Classifier of breast cancer subtypes final report

Wanli Wang
2020/05/08

1.Introduction

1.1 Background

Breast cancer has different intrinsic subtypes, which implicate distinct morphology and clinical features. The breast cancer type can be roughly classified into Luminal A, Luminal B, Basal-like, which is also called Triple-negative, HER2-enriched and Normal-like. Luminal A tends to grow slowly and has the best prognosis among the group. For Luminal B, even it shares some similarities with Luminal A in name, it has different mechanism which promotes its faster growth and worse prognosis. Basal-like breast cancer shows higher frequency among younger women with BRCA1 gene mutation, and HER2-enriched patients are often successfully treated with targeted therapies aimed at the HER2 protein. Finally, for normal-like breast cancer, while similar to Luminal A, it has slightly worse prognosis than Luminal A. So, the precise classification of breast cancer subtypes is very essential in the prognosis, choice of effective treatment, also the control of metastasis.

The advent of microarrays has provided a new platform to decipher the heterogeneity within the breast cancer. The microarrays, a driving technique behind gene expression studies in the early years, can provide assays of gene expression profiling on breast cancers. And these profiles can lead to an informative classification of the disease. In 2009, Parker et al. has taken the expression of 50 genes to build a classifier among a group of samples. And they successfully classify them into 5 subtypes. That classifier is commonly known as PAM50 classifier.

However, recently, RNA sequencing technique has largely replaced microarray as the main paradigm for gene expression profiling, and the previous classifiers which are trained on microarray data have poor performance in the classification of RNA-seq data. The Caner Genome Atlas (TCGA) preserves many molecular characteristics of breast cancer samples also normal tissues, and the gene expression inside is profiled using RNA-seq. This dataset drives us to figure out whether we can develop classifiers for the RNA-seq data, and how well we can classify them into the 5 intrinsic subtypes like microarray data.

1.2 Dataset and problem

The dataset we are going to use is from TCGA, which has 1200 samples from both breast cancer patients and normal tissues. In these 1200 samples, 600 of them has also been profiled using microarray, so we have the “ground truth” subtype information from PAM50 classifier for them. However, since PAM50 classifier is not trained on RNA-seq data, we cannot make predictions on the other samples with it directly. Thus, we are going to develop new classifier and train it on part of the samples, and then make predictions for the group which does not have class information.

In general, we have 1208 samples, in which the subtype of 626 samples have been previously assigned by PAM50 classifier, and 582 samples are unknown. We are going to build the classifier

on the 626 samples, and once the classifier is well trained, we then will move on to do predictions on the other 582 samples.

2. Methods

2.1 Data normalization

In total, we have 20466 genes within the dataset. First, we extract the genes which show no expression among all the samples, and 20187 genes are left within the dataset after this. Next, since it is a raw dataset with only reads, we apply the logarithm transformation and median centering to every expression value. After these normalized steps, we narrow the scale among the genes, also it can help to provide more meaningful analysis in feature selection, and better presentation in heatmaps.

2.2 Feature selection

For this part, we are going to select a set of genes that we can take as features in later classification from the whole 20K gene set. Basically, we pick the genes that show great significance among the 5 breast cancer subtypes. So first we split the whole dataset into the known set and unknown set. For the known set, with class information, this is the data we are going to use for both feature selection and the training and testing of the model. In general, within the known set, we use t test for every group compared with all the other groups, also between each pair of classes.

TO begin with, we take all the genes that achieved p-value smaller than 1e-8 in the one-vs-rest t test analysis, in which 69 genes are picked out. The heatmap of the samples in the known set characterized by these 69 genes are shown in Figure 1. In this heatmap, we can see that the purple, red and blue zones are well clustered, but the green and yellow zones are still mixed together. This is quite normal, since green and yellow zones represent Luminal A and Luminal B subtypes, which share more similarities in gene expression patterns compared to other classes. So, for the next step, we also include the genes which show significant differences in the eachpair t-test for Luminal A and Luminal B. With these new features included, the heatmap change to Figure2. However, in Figure2, even though Luminal A and Luminal B are better separated, the blue and yellow zones are mixed together, which represent HER2-enriched and Luminal B. Again, we add the features show significance in each-pair t test for HER2-enriched and Luminal B, the new heatmap is shown in Figure3. Now the purple and green zones are clustered together. Still, this time, we add the features between normal-like and Luminal A into the features set. And the heatmap, which is

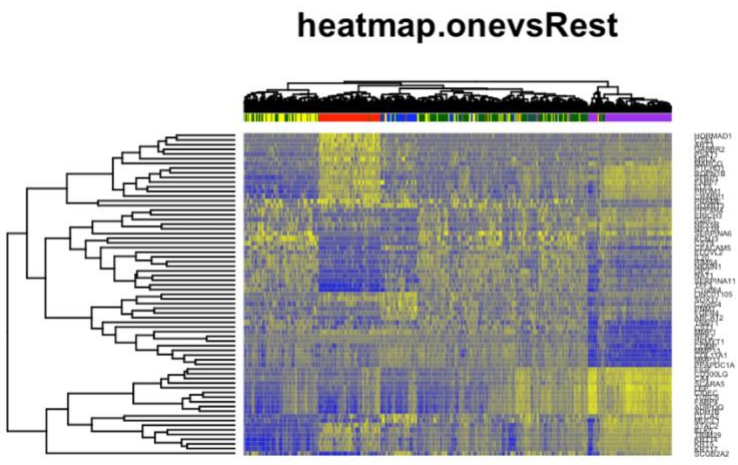

Figure 1: Heatmap with genes from one-vs-all t test.

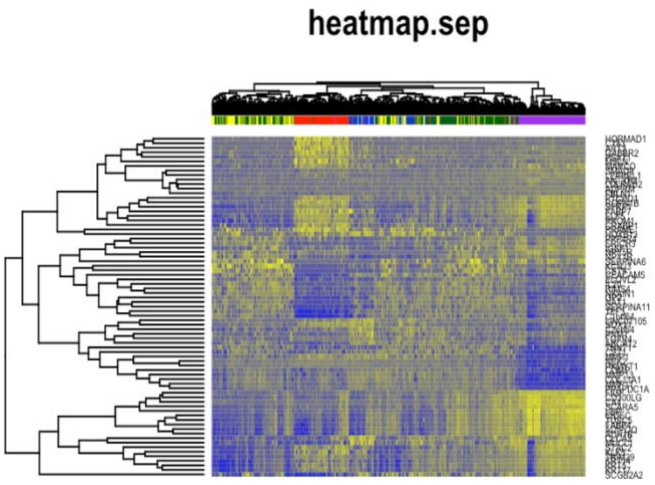

Figure 2: Heatmap with genes from one-vs-all t test, each-pair t test of Luminal A and Luminal B.

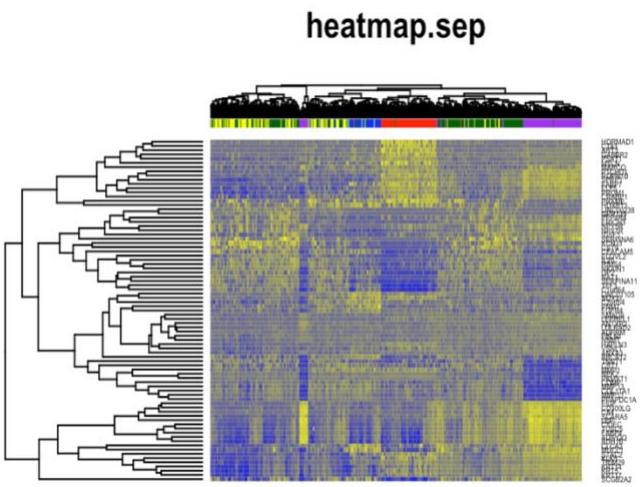

Figure 3: Heatmap with genes from one-vs-all t test, each-pair t test of Luminal A and Luminal B, each-pair t test of HER2-enriched and Luminal B.

carried out with all the features are shown in Figure 4. At this time, we can roughly see the boundary for five subtypes, and the issues between Luminal A and Luminal B is also better resolved compared to the heatmap in Figure1. Thus, the combination of all the above features is the feature set we are going to use for later classification. This feature set in total has 87 genes and we will compare it with the pam50 genes for different classifiers to choose a better feature set.

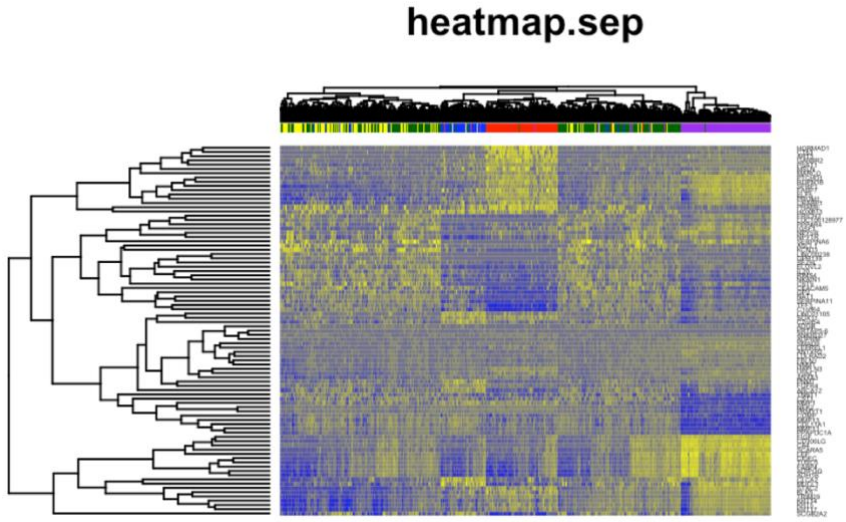

Figure 4: Heatmap with final features, which contain genes from one-vs-all t test, each-pair t test of Luminal A and Luminal B, each-pair t test of HER2-enriched and Luminal B and each-pair t test of normal-like and Luminal A.

2.3 K-fold cross validation

With the feature set, we can start training different classifiers. To make sure the classifier is not trained with bias, we take K-fold cross validation into the whole training set. The whole training set, which has known subtype information, has 626 samples, and we randomly split them into ten folds. For every time, we take nine folds of them to train the classifier and after training we make prediction on the other fold and calculate the accuracy. Finally, we take the average accuracy of the ten-fold training as the metric for the performance of a specific classifier.

2.4 Support vector machine (SVM)

The first classifier we use is SVM. It is a typical supervised machine learning algorithm mainly

used in classification. SVM makes use of a hyperplane, which acts like a decision boundary between various classes. One advantage of SVM is that it has different kernels to adapt itself to the classification of both linear and non-linear data. Here in our problem, we are going to try different kernels to find out the best classifier for our data. To classify our dataset, SVM will help us to find out the decision boundaries in the hyperplane, which is made up with the gene features we have selected or the pam50 genes. When a new sample is taken as input, it will be classified into a specific class by SVM model depending on what it learned in the training phase.

For the training phase, since we have split the whole dataset into ten folds, we are going to train SVM on nine folds every time and test on the other folds. The package I use in R is "caret". During the training of every fold, it will use bootstrapping to estimate an accuracy. After the training, we can make a prediction on the test fold. Finally, after averaging the accuracy across ten folds, we get a final accuracy for this classifier. The same process is also applied for pam50 genes. For SVM, we try three different kernels, which are linear, polynomial and radial. Since multiple parameters need to be well tuned for the polynomial and radial kernels, we include a set of parameters into the training process, and it will take the best combination of them to make further prediction. For example, in polynomial kernel, the parameters of cost C, degree and scale need to be well tuned, so we assign a list of value for every parameter. In the training process, the classifier will take bootstrapping and estimate an accuracy for every combination of parameters. As shown in Figure 5, it is an example for one-fold training of SVM polynomial classifier on the features I have selected. After searching of all the combinations of different parameters, the model finally takes the model at highest accuracy which is achieved with degree=1, scale=1 and C=0.1. For kernel radial, the parameters it has are cost C and sigma. The same parameter tuning process is also applied to it. In total, we have six different combinations of classifier and gene set.

```
Pre-processing: centered (87), scaled (87)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 564, 564, 564, 564, 564, 564, ...
Resampling results across tuning parameters:

  C      degree  Accuracy   Kappa
    0.1  1       0.8277070  0.7731973
    0.1  2       0.8215487  0.7616279
    0.1  3       0.8153186  0.7509658
    1.0  1       0.8173699  0.7603351
    1.0  2       0.8215487  0.7616279
    1.0  3       0.8153186  0.7509658
   10.0  1       0.8178209  0.7607887
   10.0  2       0.8215487  0.7616279
   10.0  3       0.8153186  0.7509658
  100.0  1       0.8178209  0.7607887
  100.0  2       0.8215487  0.7616279
  100.0  3       0.8153186  0.7509658

Tuning parameter 'scale' was held constant at a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were degree = 1, scale = 1 and C
 = 0.1.
```

Figure 5: An example of parameter tuning for polynomial SVM.

2.5 Random forest (Rf)

Random forest is much more straightforward than SVM. It works by aggregating the predictions made by multiple decision trees of varying depth. And every decision tree is trained on the bootstrapped set. When the random forest is used for classification and is presented with a new sample, the final prediction will be made with the majority of the predictions from each decision trees. Since it is quite straightforward, we do not spend too much time on tuning the parameters. We just carry out the random forest for both our features and the pam50 gene set. And take the average accuracy over ten folds for each of them.

2.6 K nearest neighbors (KNN)

KNN is a classification method based on a similarity measure, which for most of the cases is the calculation of distance between the input data. The class of a new input for a KNN classifier will be decided by the majority labels of its closest K neighbors. In our training procedure, we take different numbers of neighbors, K is set to 3, 5 and 7. Each classifier is applied for both our features and pam50 set. For KNN, we also have six combinations of classifier and dataset, and every combination is carried out under ten-fold training.

3. Results

3.1 Support vector machine (SVM)

For SVM, we summarize the average accuracy of ten folds for every combination of classifier and dataset in Table 1. In the result we can find that the pam50 genes always have a better

Average accuracy	SVM_linear	SVM_polynomial	SVM_radial
Our features (selected)	0.8577122	0.832052	0.8514895
PAM50 genes	0.8834395	0.9039288	0.9056689

Table 1: Summary for results of SVM

accuracy than our selected features. Even though in the heatmap, we think that our selected genes have separated the group with quite clear boundaries among classes, while the heatmap of pam50 genes has the Luminal A and Luminal B mixed together, the results still show better performance for pam50 genes. This may result from the fact that the clustering in pam50 heatmap is very clean, however in the heatmap with our selected features, the green panel conserves most of the samples from Luminal A but also has some noises of other classes. We think this may lead to the poor performance of our features.

3.2 Random forest (Rf)

The accuracy of random forest over ten-fold training is summarized in Table 2. Still pam50

Random forest	Accuracy
Our features (selected)	0.8739198
PAM50 genes	0.9009029

Table 2: Summary for results of random forest

genes show better performance than our selected features.

3.3 K nearest neighbors (KNN)

The accuracy of K nearest neighbors with different number of k assigned and calculated over ten-fold training is summarized in Table 3. K=5 is more suitable for these samples.

Accuracy	K=3	K=5	K=7
Our features (selected)	0.8020479	0.8225067	0.8050142
PAM50 genes	0.872406	0.8883566	0.886795

Table 3: Summary for results of K nearest neighbors

3.4 Model chosen and classification result

Summarize from the above accuracy tables, the highest accuracy is achieved at 0.906 by the combination of SVM_radial classifier and pam50 gene set. So for the next step, we are going to take the classifier that is well trained on them to classify the samples with unknown labels. First, we train SVM_radial on the whole dataset with known labels, at the same time, we set parameter cost with range of 0.1, 1, 10, 100 and set parameter sigma with range of 0.0001, 0.001, 0.01, 0.1, and 1. During the training process, the model achieved its highest accuracy with the parameter set of sigma= 0.01 and cost=10. With these parameter and classifier, we make prediction on the unknown dataset which contains 582 samples. The classification result is summarized in Table 4.

Subtype	Basal-like	HER2-enriched	Luminal A	Luminal B	Normal-like
Number of samples	101	44	293	120	24

Table 4: Summary of the classification result on unknown set.

3.5 Survival analysis

After the prediction, all the samples in the TCGA set have been assigned a specific subtype in breast cancer. For the next step, we exclude the control samples in the TCGA set first, and then extract the subtype information for all the tumor samples. We only keep the samples with survival records and merge their subtypes into the survival data frame. With the above data frame, we carry out the survival analysis among 5 subtypes of breast cancer, which is shown in Figure 6.

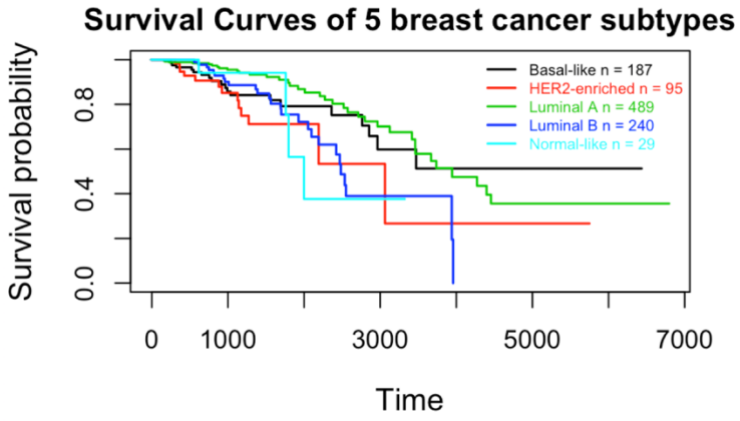

Figure 6: Survival curves of 5 breast cancer subtypes.

4. Conclusion

As a summary of all the analysis in this report, first, pam50 gene set is better than the gene set we have selected, it achieved better accuracy all the time. Second, all the classifiers can achieve fair accuracies. And the tuning of parameters is very important, cause different combination of parameters can have huge impact on the classification results. Third, for all the three models, SVM and random forest have achieved better performance for this classification problem compared with KNN model. However, in the computation time, the SVM takes the longest time to the training. Finally, in the survival analysis results, we think Basal-like and

Luminal A have better prognosis compared with the other groups. While since TCGA does not have very long follow-up time for the patients, the difference of prognosis in different subtypes here is not very obvious.