# BIM3008-Assignment3 Report

Junyang Deng (120090791)

December 4, 2022

## 1 Introduction

Breast cancer is the most common type of cancer for females. One important task for researchers to improve the survival rate of BRCA patients is to identify the cancer stage and apply different treatment strategies. We can train a model to classify cancer stages using RNA-seq data from patient samples. In this project, I used support vector machine (SVM), random forest (RF), and k-nearest neighbor (KNN) to classify RNA-seq data. Then, 5-fold cross-validation is used to select the best hyperparameters for the model. Finally, we used the best model to predict the cancer stage of the test set.

## 2 Procedure

### 2.1 Data Retrieval

Data required for this project is retrieved from the UCSC Xena database. We used the FPKM normalized gene expression RNA-seq data and phenotype labels from the BRCA dataset.

### 2.2 Data Preprocessing

Original reads data has 60483 rows and 1218 columns, each row is a gene and each column is a sample. The corresponding phenotype label table has 1284 rows and 140 columns. Each row represents a sample and each column represents a phenotype label. Only sample IDs and cancer stage diagnoses are used in this project, so the rest of columns are removed. Among the samples, 2 and 12 samples are removed because they have no labels and no diagnoses, respectively. Then, to make the classification task more realistic, we combined the substages of cancer. Originally, there are 13 diagnoses labels (including `not reported`). After combinding the substages, only 5 labels remain (See Table 1 and 2).

Table 1: Original cancer stage labels

| Stage | Number | Stage | Number |
|---|---|---|---|
| stage iia | 415 | stage iv | 22 |
| stage iib | 310 | stage x | 13 |
| stage iiia | 178 | not reported | 12 |
| stage i | 114 | stage ib | 7 |
| stage ia | 94 | stage ii | 6 |
| stage iiic | 77 | stage iii | 2 |
| stage iiib | 32 | | |

Table 2: Cancer stage labels after processing

| Stage label | Number |
|---|---|
| stage i | 215 |
| stage ii | 731 |
| stage iii | 289 |
| stage iv | 22 |
| stage x | 13 |

Two dataframes needed to be merged before further analysis. Here, the `merge` function in pandas is used. Before merging, we first transposed the expression dataframe and changed the names of the `ID` columns in both dataframes into `sample_ID`. The merged data frame has 1204 rows and 60845 columns.

After merging two datafremaes, samples are divided into train and test sets using the `train_test_split` function in `sklearn`. 30% of data are randomly assign to test set.

### 2.3 Model selection and cross-validation

Three algorithms, support vector machine (SVM), random forest (RF) and k-nearest-neighbor (KNN) are used to solve this problem. All models used in this study are from `sklearn` (as shown in Table 3). Baseline performance for all three models are shown in Figure 1-3.After obtaining the baseline performance, we conduct cross-validation to find the best hyperparameters. Class `sklearn.model_selection.GridSearchCV` is to perform this process. Results for cross-validation hyperparameter tuning can be access through `grid.best_params_`. The results are listed in Table 4.

Table 3: Algorithms used in this study

| Algorithm | Class in sklearn | Key Parameters |
|---|---|---|
| SVM | `sklearn.svm.svc` | gamma, kernel |
| RF | `sklearn.ensemble.RandomForestClassifier` | n_estimators, max_depth, min_samples_split, ... |
| KNN | `sklearn.neighbors.KNeighborsClassifier` | n_neighbors |

```
Accuracy of SVM: 0.511049723756906
Confusion matrix:
 [[ 19  45   8   0   0]
 [ 19 146  35   0   0]
 [ 11  51  20   0   0]
 [  1   1   3   0   0]
 [  0   3   0   0   0]]
```

Figure 1

```
Accuracy of RF: 0.5524861878453039
Confusion matrix:
 [[  0  72   0   0   0]
 [  0 199   1   0   0]
 [  0  81   1   0   0]
 [  0   5   0   0   0]
 [  0   3   0   0   0]]
```

Figure 2

```
Accuracy of KNN: 0.47790055248618785
Confusion matrix:
 [[  3  62   7   0   0]
 [ 18 155  27   0   0]
 [  6  61  15   0   0]
 [  0   4   1   0   0]
 [  0   3   0   0   0]]
```

Figure 3

```
# settings for SVM
svm_params = {
    'kernel':['linear', 'sigmoid', 'poly', 'rbf'],
    'gamma':['auto', 'scale'],
    }
grid1 = GridSearchCV(SVC(), svm_params, cv=3, verbose=3, n_jobs=-1)
# settings for RF
rf_params = {'bootstrap': [True, False],
             'max_depth': [5, 15, 50, 100, None],
             'max_features': [20, 50, 80],
             'min_samples_leaf': [1, 2, 4],
             'min_samples_split': [2, 5, 10],
             'n_estimators': [10, 100, 200]}
grid2 = GridSearchCV(RandomForestClassifier(), rf_params, cv=3, verbose=3, n_jobs=-1)
knn_params = {'n_neighbors':[5, 10, 20, 50, 100]}
# settings for KNN
grid3 = GridSearchCV(KNeighborsClassifier(), knn_params, cv=3, verbose=3, n_jobs=-1)
```

Table 4: Best parameters

| Algorithm | Best parameters | Performance |
|---|---|---|
| SVM | 'gamma': 'auto', 'kernel': 'sigmoid' | 0.5524 |
| RF | 'bootstrap': False, 'max_depth': 50, 'max_features': 50, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 100 | 0.5414 |
| KNN | 'n_neighbors': 50 | 0.5497 |

## 2.4 Feature selection

Apparently, 64000+ features are definitely too much, and most of them must be noise. To improve model performance, we can do feature selection to pick out important genes for the classification task. Two different approaches have been used in this study. One is to apply traditional feature selection methods. Another approach is to select features based on prior biology knowledge. These two approaches are elaborated in the following subsections.

### 2.4.1 Select features based on correlation

Genes with have constant expression level among different samples not only could not benefit stage classification, but also create noise for the task. Therefore, the first feature selection method we used is to set variance threshold for genes. The `sklearn.feature_selection.VarianceThreshold` transformer is used to complete this job. We set the variance threshold to be 0.5. After transformation, 5685 features (genes) were left.

The second method is to choose features that are correlated with labels. Correlation is evaluated by chi-square. This idea is realized using the `sklearn.feature_selection.SelectKBest` selector and the `sklearn.feature_selection.chi2` method.

We selected 566 features with highest chi-square scores. The number of features selected were decided by a p-value threshold (<0.01).

#### 2.4.2 Select features based on prior biology knowledge

Since we are predicting breast cancer stages based on gene expression profile, knowing which gene is important for stage classification will be very helpful. 109 genes were selected from literature [1], including some growth factors like PDGF and growth-related pathways (MAPK, PI3K-Akt, ...) (The full list of selected genes are in the `list_of_genes2.txt` file).

### 2.5 Result

Model performance (evaluate by accuracy) using three models and is calculated on original features and three sets of newly selected features (Table 5). Sadly, any of feature selection methods cannot yield significantly better result. For a 5-class classification problem, an accuracy of 50% seems acceptable. However, the best accuracy (0.5552) is obtained when a model classified almost all samples into the stage II (by random forest classifier, after feature selection). The same situation persists even when the problem was changed into a binary classification problem. This means that current models don't have the ability to distinguish stage II and other stages. More information, such as histological image data and proteomic data, should be intergrated to make a better result.

Table 5: Performance of three models on different features

| Model | Original data k=60428 | Variance threshold k=5675 | Chi-square k=566 | Prior knowledge k=109 |
|---|---|---|---|---|
| SVM | 0.5524 (33.5s) | 0.5497 (0.4s) | 0.5497 (0.2s) | 0.5552 (0.2s) |
| RF | 0.5525 (13.9s) | 0.5497 (0.2s) | 0.5525 (1.1s) | 0.5552 (0.6s) |
| KNN | 0.4779 (1.8s) | 0.5234 (0.6s) | 0.5328 (0.8) | 0.5497 (0.4s) |

## 3 Workflow of the codes

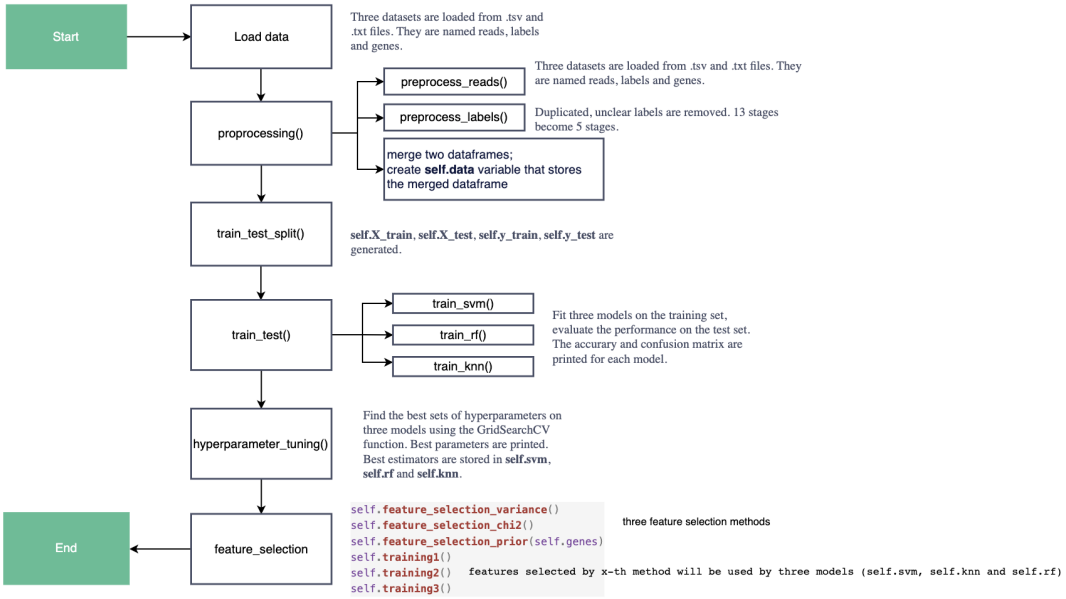


Figure 4

## References

[1] Aatish Thennavan, Francisco Beca, Youli Xia, Susana Garcia-Recio, Kimberly Allison, Laura C Collins, M Tse Gary, Yunn-Yi Chen, Stuart J Schnitt, Katherine A Hoadley, et al. Molecular analysis of tcga breast cancer histologic types. *Cell Genomics*, 1(3):100067, 2021.