downstream analysis.
(2) Dimensionality reduction. We reduced the original data with an enormous number of distinct dimensions(at least 10000+) to a 3 column dataframe only 2 principle components, which greatly facilitated our following clustering step. Here we called scikit-learn, a python machine learning package to implement PCA and t-SNE separately.
(3) Clustering. We called the K-means clusterer in scikit-learn to implement clustering to our data and two evaluation functions to evaluate the performance of K-means.

These were relatively simply following the instructions so that it is not worthy of much attention.

Week 4

In week 4 we integrated all our previous works to one pipeline .py file and strove to improve the performance of our program. Our directions of improvement were divided into 3:

1 Eliminating the horseshoe effect of PCA.

When processing non-linear data, the axis of the second principal component are twisted, thus distorting the results. We managed to apply normalization and log-transformation by referring to the official online tutorial of Galaxy Training and Scanpy.

2 Adding the number of Principle Components in PCA.

By figuring out the principles of PCA, we attempted to add the number of principal components, thus greatly improving the performance.

3 Improving the clustering algorithm.

(1) Manually selecting the proper starting points of K-means algorithm.
(2) Applying other clustering algorithms provided by scikit-learn. By gaining some basic understandings about machine learning, we attempted to apply several different clustering algorithms in the user guide of scikit-learn, be they supervised or unsupervised. Here is the list of clustering algorithms that we've tried:

Unsupervised

- K-Means
- Agglomerative Clustering
- Spectral Clustering
- Mean Shift
- Affinity Propagation
- Mini Batch K-Means

Supervised

- Linear Discriminant Analysis
- K-Neighbors Classifier
- Quadratic Discriminant Analysis
- SVC

Among those algorithms, we selected K-means for unsupervised learning and LDA for supervised learning in our pipeline Jupyter Notebook, but the usage of them were included in the cluster function of analysis.py in the form of annotations.

(4)Adjusting the filtering conditions of the preprocessing procedure.

- Result: Pipeline

We adjusted the usage of the function “whole_pipeline” by adding two parameters: pca_n is the number of principal components you choose to keep when performing PCA; algorithm is the choice of algorithm you use, the options of which are namely “Supervised LDA” and “Unsupervised k-means”. We decided to separate them into two cells in our Jupyter file. Overall supervised LDA may have better performances.

Result:

PCA-processed(keeping 10 PCs), K-Means clustered data have an average NMI of around 0.7 and an average ARI of 0.6.

TSNE-processed, K-Means clustered data have an average NMI of around 0.6 and an average ARI of 0.5.

PCA-processed(keeping 10 PCs), LDA clustered data have an average NMI of around 0.8 and an average ARI of 0.75.

TSNE-processed, LDA clustered data have an average NMI of around 0.65 and an average ARI of 0.6.

We kept 10 PCs in our, considering that this won’t cost much calculation resources(at least far faster than tsne).However, keeping 3 PCs for LDA clustered data would also give an NMI of around 0.7, which is also decent.