International Core Journal of Engineering Volume 6 Issue 12, 2020
ISSN: 2414-1895 DOI: 10.6919/ICJE.202012_6(12).0061

Using Data Processing Method and Different Models to Predict Cell Types from Cell's Gene Profiles

Jiayi Sun[1], Yiwei Bian[2], Wenzhi Wang[3], Logan Muyu Cheng[4]

[1]Nanjing Jinling High School, Nanjing, Jiangsu 210005, China;

[2]Nanjing Foreign Language School, Nanjing, Jiangsu 210018, China;

[3]The MaxDuffie School Nanjing, Nanjing, Jiangsu 210002, China;

[4]Tsinghua International School, Beijing, Beijing 100084, China.

Abstract

The purpose of this project is to predict cell types from gene expression. The main focus of our project is to use different models to predict single cell types. The first step of this project is to use feature selection to eliminate irrelevant data sets with less value in order to filter out the valuable datasets we can then use. We then selected five cell types from a total of twenty-one cell types. Moreover, we used the method of PCA for dimensional reduction. Furthermore, the four models we used are Naive Bayes, Decision Tree, Random Forest, and Logistic Regression. For the results, all four of our models showed diverse outcomes. With the results from Logistic Regression having the highest accuracy, the results from Random Forest having the best stability, and the results from Naive Bayes and Decision tree having high accuracy for certain cell types. Our work tends to make the cell type prediction process more efficient by training models based on the cell types from the data in gene's profiles than the traditional biological method in the lab.

Keywords

Feature Selection; PCA; Classifications; Confusion Matrix.

1. Introduction

In the article *Competency of different cell models to predict human hepatotoxic drugs* [1], it described the advantages and limitations of hepatic cell-based models for early safety risk assessment during drug development. These models include hepatocytes cultured as monolayer, collagen-sandwich; emerging complex 3D configuration; liver-derived cell lines; stem cell-derived hepatocytes. Similarly, our work is about using different models to predict single cell types. The topic of our work is very significant in the real world, and it can be used in many different real-world scenarios. For instance, in laboratory situations, when scientists obtain genes and they wish to figure out the cell types, they would be able to use the methods regarding modeling instead of the conventional biological method, which is way less efficient and a lot more time consuming. Therefore, it is proven that using models can be a very efficient way when it comes to processing data and finding results. In the work, we used the methods of feature selection and PCA together with the four models of Naive Bayes, Decision Tree, Random Forest, and Logistic Regression. We also used confusion matrices to display the accuracy of the results visually. Feature selection helps us find valuable data sets and eliminate irrelevant data sets with less value. After completing the work gf, the results suggest that feature selection before PCA is very necessary because it improves the precision rate to varying degrees. The study indicates that, by using variance to select feature, the best selection occurs when around 1600

ISSN: 2414-1895 DOI: 10.6919/ICJE.202012_6(12).0061

features are kept. Moreover, in complex models, Random Forest and Logistic Regression have higher precision rates than Naive Bayes, and Decision Trees.

2. Methods

2.1 Datasets used in our analysis

We use the same data source as the data used in A web server for comparative analysis of single-cell RNA-seq data [2], which are from GEO and Array Express. The dataset contains a total of 24244 single cells, including 21389 in train's and 2855 in test's. Each cell has 20499 genes (features), which are in RPKM unit. The labels are the types of each cell. Since the RPKM (Reads Per Kilobase per Million mapped reads) values are already normalized so we did no additional normalization. Some of the cell-types are held out because the data is from different experiments, and therefore not all cell types are in both train and test. There are 21 test cell types and 46 train cell types. This constraint will also affect our choice of cell types.

2.2 Missing data

We noticed missing values in the data (represented by 0) and assumed it will cause an effect on the results. But we want to know how the models will respond to the data so we did feature selection at first. The performance of models wasn't very good. Thus, we decided to try to eliminate the features with many missing values. Nevertheless, when we sorted and displayed the number of missing data in each feature of each cell type, we could not find the limit for feature selection. Therefore, we used the average value of each feature of each cell type to fill missing values and it did show better precision.

2.3 Feature selection

For the number of cell types, we decided to use 5 cell types which is not too few nor too complicated to be for the models. Initially, we chose 5 cells randomly in the train data. (UBERON:0002435 striatum, UBERON:0001851 cortex, CL:0002322 embryonic stem cell, UBERON:0001003 skin epidermis and UBERON:0001898 hypothalamus) However, when we tried to correspond them to the same types in the test data, we found not all cell types have corresponding data in test and train data. Therefore, we switched two of the cell types (UBERON:0002435 striatum and UBERON:0001898 hypothalamus) with two new cell types (UBERON:0000044 dorsal root ganglion and CL:0000037 hematopoietic stem cell). Then we used variance to filter the features which is more suitable for the PCA and the models. In our feature selection, we chose the features whose variance are bigger than 10, 20, 30, 50, 100, 150, 200 and 300, and found that variance larger than 300 usually has the best overall performance. However, when we tested using different models, random forest shows better performance when using the data which is filtered by the variance of 50, so we decided to use the data filtered by 50, 100 and 300.

2.4 PCA

PCA (Principle Components Analysis) is an efficient method in reducing dimensions of data in unsupervised-learning. In this paper we use the SVD method. It mainly reduces dimensions by rotating the coordinates. To do so, we first see our sample sets as matrix X. Then we calculate the co-variance matrix C of the sample set. The elements in this matrix's diagonal shows the variance in each dimension.

$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{bmatrix}$$

$$\operatorname{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{n-1}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202012_6(12).0061

$$C = \begin{pmatrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) & \mathrm{cov}(x,z) \\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) & \mathrm{cov}(y,z) \\ \mathrm{cov}(z,x) & \mathrm{cov}(z,y) & \mathrm{cov}(z,z) \end{pmatrix}$$

Due to the nature that covariance matrix C is a real symmetric matrix. We can get n linearly independent non-zero eigenvectors and the feature matrix E formed by the feature matrix satisfy that:

$$E^T C E = \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{bmatrix}$$

Then we suppose the formula for dimensional reduction matrix is Z=XU where Z denotes as:

$$Z = \begin{bmatrix} z_1^{(1)} & z_2^{(1)} & \cdots & z_n^{(1)} \\ z_1^{(2)} & z_2^{(2)} & \cdots & z_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{(m)} & z_2^{(m)} & \cdots & z_n^{(m)} \end{bmatrix}$$

Since each dimensional feature is linearly non-relevant, the covariance is zero for each feature (cov(X,Y)=0). And the covariance matrix D can denote as:

$$D = \frac{1}{m} Z^T Z = \begin{bmatrix} \frac{1}{m}\sum_{i=1}^{m}(z_1^{(i)})^2 & 0 & \cdots & 0 \\ 0 & \frac{1}{m}\sum_{i=1}^{m}(z_2^{(i)})^2 & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & & \frac{1}{m}\sum_{i=1}^{m}(z_n^{(i)})^2 \end{bmatrix}$$

By substituting Z we can get:

$$\begin{aligned} D &= \frac{1}{m} Z^T Z \\ &= \frac{1}{m}(XU)^T (XU) \\ &= \frac{1}{m} U^T X^T X U \\ &= U^T (\frac{1}{m} X^T X) U \\ &= U^T C U \end{aligned}$$

And by Z=XU, we can get the dimensionally-reduced matrix Z.This is the process of PCA. [3,4]

In our study, we tried different components in the PCA in getting the most precise results. Therefore, the PCA components for each model is different.

2.5 The models chosen for the anaylsis

Note: Due to time limitation, we did no adjustment to Naïve Bayes Classifier, Logistic-regression Classifier and Decision Tree Classifier.

2.5.1 Naïve Bayes Classifier

We choose this model because it is able to proceed multi-class classification and not very sensitive to missing data. The Naïve Bayes classifier has the full classification above rule as below:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202012_6(12).0061

$$\hat{y} = \arg\ max_v p(y = v \mid X)$$
$$= \arg\ max_v \frac{p(X \mid y = v)p(y = v)}{p(X)}$$
$$= \arg\ max_v \prod_j p_j(x^j \mid y = v)p(y = v)$$

This is based on the assumption that the attributes are conditionally independent:

$$p(X \mid y) = \prod_j p_j(x^j \mid y)$$

Where X is denoted as:

$$X = \begin{bmatrix} x^1 \\ \vdots \\ x^n \end{bmatrix}$$

However, due to this assumption, the final results may be biased because the genes may not be totally independent.

2.5.2 Logistic Regression Classifier

The logistic regression uses the sigmoid function (shown below) to make classifications. In our analysis, we use the logistic-regression classifier because it is efficient and can produce precise results using the default parameters.

$$p(y = 0 \mid X; \theta) = g(\mathrm{w}^{\mathrm{T}}X) = \frac{1}{1 + e^{\mathrm{w}^{\mathrm{T}}x}}$$
$$p(y = 1 \mid X; \theta) = 1 - g(\mathrm{w}^{\mathrm{T}}X) = \frac{e^{\mathrm{w}^{\mathrm{T}}x}}{1 + e^{\mathrm{w}^{\mathrm{T}}X}}$$

2.5.3 Decision Tree Classifier

Decision tree classifier uses a tree-like model which has internal nodes which corresponds to attributes, leaves corresponding to outcome and edges denote assignment to make classification. We choose this classifier because it is suitable for high dimensional data and does not require much calculation. However, it has limited capability in dealing with missing values. We can use information gain to calculate the performance of the decision tree. Information gain is calculated by the difference of entropy in each node. Entropy has the formula of:

$$H(X) = \sum_i -p(X = i)\log_2 p(X = i)$$

2.5.4 Random Forest Classifier

Random forest classifier is a stable classifier which use decision tree from each bootstrap and use model averaging to generate the best result. This classifier can also reduce the risk of over-fitting. The model is shown in Figure 1, which is retrieve from professor Ziv Bar Joseph "Introduction to Machine Learning Ensemble of trees: Begging and Random Forest"slide 5.

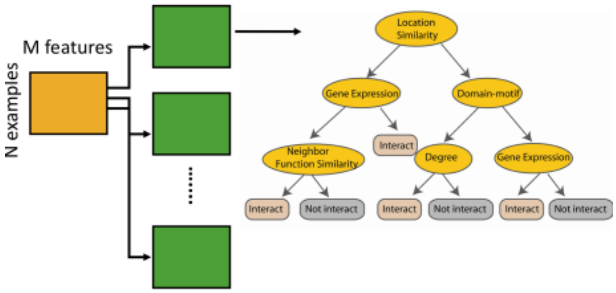

Figure 1: Random Forest Model

ISSN: 2414-1895 DOI: 10.6919/ICJE.202012_6(12).0061

In our analysis, we set the parameter of the maximum number of features in our python program to the square root of the component number in the PCA. We tested several values around the value. For example, one of our models used the component number of 100, so we tested the number of maximum features from 5 to 15.

2.6 Results Presentation

We constructed confusion matrix to provide a visual way to present the accuracy of the results. Since we have 5 cell types, we have confusion matrix of 5*5 which has real cell as horizontal axis and predict cell in the vertical axis. From the matrix we can easily see how many real cells of type x is predicted as cell type y. The depth of colors can help us more in seeing the numbers. (the deeper the color, the more the cell is predicted) Matrix are constructed for each of the 4 models we used.

For the final scoring, we tried kappa (which is based on the confusion matrix) and weighted scores other than the precision generated by the model in python. Kappa score is a way to assess consistency statistically which has a range of -1 to 1. It is calculated as the formula shown below:

$$\mathrm{k} = \frac{P_O - P_\mathrm{e}}{1 - P_\mathrm{e}}$$

where PO stands for overall classification accuracy and Pe stands for SUM (the number of real sample of type i *predicted number of type i)/total sample number^2. The grades of kappa are very low(0-0.2), fair(0.2-0.4), average(0.4-0.6), high(0.6-0.8), almost consistent(0.8-1.0). [5]

Though the kappa score can show different scores, it barely shows better scoring than the weighted scores. Weighted score, on the other hand, does show better performance in scoring since it takes account the percentage of each cell in the data.

3. Result

The classification of cell types is arduous work, which involves the identification of ten thousand of genes, and the modeling of the proper features to represent the cells, as well as improving the efficiency and accuracy of the models.

In our project, we successively tried 4 types of classifiers (Naïve Bayes, Decision Tree, Random Forest, and Logistic Regression) for the classification of cell types. Instead of using the all 21 cell types for modeling, we simply selected 5 of them (UBERON:0001851 cortex, CL:0002322 embryonic stem cell, UBERON:0001003skin epidermis, UBERON:0000044 dorsal root ganglion, and CL:0000037hematopoietic stem cell) which are the cell types with both their percentage in train data and the percentage in test date the highest among all cell types. We did feature selection for sample variance >50, >100, and >300 to get the corresponding feature numbers 2427, 1688, 913. Then we used PCA for dimensionality reduction. All the four classifiers showed diverse results, in which the results from Logistic Regression showed the highest accuracy, the results from Random Forest showed the best stability, and the results fromNaïve Bayes and Decision tree showed high accuracy for certain cell types (shown in Table 1).

3.1 Feature Selection

In the process of feature selection, we first tried it on the full sample and found the result is not good, because some of the data we process have no expression or expressed zero on certain genes. Thus, we replace those null values with the mean. Then, we tried various variances selection standard. Among the ten sets of feature selection data with variances>10, >20, >30, >50, >100, >150, >200, >300, the input with variance >50, >100, and >300 came up to be suitable for the input datasets. Thus we determined the corresponding feature numbers 2427, 1688, 913 (shown in Table 2).

3.2 PCA

In the section of data processing, we also did PCA-reduction to reduce the dimensionality of the features. We use components in the range between 300 and 5 in order to reach the highest accuracy for different classifiers.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202012_6(12).0061

After replacing the null values in the feature selection, the results of PCA would be more or less disperse than the real value. Therefore the result from the classifiers would also be affected and becomes inaccurate comparing to the real values.

3.3 Classifiers

3.3.1 Naïve Bayes

Table 1: Highest precision score of each model

Model type	Naive Byes	Decision Tree	Random Forest	Logistic Regression
Precision Score (weighted)	0.7548	0.7218	0.8211	0.8973

Table 2: Final Feature Selection Result

Variance	>50	>100	>300
Feature Numbers	2427	1688	913

Cell type	Percentage in train data	Percentage in test data
UBERON:0001851 cortex	0.0773	0.0932
CL:0002322 embryonic stem cell	0.0738	0.1254
UBERON:0001003 skin epidermis	0.0732	0.2375
UBERON:0000044 dorsal root ganglion	0.0473	0.0431
CL:0000037 hematopoietic stem cell	0.0302	0.0567

For NB, we initially reached a fair outcome of a precision rate of approximately 75% in all three outputs of feature selection. The changes in the PCA component do not result in a great change in the precision rate of feature selection that var >50, with a maximum changing rate of 0.4, but result in a drastic change in the precision rate of feature selection that var >300, with a maximum changing rate of 0.33 (shown in Table 3).

3.3.2 Decision Tree

DT was proved not to be an adequate method for our dataset, which showed its highest accuracy just crosses the edge of 70%. We also found that while the number of PCA components decreases, the general precision rate increases. We assumed that in low dimensional condition, the mapping becomes sweeping that it is easier for DT to classify each cell types at the nodes. In addition, although DT did not make a great classification of the cell types, it provided a reliable foundation for Random forest, which we employed later in the following part. The DT's result is shown in Table 4.

3.3.3 Random Forest

For RF, we fitted different parameters in the model to approach the optimal setting. The result of RF depended largely on the result of the DT. In all the situation we attempted, the one with the highest accuracy reached 0.82. In contrast to the DT, RF showed, in general, a higher accuracy when the PCA component increase. We supposed that high dimensionality might not seriously influence the effect of RF, but the lost information in the process of dimensionality reduction may cause a negative effect on the accuracy of RF. Meanwhile, the fluctuation of the output of RF is less than 12% in all three output of feature selection, and less than 5% in the output of feature selection that var >50, and the range of the PCA component is also the widest in all 4 classifiers we took, so RF successfully showed its stability in working in our dataset. The result of RF is shown in Table 5.

3.3.4 Logistic Regression

Though we have run LR in the end, it showed an accuracy amazingly high of nearly 90% in the output of feature selection that var >50 and the output of feature selection that var >100. As shown in the

ISSN: 2414-1895 DOI: 10.6919/ICJE.202012_6(12).0061

table, it is easy to see that when the PCA component is 47, the result reached an optimal configuration in the dataset we chose, which we considered it as where the model converges. The result of LR is shown in Table 6.

3.4 Confusion Matrix

After testing the 4 types of classifiers, we also made the corresponding confusion matrixes to make a visualized contrast between them all. Because the colors of each square in the matrix represent the proportion of cells classified indifferent catalogues, and the clinodiagonal represents the correct classification, we can compare the differences in the color of the squares to get a comparison of the accuracy of different methods. In all four classifiers, we can see that Logistic Regression has made the least errors and maximum promotion in the correct catalogue, which showed its best precision rate in all four classifiers in the chosen dataset. We could also see that all 4 classifiers showed a higher accuracy on certain cell types, which might be useful information for future studies. The confusion matrixes are shown in Figure 1.

Table 3: Precision Score of Naive Bayes

PCA	var >50	var>100	var>300
110	0.701623	0.709032	0.715101
100	0.710496	0.712317	0.403258
80	0.719159	0.727427	0.738467
50	0.730384	0.742553	0.514648
48	0.746342	0.728457	0.730904
30	0.721637	0.754785	0.685190
20	0.709051	0.701326	0.733283
10	0.717615	0.727917	0.727236

Table 4: Precision score of Decision Tree

PCA	var >50	var >100	var>300
110	0.705511	0.509510	0.427351
100	0.547533	0.565216	0.383679
80	0.556052	0.502083	0.542676
50	0.567821	0.612238	0.623814
30	0.625946	0.644418	0.590588
20	0.593242	0.641802	0.670820
10	0.701792	0.683828	0.691013
5	0.667342	0.721814	0.660019

Table 5: Precision Rate of Random Forest

Feature Selection	var >50		var >100			var >300	
PCA\n_feature	5	10	5	8	10	5	10
200	0.821054	0.803541	0.800413	0.798662	0.791721	0.798613	0.801256
105	0.793117	0.788130	0.795394	0.801448	0.781144	0.789421	0.792732
100	0.773651	0.787519	0.790578	0.780830	0.780381	0.799095	0.766305
80	0.776467	0.778148	0.782013	0.767330	0.774066	0.786501	0.779678
50	0.790564	0.784052	0.786897	0.782455	0.777060	0.778383	0.778706
30	0.808718	0.808610	0.794205	0.784006	0.778218	0.795133	0.794090
20	0.765571	0.732746	0.725864	0.713258	0.702967	0.782377	0.768188
10	0.724812	0.710945	0.716911	0.742333	0.740904	0.741492	0.708627

ISSN: 2414-1895 DOI: 10.6919/ICJE.202012_6(12).0061

Table 6: Precision Rate of Logistic Regression

PCA	var >50	var >100	var >300
110	0.879554	0.871803	0.868476
100	0.875283	0.875804	0.866914
80	0.887470	0.881379	0.874984
50	0.890802	0.886627	0.887658
47	0.897282	0.891003	0.886498
30	0.841266	0.844209	0.821406
20	0.872524	0.868354	0.845601
10	0.832594	0.789804	0.799577

4. Discussion

We predicted the cell types from their gene expression profiles by using four models, which are Naive Bayes, Decision Tree, Random Forest, and Logistic Regression.

Five types of cells were selected to develop the models from the total twenty-one cells we get. RF model and LR model have relatively higher precision scores than NB model and DT model. All the four models predict cell types well and get moderate precision score. The Table 1 shows the highest precision score we get from these four models. Moreover, before the modeling, we did data pre-processing. First, we fill the null value by the mean of each gene in each cell types. Then, we did feature section base on the variance. Three variance standards were chosen at last for feature selection. We select features' variances that are larger than 50, 100, and 300, and built the models base on this three data sets. After feature selection PCA (Principle Component Analysis) was used to reduce dimensionality of the data.

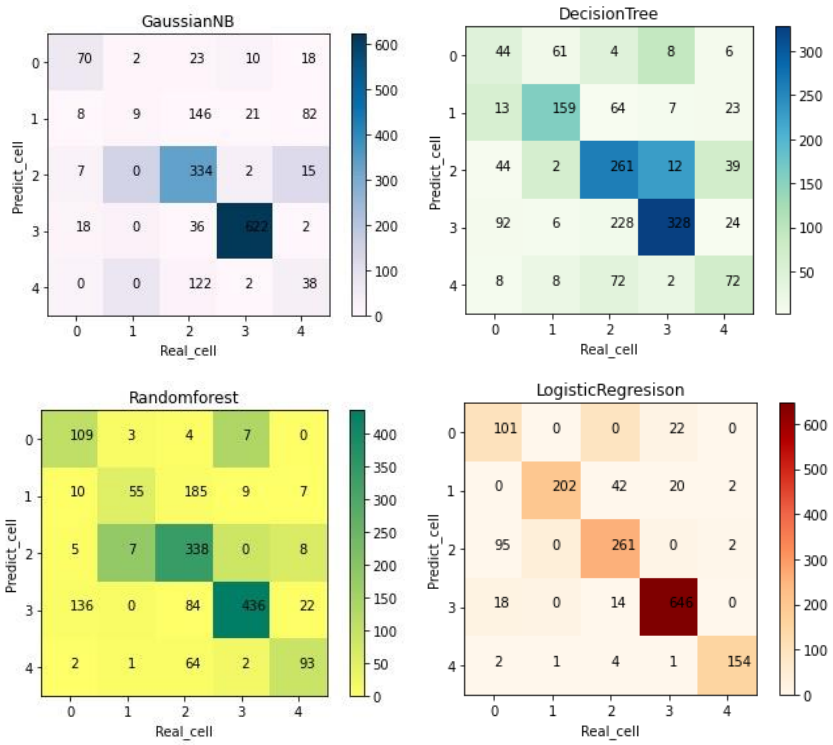

Figure 2. Confusion Matrixes of the four models

ISSN: 2414-1895 DOI: 10.6919/ICJE.202012_6(12).0061

Among the four models we use, LR model has the best outcome. The precision score of the LR model is nearly 90 percent, which predict better compare to the 70 or 80 precent precision score of other three models. Despite the RF Model, other three models did not revise the default parameters. Therefore, the high precision score of the LR model mainly attributed to the model structure itself and the data processing before the modeling. The highest precision score of the LR model was get when the component of PCA is 47, and the variance selection standard are 50 or 100. For the RF Model, its highest precision scores were gotten when the PCA component are 200 and 105, and max_feature of RF model are 5 and 8, which diverge from the previous belief that the max_feature set as the square root of the component number best [6]. Another two models, NB and DT, they are simper compare to the two that discussed above; thus, it is reasonable that the precision scores of these simple models are lower.

Despite the difference of predicting accuracy between each model, the predicting accuracies between each cell types are also different. From the Confusion Matrix that shown in Figure 2, cell type 3 (UBERON:0001003 skin epidermis) has the highest prediction rate, and Cell type 0 (UBERON:0000044 dorsal root ganglion) has relatively the lowest. The deeper the color means the greater number of cells are predicted in that block.

The biggest and the most important part of this project is data processing. There are originally 20499 features (genes) in each cell, but a large amount of them are zero. These genes are express in the RPKM (Reads Per Kilobase per Million mapped reads) unit, which processing the gene length through normalization [7], which means zero of the RPKM stand for missing data or not existed genes. The equation for calculating RPKM is shown in Figure 3.

However, the problem that exist in our data processing process is that we replace the zero value of RPKM by the means of each genes, which may not be reasonable biologically, because not all the zero means the experiment failed to get the gene's information. Although there may have missing data, some genes actually do not exist even they are in the same cell type. Therefore, by simply filled all the missing data with mean modified the original data, and somehow make the result unreal. Moreover, the PCA reduction is not effective and successful as we thought and it seems not fit with the previous belief that larger the component number, higher the precision score [8]. Only in the Random Forest model, the high precision score occurs when the component number increase. As a result, more component and more information do not help most of our model increase accuracy. It indicates that for input information not the more the better as some information may be useless and unnecessary, and even may effective the model negatively.

$$RPKM\ \ of\ a\ gene = \frac{Number\ of\ reads\ mapped\ to\ a\ gene \times 10^3 \times 10^6}{Total\ number\ of\ mapped\ reads\ from\ given\ library\ \times gene\ length\ in\ bp}$$

Figure 3: Equation of RPKM

5. Conclusion

In this paper, we use data processing methods and develop four models that is widely used by Machine learning, for predicting the cell types through their genes' expressions. We reach the following conclusion that the best precision score is around 90 percent by using the Logistic Regression Model. When the PCA component increases, not all the model increases their predicting rate.

While the results are courting, there still have many things haven't done or discovered in the project, so we have some future work to explore. These include finding a better method to do data pre-processing while not using the mean value, figuring out more efficient feature selection method, studying the relationship between PCA component number and model effectiveness, developing model of different cell types, and discovering the importance of different genes in each cell. The most

ISSN: 2414-1895 DOI: 10.6919/ICJE.202012_6(12).0061

challenging one is to find the importance of each genes, because the number of genes are large and for different cell types, there major genes are different. In order to figure out most of them, large amount of calculation will be down.

References

[1] Gómez-Lechón, M. J., Tolosa, L., Conde, I., & Donato, M. T. (2014). Competency of different cell models to predict human hepatotoxic drugs. Expert opinion on drug metabolism & toxicology, 10(11), 1553–1568. https:// doi.org/10.1517/17425255.2014.967680

[2] Alavi, A., Ruffalo, M., Parvangada, A. et al. (2018) A web server for comparative analysis of single-cell RNA-seq data. Nat Commun 9, 4768. https://doi.org/10.1038/s41467-018-07165-2

[3] CSDN. (2019) Python implements PCA. blog.csdn.net/weixin_42051109/article/details/89416727.

[4] CSDN. (2019) PCA basic principles and principle derivation + PCA calculation steps explanation + PCA examples show the mathematical solution process. blog.csdn.net/u012421852/article/details/80458340.

[5] CSDN. (2019) Summary of evaluation indicators for two-class and multi-class problems. blog. csdn. net/ wf592523813/article/details/95202448.

[6] JianShu. (2019)“Random Forest Parameter Description.” www.jianshu.com/p/b07f552564bc.

[7] Biostar. Gene Expression Units Explained: RPM, RPKM, FPKM and TPM.Biostar, www. biostars. org/ p/273537/.

[8] JianShu. (2019) “PCA(Principal Component Analysis).” www.jianshu.com/p/39d22980dd61.