Reimplementing and Improving in RaceID

Pei Lin
School of Information Science and Technology
ShanghaiTech University

***Abstract*—Clustering analysis has been widely applied to single cell RNA-sequencing data to discover cell types and cell states, and many algorithms have been developed in recent years. RaceID was published in Nature 2014 which offers an effective way to identify rare cell types such as stem cells, short-lived progenitors, cancer stem cells, or circulating tumour cells. RaceID uses specific data filtering methods and takes K-means as it clustering method during which gap statistic is adopted to affirm the K.**

Firstly, I reimplemented the major functions of RaceID including data filtering, data clustering and dimension reduction by Python. Furthermore, paralleled calculating was used to accelerate the process and the finish time was reduced by nearly 49%. After testing on the original dataset, I found another dataset from a paper and plot the outcome after adjusting the metric.

I. Introduction

With the rapid development of scRNA-seq technology, large amounts of scRNA-seq data have been generated, which provide great opportunities and challenges to computational biology. As the basic constructing unit of organisms, cells vary greatly in types and states. Computational analysis of scRNA-seq data can help to understand biological processes and their mechanisms.

As shown in Fig.1, the whole pipeline of RaceID starts with data filtering and scaling which is the most important process and the meaningless cells and RNA are filtered without destroying the sparsity. For K-means algorithm, because clustering problem is not a convex problem and always converges to a local optimal solution, many methods for choosing K and cluster centers are designed. In RaceID, gap statistic which uses Monte Carlo estimation is adopted after sampling several times to find the best K. However because the sampling random, the result can also be random only if cost more time to sample more times. After different metrics (Euclidean, Pearson, Spearman) are used, the outcome of K-means can be varied so that RaceID uses silhouette score to evaluate the results. For RNA-seq data is in high dimension and the clusters can not be directly perceived, dimension reduction must be applied to visualized the outcome. RaceID uses t-SNE which is a manifold learning method to finish dimension reduction.

I reimplemented the above process in Python because Python can be seen as a faster and more stable language which also supports many science package such as SKlearn(a machine learning package which offers clustering functions and dimension reduction functions). Although, the plots in Python does not look as professional and awesome as plots in R.

To accelerate the calculation, I use MapReduce theory and the package in Python named threading to paralleled the program. In each loop, each iteration is dispatched to an independent thread and gather the outcome together.

The main results and contributions of this report are summarized as follows:

- **Data**.
- **Clustering**.
- **Dimension reduction and visualization**.
- **Paralle programming**.

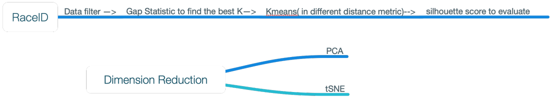

RaceID
Data filter —>
Gap Statistic to find the best K—>
Kmeans(in different distance metric)-->
silhouette score to evaluate
Dimension Reduction
PCA
tSNE

Fig. 1. Workflow of RaceID.

II. data

The first dataset is found in original paper[1] and the other is from Liying Yan[2]. Both of this data is based on unique molecular identifier(UMI) which is a method that uses molecular tags to detect and quantify unique mRNA transcripts and the big value presents that the RNA express more.

Datasets are sparse and sparsity can be regarded as a good characteristic when matrix calculations are required. RaceID uses median to normalize and scale the data which has not been adopted in most scaling algorithms although it does not destroy the sparsity:

$$RaceID\ Scaling = \frac{data_{ij}}{\sum_i data_{ij}} * median$$

To improve this, original way is replaced by Log. The reason is that logarithm does not change the relative relationships among data but compresses the scale of the variables. For distance-based clustering methods such as K-means, feature scaling impacts distance measure between cells. For example, if one of the features has a broad range of values, the Euclidean distance will be dominated by this particular feature. Thus, the ranges of all features should be normalized so that each feature contributes almost equally to the final distance. On the other hand, Log is the most powerful scaling tool as far as scRNA-seq data is considered. For example, SC3, BISCUIT and NMF all use Log.

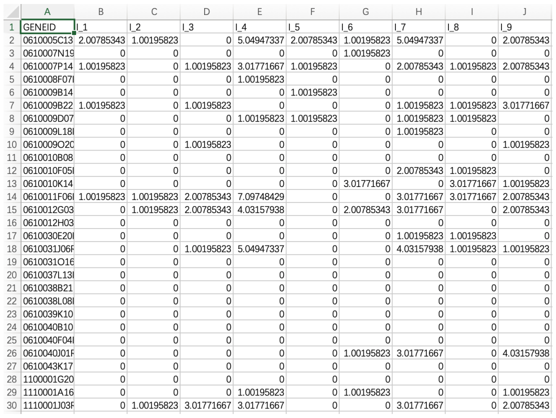
For data filtering, meaningless and insignificant RNA should be filtered by setting threshold:
Fiter RNA:

$$\#(RNA > ThresholdOfExpression) > ThresholdOfNumbers$$

which will filter the housekeeping genes.
Filter cell:

$$(\sum RNA) < Threshold$$

which express too less.



	A	B	C	D	E	F	G	H	I	J
1	GENEID	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9
2	0610005C13	2.00785343	1.00195823	0	5.04947337	2.00785343	1.00195823	5.04947337	0	2.00785343
3	0610007N19	0	0	0	0	0	1.00195823	0	0	0
4	0610007P14	1.00195823	0	1.00195823	3.01771667	1.00195823	0	2.00785343	1.00195823	2.00785343
5	0610008F07I	0	0	0	1.00195823	0	0	0	0	0
6	0610009B14	0	0	0	0	1.00195823	0	0	0	0
7	0610009B22	1.00195823	0	1.00195823	0	0	0	1.00195823	1.00195823	3.01771667
8	0610009D07	0	0	0	1.00195823	1.00195823	0	1.00195823	1.00195823	0
9	0610009L18I	0	0	0	0	0	0	1.00195823	0	0
10	0610009O20	0	0	1.00195823	0	0	0	0	0	1.00195823
11	0610010B08	0	0	0	0	0	0	0	0	0
12	0610010F05I	0	0	0	0	0	0	2.00785343	1.00195823	0
13	0610010K14	0	0	0	0	0	3.01771667	0	3.01771667	1.00195823
14	0610011F06I	1.00195823	1.00195823	2.00785343	7.09748429	0	0	3.01771667	3.01771667	2.00785343
15	0610012G03	0	1.00195823	2.00785343	4.03157938	0	2.00785343	3.01771667	0	2.00785343
16	0610012H03	0	0	0	0	0	0	0	0	0
17	0610030E20I	0	0	0	0	0	0	1.00195823	1.00195823	0
18	0610031J06R	0	0	1.00195823	5.04947337	0	0	4.03157938	1.00195823	1.00195823
19	0610031O16	0	0	0	0	0	0	0	0	0
20	0610037L13I	0	0	0	0	0	0	0	0	0
21	0610038B21	0	0	0	0	0	0	0	0	0
22	0610038L08I	0	0	0	0	0	0	0	0	0
23	0610039K10	0	0	0	0	0	0	0	0	0
24	0610040B10	0	0	0	0	0	0	0	0	0
25	0610040F04I	0	0	0	0	0	0	0	0	0
26	0610040J01R	0	0	0	0	0	1.00195823	3.01771667	0	4.03157938
27	0610043K17	0	0	0	0	0	0	0	0	0
28	1100001G20	0	0	0	0	0	0	0	0	0
29	1110001A16	0	0	0	1.00195823	0	1.00195823	0	0	1.00195823
30	1110001J03R	0	1.00195823	3.01771667	3.01771667	0	0	3.01771667	0	2.00785343

Fig. 2. Data format of RaceID. Column presents cells and rows presents rows. Each number stands for UMI(unique molecular identifier) of each RNA in each cell.

III. Clustering

The main purpose of RaceID is using clustering algorithms to find a special cluster which can present a rare cell type. K-means is used in RaceID with applying different metrics such as Euclidean, Spearman and Pearson. All of these metrics use different methods to calculate the distance between each points which means the similarity of points can be gotten from different ways. Certainly, the results will be varied among different metrics.

K-means is known as as a non-convex problem which always converges to a local optimal solution without choosing a proper K and initial centers. To find the best K, gap statistic is used to evaluate the performance and estimate the best K.

$$Gap_n(k) = E(logW^*) - logW$$

W presents the sum of distance of each point after normalized while W* is the distance of sampled data. The core idea of gap statistic is Monte Carlo which samples enough random points between the range of origin dataset for several times. Gap value estimate the performance of clustering in each K abstractly by contrast the real data with sampled data in normal distribution. Higher gap value means in such K condition, the outcome of K-means using K clusters has a great difference with the clusters in null reference distribution and it presents a better performance. The estimate of the optimal clusters will be value that maximize the gap statistic.In Fig.3, the detailed process of gap statistic is introduced.

However, gap statistic's time complexity can be higher when user choose more samples and sampling for more times to get better and more accurate estimation. So, parallel calculation is adopted to accelerate and will be discussed in V.

Since gap statistic is a random algorithm, the outcome can change with different random seeds used in sampling.

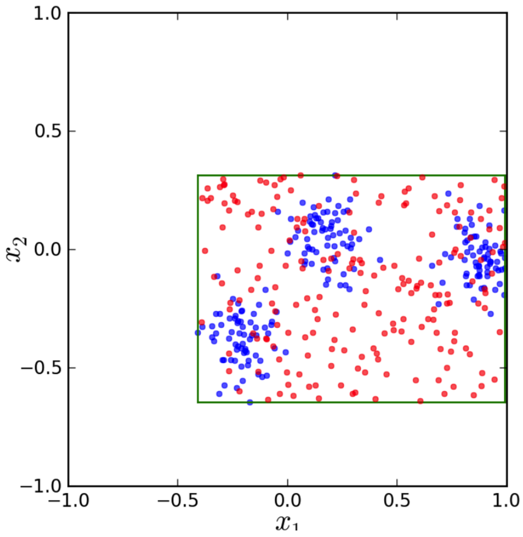


Fig. 3. Gap statistic in 2-dimension. Using blue points to present the real data set and red points stands for random sample data with Gaussian distribution.The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. The estimate of the optimal clusters will be value that maximize the gap statistic.

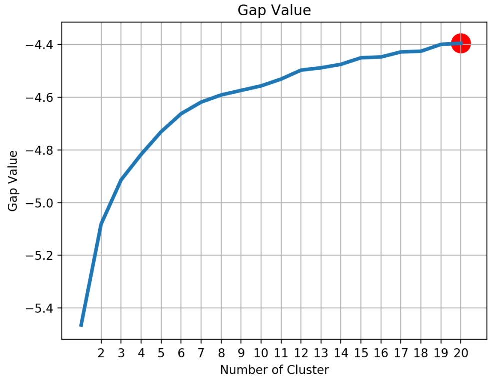


Fig. 4. It is the results of my experiments under $max_K = 20$.

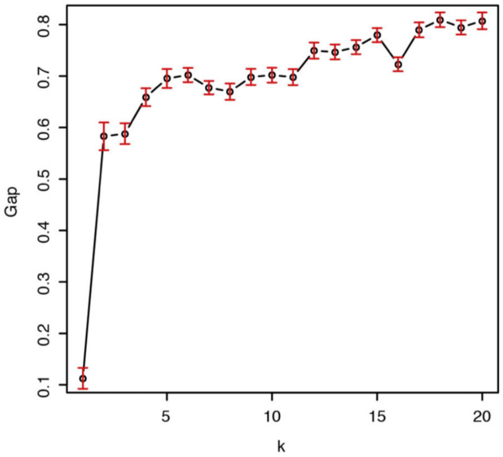


Fig. 5. It is the original results of RaceID under $max_K = 20$.

IV. Dimension reduction and visualization

The main dimension reduction algorithms can be classified into two types, the one for linear data, the other for nonlinear and even manifold. Principal components analysis(PCA) and t-distributed stochastic neighbor embedding(t-SNE) are adopted in RaceID.

PCA is widely used in linear data ming which helps to get the linear characteristics and expose the inner information. The main idea of PCA is singular value decomposition(SVD) and choose the eigenvectors which corresponds to the maxima eigenvalues. Along these chosen eigenvectors, the data can project in the direction with biggest weight.

$$X(data) = U\Sigma V^T$$

However, the results of PCA is terrible in RNA-seq data. The reason is that the data is not linear, even biologists can not explain the meaning of linear combinations of these dimensions in which presents the single RNA's expression.

t-SNE is a manifold learning algorithm which uses a manifold model to keep the characteristic in locality by decreasing the weight of points which is far.

The similarity of data point x_j to datapoint x_i is the conditional probability ,$P_{j|i}$, that x_i would pick x_j at its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i. σ_i is the variance of the Gaussian distribution that is centered on datapoint x_i.

$$P_{j|i} = \frac{\exp(-||x_i - x_j||^2/2{\sigma_i}^2)}{\sum_k \exp(-||x_i - x_k||^2/2{\sigma_i}^2)}$$

Dimension reduction can be regarded as a process in which data in high dimension is mapped into the data in low dimension. $Q_{j|i}$ is the similarity of y_i and y_j where y_i corresponds to x_i and y_j is from previous x_j. We set the variance of Gaussian that is employed in the computation of the conditional probability $Qj|i$ to $\frac{1}{\sqrt{2}}$. Setting the variance in low dimension only changes the scale without any other side effect.

$$Q_{j|i} = \frac{\exp(-||y_i - y_j||^2)}{\sum_k \exp(-||y_i - y_k||^2)}$$

Finally, SNE minimize the sum of Kullback-Leibler divergence s over all data points which can estimate the mismatch between two sets.

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}}$$

σ_i is the last parameter to select. SNE defines a parameter named perplexity and performs a binary search for the value of σ_i that produces a P_i with a fixed perplexity.

$$Perp(P_i) = 2^{H(P_i)}$$

$H(P_i)$ is Shannon entropy of P_i measured in bits:

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}$$

After trying, choose a fixed perplexity.

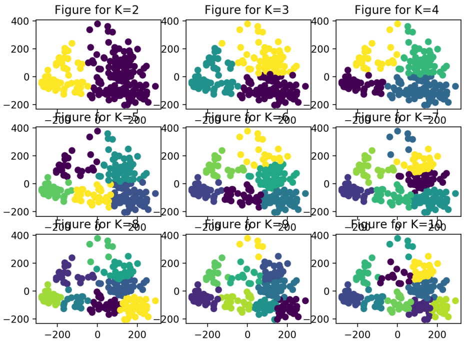


Fig. 6. The results of 2-dimension clustering after PCA in $max_K = 10$. Different clusters are presented in different colors. Obviously, points gather together and hard to cluster.

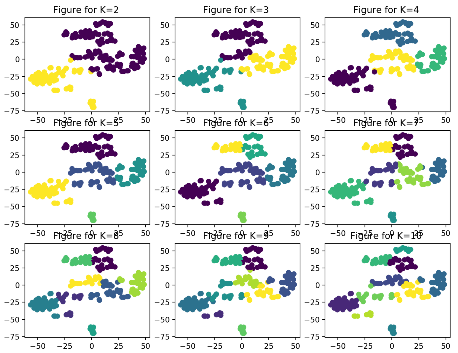


Fig. 7. The t-SNE results from my expriment in 2-dimension with perplexity=29.8. Different colors presents different clusters, the outcome looks better than PCA's.

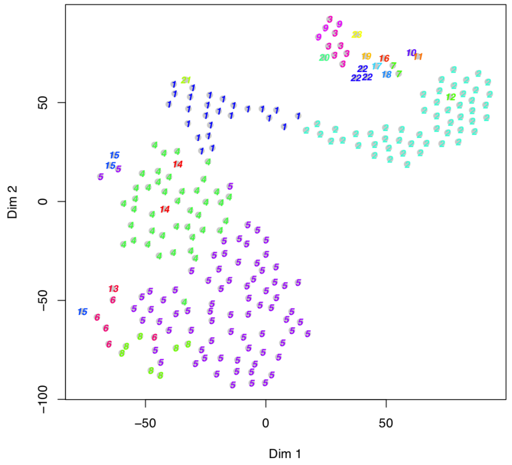
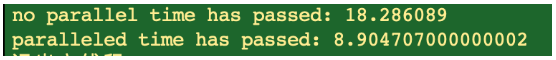


Fig. 8. The t-SNE results from RaceID in 2-dimension with default perplexity=30.

V. Paralle programming

To optimize and accelerate the calculation, parallel programming is comprehensively applied in multi-threads computers.

MapReduce is the most famous parallel model published in 2008. I put MapReduce model in my algorithm which helps to accelerate getting gap value and best K. In loop, each iteration is despatched to a child-thread and all of these threads calculate in the same data simultaneously. After the calculation done, all the outputs are gathered to store in a same place.



```
no parallel time has passed: 18.286089
paralleled time has passed: 8.904707000000002
```

Fig. 9. The performance of parallel calculation. The first line printed the time used without multi-threads and the second line presents the time used in parallel. The environments to calculate these two results are same. The same work finished in R needs 14.2576s that costs more 60% time.

VI. Reference

[1]*Single-cell messenger RNA sequencing reveals rare intestinal cell types*, Dominic Gru, Anna Lyubimova, Lennart Kester

[2]*Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells*, Liying Yan, Mingyu Yang, Hongshan Guo, Lu Yang

[3]*Estimating the number of clusters in a data set via gap statistic*, Robert Tibshirani, Guenther Weather

[4]*Visualizing data using t-SNE*, Laurens van der Maaten, Geoffery Hinton

[5]*Principal component analysis*,SvanteWold, Kim Esbensen, Paul Geladi

[6]*k-means++: The advantages of careful seeding*,D Arthur, S Vassilvitskii

[7]*MapReduce: simplified data processing on large clusters*, J Dean, S Ghemawat