Feature selection by LASSO in gene expression data

Minstein, July 7th 2017

Data preparation:

We collected gene expression data in 461 leukemia samples with 1394 measurements from Broad Institute's Cancer Program Legacy Publication Resources (http://portals.broadinstitute.org/cgi-bin/cancer/publications/view/161). There are 28 different leukemia subclasses and we divided them into three major classes: ALL-T, ALL-B and AML. The sample size for these three classes are 64, 157 and 240, respectively.

To generate a training dataset, we individually sampled 32 samples from these classes and the final training dataset contains 96 samples with 1394 measurements. Using the same method, we generated a test dataset of equal size, which is separated from the training dataset.

Before training a model, we preprocessed the data by setting lower threshold to the average 0.05 quantile and upper threshold to the average 0.95 quantile among samples. Next, we set values to lower threshold when values are smaller than lower threshold and set values which are larger than upper threshold to upper threshold. After that, we performed column rank normalization on every samples, so as to make every samples comparable. Details for the algorithm are listed in the source code.

Model training:

We applied LASSO to reduce the dimension of the gene expression data, of which the size of measurements (genes) are much larger than that of the samples. R package `glmnet` implements LASSO and we selected `cv.glmnet` function to perform cross-validation to choose the appropriate parameters for our model.

Two selected λ are indicated by the vertical dotted lines in Figure 1. `lambda.min` is the value of λ which gives minimum mean cross-validated error and the other λ saved is `lambda.1se`, which gives the most regularized model such that error is within one standard error of the minimum. To control our model's degree of freedom, we selected `lambda.1se` as our parameter.

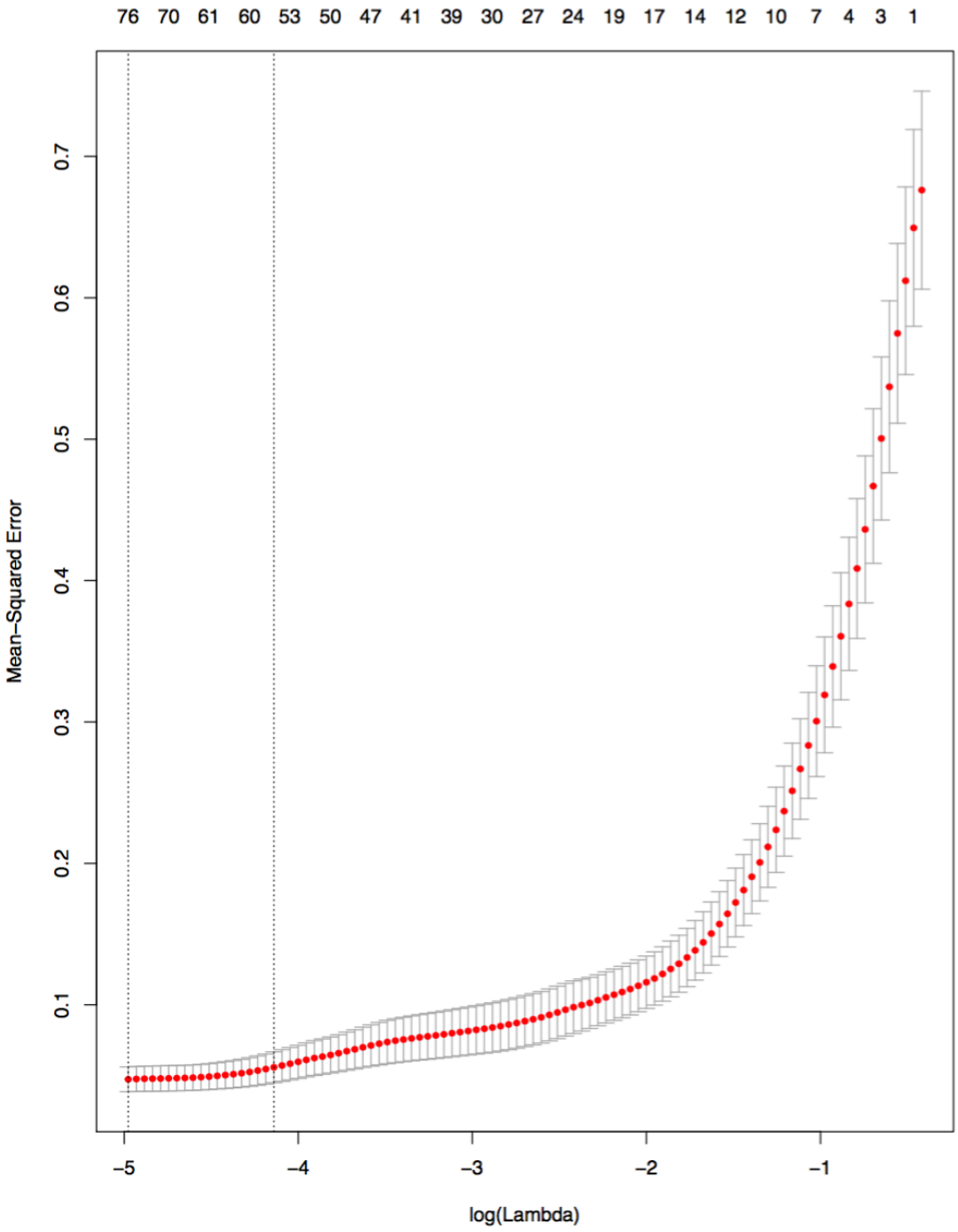

Figure 1. Mean-Squared Error changes by lambda in model selection.

We clustered the samples in training dataset itself by the features selected from our model. As showed in Figure 2, the different classes are well separated from each other. To test our model's accuracy, we classified the rest 365 samples, which are not used in model training. The performance of our model is generally satisfactory as the total predicting accuracy reaches 93% (Figure 3).

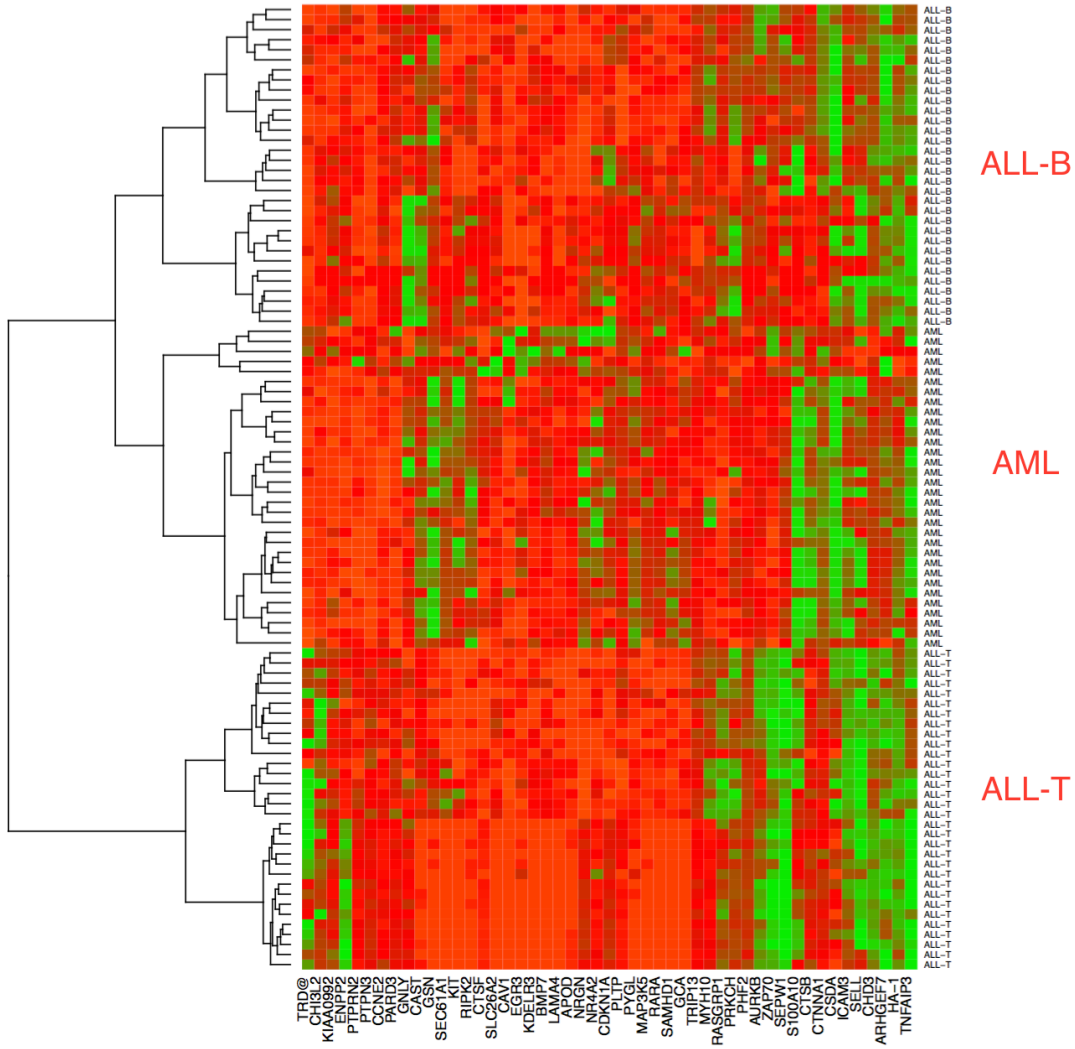

Figure 2. Heatmap of the samples in the training dataset. Rows are features selected from our model.

```
> print(c("ALL-T accuracy: ",ALLT.accuracy,"Test sample size: ",length(test_ALLT.pos)))
[1] "ALL-T accuracy: "    "0.90625"             "Test sample size: " "32"
> print(c("ALL-B accuracy: ",ALLB.accuracy,"Test sample size: ",length(test_ALLB.pos)))
[1] "ALL-B accuracy: "    "0.976"               "Test sample size: " "125"
> print(c("AML accuracy: ",AML.accuracy,"Test sample size: ",length(test_AML.pos)))
[1] "AML accuracy: "      "0.913461538461538"   "Test sample size: " "208"
> print(c("Total accuracy: ",accuracy,"Total test sample size: ", length(test.pos)))
[1] "Total accuracy: "          "0.934246575342466"         "Total test sample size: "
[4] "365"
```

Figure 3. The accuracy of our model on the rest samples.

To test whether the features can efficiently classify unknown samples, we clustered the test dataset by the features generated from our trained model. As showed in Figure 4, except for six samples which in fact belongs to ALL-B class, other samples are well clustered.

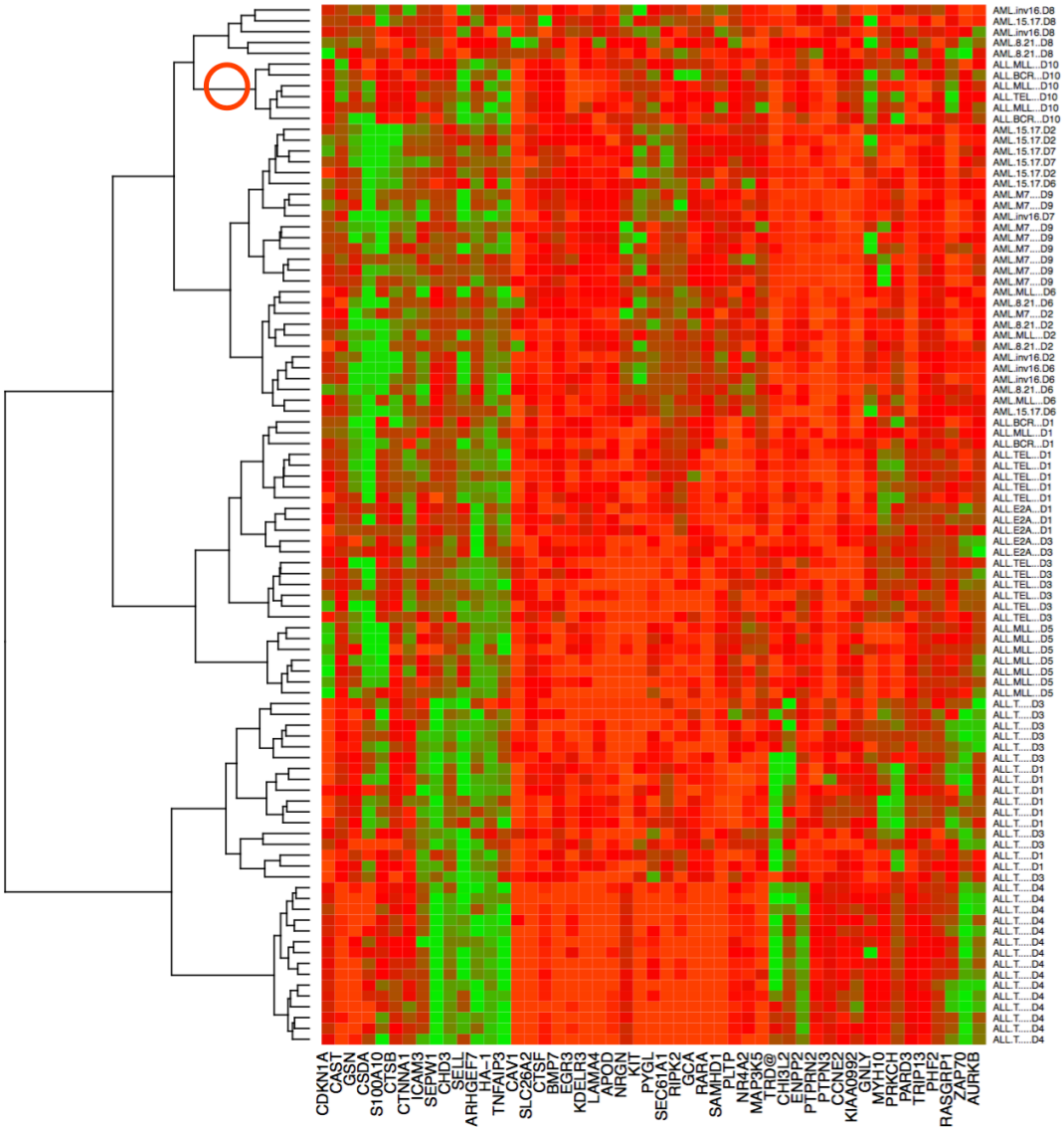

Figure 4. Heatmap of the samples in the test dataset. Rows are features selected from our trained model. The red circle indicates the wrong clustering of six samples, which in fact belongs to ALL-B.

In conclusion, LASSO can efficiently deal with feature selection in gene expression data. Our method shows good performance and might be applicable in clinical use in future.

Note:
Source code: LASSO-feature-selection.R
Dataset: Leukemia2.all.cls; Leukemia2.all.gct