Project 2: Single-Cell Gene Expressions

Qin Huang, Mengxiao Luan, Ou Sha, Kindle Zhang

1 Introduction

Single-cell RNA sequencing (ScRNA-seq) technologies have revolutionized biology by measuring gene expression levels of thousands of genes at the individual cell level. This technology has greatly enhanced our comprehension of cellular functions. However, the variability in gene expression levels among individual cells of the same type, a phenomenon known as cell heterogeneity, presents a complex issue. Clustering analysis can help navigate this complexity and identify sub-types in cases of cell heterogeneity.

Breast cancer is the second most common cancer in women. The intersection of breast cancer and ScRNA-seq presents a powerful pathway for exploring the dynamic nature and diversity within cancer tumors at the cellular level [1]. In this project, we aim to identify gene clusters in the data and explore gene expression signatures in each of them. Principal Component Analysis (PCA) is first used to determine the number of principal components needed to capture the variability of gene expression. A Gaussian-Mixture model is then fitted to the principal component scores. Expectation-Maximization (EM) Algorithm is implemented to estimate the model and identify the cell clusters. Ultimately, Support Vector Machine Recurve Feature Elimination (SVM-RFE) algorithm is used to find the gene signatures in each of the cell clusters.

2 Principal Component Analysis

The raw data contains gene expression levels of 558 genes (columns) from 716 cells (rows) of breast cancer tumors. Principal Component Analysis (PCA) is a flexible statistical method for dimensional reduction. PCA condenses a dataset of cases and variables into its essential features, known as principal components. The first principal component is the direction in space along which projections have the largest variance. The second principal component is the direction which maximizes variance among all directions orthogonal to the first principal. In summary, the kth component is the variance maximizing direction orthogonal to the previous k-1 components [2]. Mathematics behind principal components mentioned above: Stack n data vectors into an $n * p$ matrix, x. The projections are xw, an $n * 1$ matrix. The variance is:

$$\sigma^2_{\vec{w}} = \frac{1}{n}\sum(\vec{x_i} * \vec{w})^2 = \frac{1}{n}(xw)^T(xw) = w^T\frac{x^Tx}{n}w = w^Tvw \tag{1}$$

To chose a unit vector $\vec{w}$ so as to maximize $\sigma^2_{\vec{w}}$, the Lagrange multiplier λ is used.

$$\begin{aligned} L(w,\lambda) &= \sigma^2_w - \lambda(w^Tw - 1) \\ \frac{\partial L}{\partial \lambda} &= w^Tw - 1 \\ \frac{\partial L}{\partial w} &= 2vw - 2\lambda w \end{aligned} \tag{2}$$

Setting the derivatives to zero at the optimum,

$$\begin{aligned} w^Tw &= 1 \\ vw &= \lambda w \end{aligned} \tag{3}$$

Thus, desired vector w is an eigenvector of the covariance matrix v, and the maximizing vector will be the one associated with the largest eigenvalue λ.

A scree plot showing eigenvalues and the proportion of variance explained is then can be used to determine the number of principal components that provided the most information without significant loss of information. From the scree plot, the 'elbow' point signifies where the eigenvalues plateau, typically with eigenvalues exceeding 1. Components to the left of this point should be considered significant and retained. Based on Figure 1, the 'elbow' point is identified at the 24th principal component, where the eigenvalues surpass 1 for the first 24 principal components. The 24 principal components are subsequently utilized for implementing the EM algorithm.

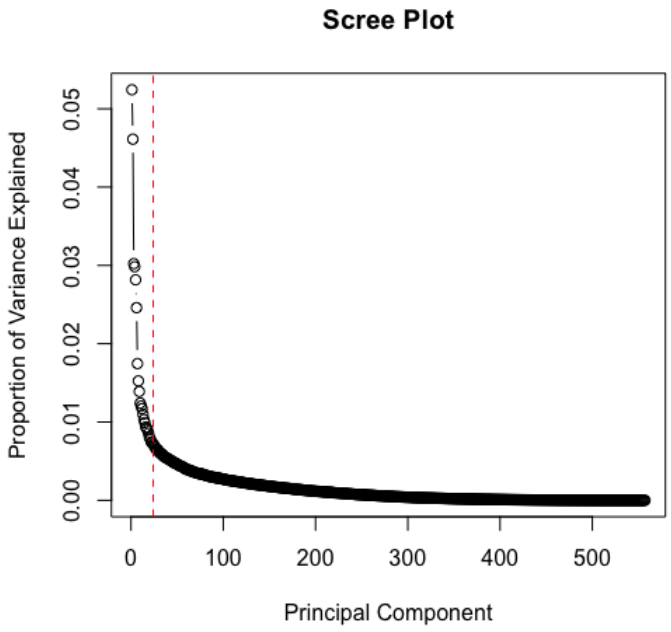


Figure 1: PCA scree plot

3 EM Algorithm

3.1 Methodology

After the PCA analysis, we got a brand new data set with n observations and each one with a 24 dimension. Then we will use the EM algorithm to classify all the cells in different groups and estimate the parameters included in our model. First, we assume all the cells follow a Gaussian Mixture model like this:

$$x_i \begin{cases} N(\mu_1, \sum_1) & \text{, with probability } p_1 \\ N(\mu_2, \sum_2) & \text{, with probability } p_2 \\ ... & \\ N(\mu_k, \sum_k) & \text{, with probability } p_k \end{cases} \tag{4}$$

We assume there are k clusters. Then according to the equation we can derive the likelihood function of x_i here:

$$L(\theta, \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) = \prod_{i=1}^{n} \sum_{j=1}^{k} p_j f(\mathbf{x}_i; \mu_j, \sum_j) \tag{5}$$

After introducing a latent variable $\gamma_{i,j}$, which can show whether i_{th} observation belongs to j_{th} cluster, we can get its formula and its pdf. By using equations above, we can derive the final complete log likelihood function as follows:

$$\begin{aligned} r_{i,j} &= I\{\mathbf{x}_i \text{ belongs to cluster } j\} \\ l(\theta; \mathbf{x}, \mathbf{r}) &= \sum_{i=1}^{n} \sum_{j=1}^{k} r_{i,j} \left[\log p_i + \log f(\mathbf{x}_i; \mu_j, \sum_j) \right] \end{aligned} \tag{6}$$

We will input a starting value in our EM model. We choose an observation randomly from dataset as my starting μ and a diagonal matrix as starting $\sum$, and assume all the p are equal. In the E-step, we calculate the posterior probability using the current parameters:

$$\gamma_{i,j}^{(t)} = \frac{\pi_k N(x_i|\mu_k, \sum_k)}{\sum_{j=1}^{K} \pi_k N(x_i|\mu_k, \sum_k)} \tag{7}$$

In the M-step, we will use the find the parameters which can maximize the log likelihood. We derivative the log likelihood function and make marginal pdf of all parameters equal to 0:

$$\begin{aligned}
\mu_k^{(t+1)} &= \frac{1}{n_k}\sum_{i=1}^{n}\gamma_{i,k}\mathbf{x}_i \\
\sum_k^{(t+1)} &= \frac{1}{n_k}\sum_{i=1}^{n}\gamma_{i,k}(\mathbf{x}_i-\mu_k^{(t+1)})(\mathbf{x}_i-\mu_k^{(t+1)})^T \\
p_k^{(t+1)} &= \frac{n_k}{n}
\end{aligned} \tag{8}$$

Then we repeat the above steps for hundred times and stop to get a final result.

3.1.1 Update to the model

- When finding the starting μ, replace original method (pick randomly in dataset) with a **Kmeans** method, and repeat for 20 times to reduce randomness.
- At first we iterate the EM step for 100 times, but its inefficient and sometimes can't lead to a result successfully. We will jump out of the iteration when the log likelihood converges to a stable value.
- When calculating the pdf of x_i given other parameters, we always have a problem with extremely small result which will lead to a Nan value. We use the Log-sum-exp method to solve this problem.
- The dataset we classify is of large size and so complicated that it will spend us lot of time to run the code. Therefore, we use the parallel computing in our calculating process.
- We find that sometimes some cluster has no cells in it at all when the **ncluster** is a large number like 8, 9, 10. We control our biggest **ncluster** number as 7.

When choosing the starting μ, the reason why I choose 20 times as my **Kmeans** method parameter is that it's the most efficient with the lowest running time. We can find according to this picture that the more times we repeated, the staring value we got will be more stable and we need less steps to make log likelihood converge. However, the time didn't decrease strictly, so we should find which is the optimal repeated times.

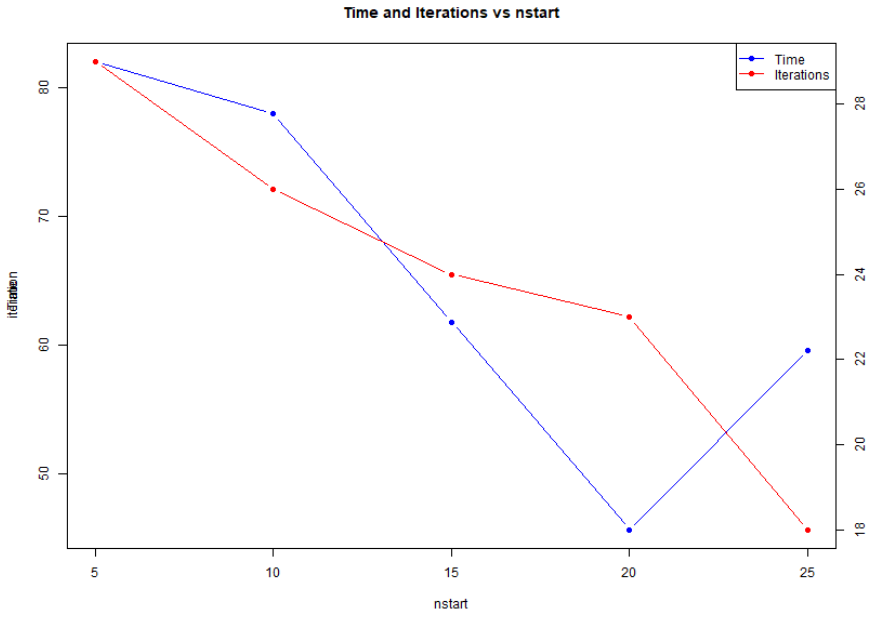
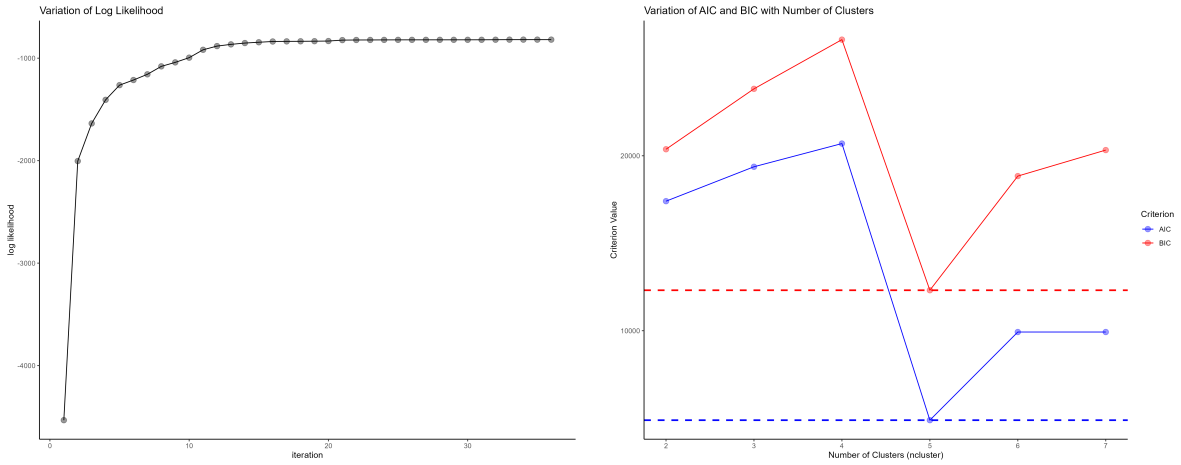


Figure 2: Kmeans repeated times vs code running time

3.2 Result

First we checked the variation trend of log likelihood during iteration to make sure it could converge as expected before reaching the maximum number of iterative times. As shown in Figure 3(a), the log likelihood reached convergence after 20 iterations when the cluster number equals to 5. The convergence was similar for other numbers of cluster.



(a) (b)

Figure 3: (a) Variation trend of log likelihood during iteration; (b) Variation of AIC and BIC as number of cluster varies from 2 to 7;

To determine the optimal number of clusters, we applied two criteria: Akaike Information Criterion(AIC) and Bayesian Information Criterion(BIC), to the results obtained from the EM algorithm. The first term in both formula (Eq.9) stands for penatly and the second term is the likelihood. We tested these two criteria for cluster number ranging from 2 to 7. According to Figure 3(b), both AIC and BIC were minimized when the number of cluster was set as 5. Thus we would focus on the corresponding clustering results for 5 clusters, as summarized in Table 1.

$$\begin{aligned} AIC &= 2p - 2\ln(L) \\ BIC &= p\ln(n) - 2\ln(L) \end{aligned} \tag{9}$$

Cluster	**1**	**2**	**3**	**4**	**5**
Cell Numbers	88	27	376	42	183

Table 1: Clustering result for 5 clusters

4 Signature Genes

4.1 Methodology

To select signature genes of each cells, we applied support vector machine - recursive feature elimination (SVM-RFE).

For SVM-RFE, it contains following steps. 1) Initialization: SVM-RFE first trains an SVM model using all available features. SVM is a supervised learning algorithm used for classification or regression problems, with the core idea of finding the optimal boundary (i.e., maximum margin) between data categories. 2) Computing Feature Weights: After training, the importance of each feature is calculated as the weight. 3) Feature Elimination: Based on the absolute values of weights, the feature with the smallest weight is removed because features with smaller weights contribute less to the model. 4) Iteration: After updating the dataset, a new SVM model is trained, and the above process is repeated. This process is carried out recursively, and the least important feature at each iteration is removed. 5) Termination Condition: This process continues until a preset number of features is reached, or the model performance reaches a specific criterion.

4.2 Result

In our project, we first set up the predictor matrix which contained genes information and the response of cluster number. Then, we built model based on repeated cross-validation and test different sizes of signature features. Based on the accuracy

of the model, we selected all the signature genes of each cluster and listed out the top 3 signature genes referring to the importance ranking. To validate our results, we applied boxplot to draw the distribution of signature genes.

cluster	number of signature genes	top 3 (importance score)
1	100	myl9(0.0139), gstm1(0.0136), tpm1(0.0129)
2	25	smoc1(0.1782), cst3(0.0453), trf(0.0411)
3	150	rgs5(0.0548), kcnj8(0.0349), higd1b(0.0263)
4	50	csn3(0.0219), rn45s(0.0207), c3(0.0191)
5	25	spon1(0.0968), serpinf1(0.0691), olfml3(0.0479)

Table 2: Results of signature genes of each cluster

According to Table 2, each cluster have different number of signature genes. For example, cluster 1 has 100 signature genes while cluster 2 has 25. The top 3 signature genes are also listed in the table. For cluster 1, myl9, gstm1 and tpm1 are signature; for cluster 2, smoc1, cst3 and trf are signature; for cluster 3, rgs5, kcnj8 and higd1b are signature; csn3, rn45s and c3 are signature in cluster 4; spon1, serpinf1 and olfml3 are signature in cluster 5. The importance score represented how much the genes contributed to the differentiation of different clusters.

To valid the genes are signature, we used the boxplots. Figure 4 is an example of our validation process. Other four plots could be checked in Appendix.

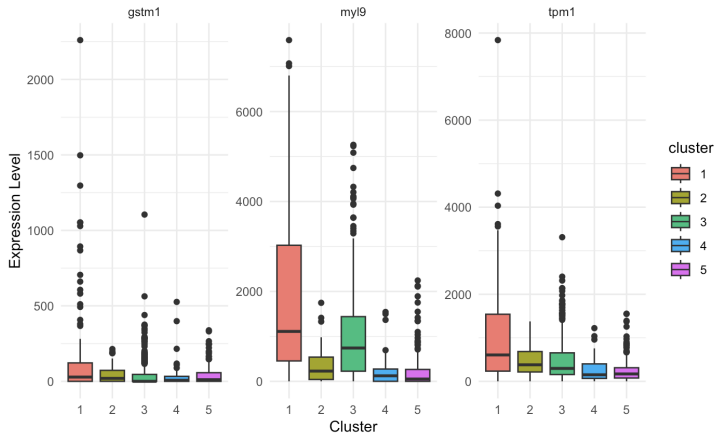


Figure 4: Distribution of experession level of top 3 signature genes in cluster 1 across all clusters

As shown in Figure 4, distribution of the expression level of top 3 signature genes selected in cluster 1 across all clusters are drawn. We could obviously observe that the median values of the expression level of these genes in cluster 1 are much higher than those in other 4 clusters. It indicates these 3 genes are signature for cluster 1.

5 Conclusion and Discussion

In this project, we started with PCA for dimensional deduction. Then we modified Gaussian-mixture model with EM algorithm to cluster the cells. Finally, we conducted SVM-RFE for the selection of signature genes.

Above all, we selected 5 as our optimal cluster number based on AIC and BIC values. As mentioned before, we observed different signature genes type and different number of signature genes in each cluster.

For our project, we had two observations to discuss. Firstly, we found that the cumulative proportion at the 24th principle component selected in our PCA step is only around 40%. The possible reasons might be the zero-inflation and the scale procedure. We noticed that there are lots of 0s in the dataset and it might affect the explanation. Also, when we conducted PCA, we rescaled the dataset, it might also affected since orgin data has significantly different expression level.

Secondly, we noticed that the numbers of cells in each cluster are different. For example, cluster 3 has 376 cells while cluster 2 only have 27 cells. This might be because the model we applied in this project is mostly based on mathematical method. In the future, we can take advantage of the knowledge of genetic to further improve the model.

6 Reference

[1] Huang D, Ma N, Li X, et al. Advances in single-cell RNA sequencing and its applications in cancer research. J Hematol Oncol. 2023;16(1):98. Published 2023 Aug 24. doi:10.1186/s13045-023-01494-6

[2] Greenacre, M., Groenen, P.J.F., Hastie, T. et al. Principal component analysis. Nat Rev Methods Primers 2, 100 (2022). Published 2022 Dec 22. doi:10.1038/s43586-022-00184-w

7 Appendix

7.1 Contributions

Ou Sha: Introduction, PCA

Kindle Zhang: EM algorithm

Mengxiao Luan: EM algorithm

Qin Huang: Signature gene, conclusion and discussion

7.2 Plots

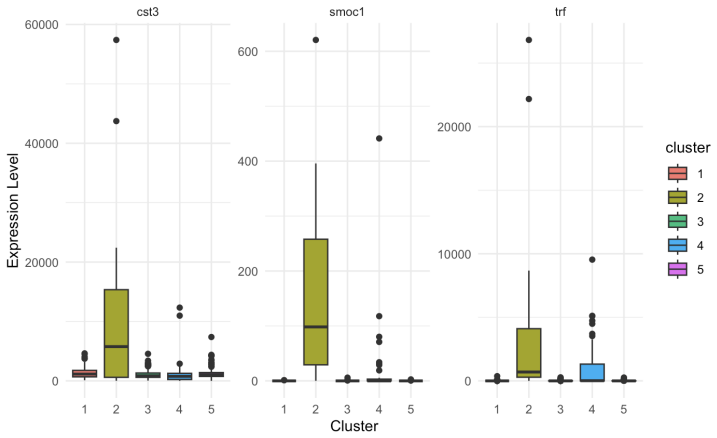


Figure 5: Distribution of experession level of top 3 signature genes in cluster 2 across all clusters

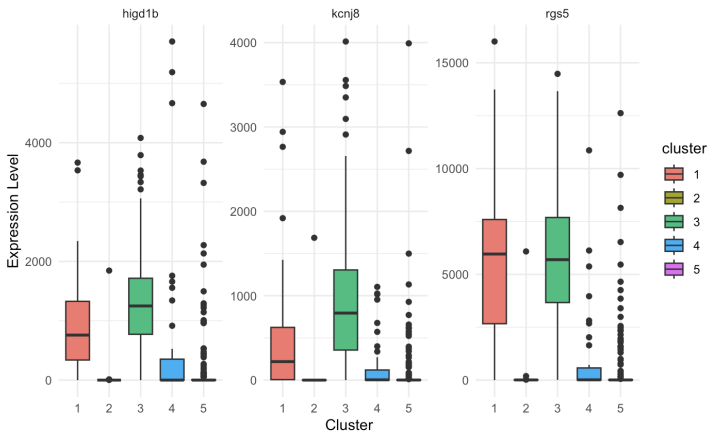


Figure 6: Distribution of experession level of top 3 signature genes in cluster 3 across all clusters

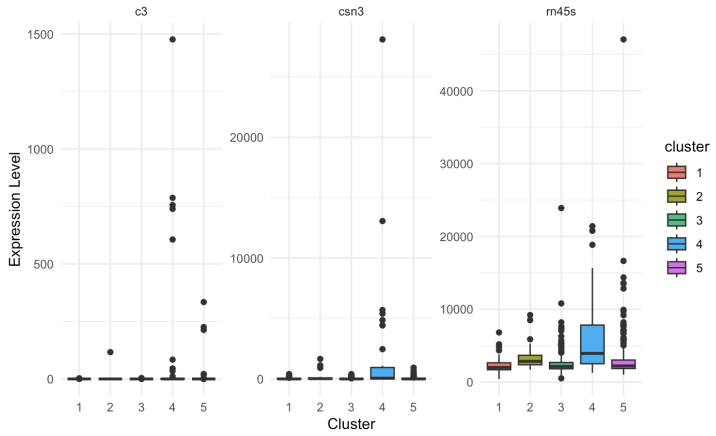


Figure 7: Distribution of experession level of top 3 signature genes in cluster 4 across all clusters

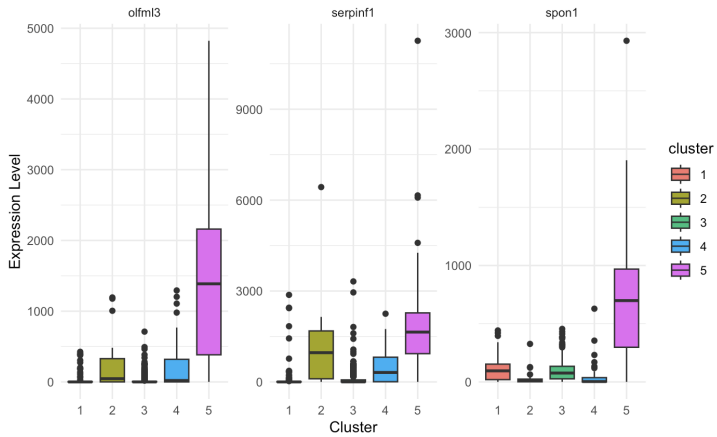


Figure 8: Distribution of experession level of top 3 signature genes in cluster 5 across all clusters