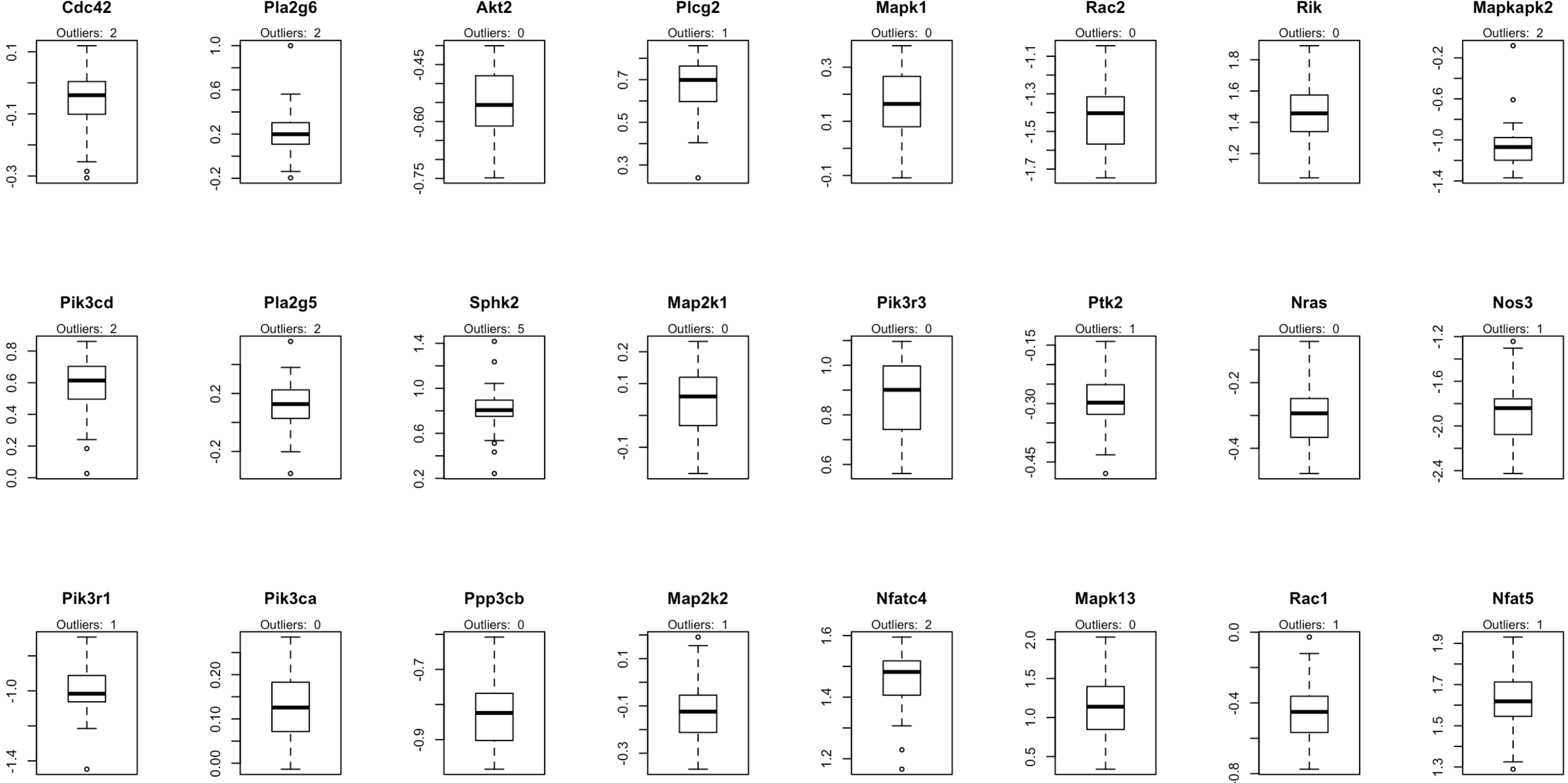
Final project

By Ming han, Lei Kang, and Yanyu Long

In this project, we used best subset, forward stepwise, backward stepwise and Lasso to find the best yet simple model. All these methods perform variable selection, therefore can help to identify the most important genes in predicting Mapk1. For the first three method, we used Cp, BIC and cross-validation to find the best models; for Lasso, we used cross-validation to determine the best lambda and then the best model. Then, all the methods, together with random forest and Ridge (which do not perform variable selection but we included them to compare with the simpler models generated by other methods), are compared using 10 fold cross-validation. The result shows that besides random forest and Ridge, Lasso produces the lowest test MSE. Since Lasso can select a small number of variables, we used this method to produce our final model, which is **Mapk1 ~ Rik + Pik3cd + Pik3r3 + Rac1 + Nfat5**. We also checked the linear assumptions of the model.

1. Summary of the data

Check the distribution of all the data:



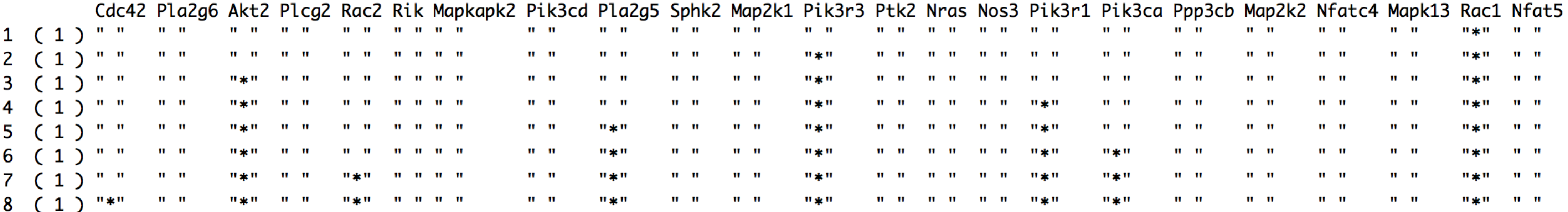
*Figure 1. boxplots of all the variables*

There are some outliers in some variables (1.5 IQR). In the later analysis we did not exclude them, because the sample size is small (n = 40); if all the outliers are excluded, the sample pool will be reduced to about 25.

1. Best subset, forward and backward models

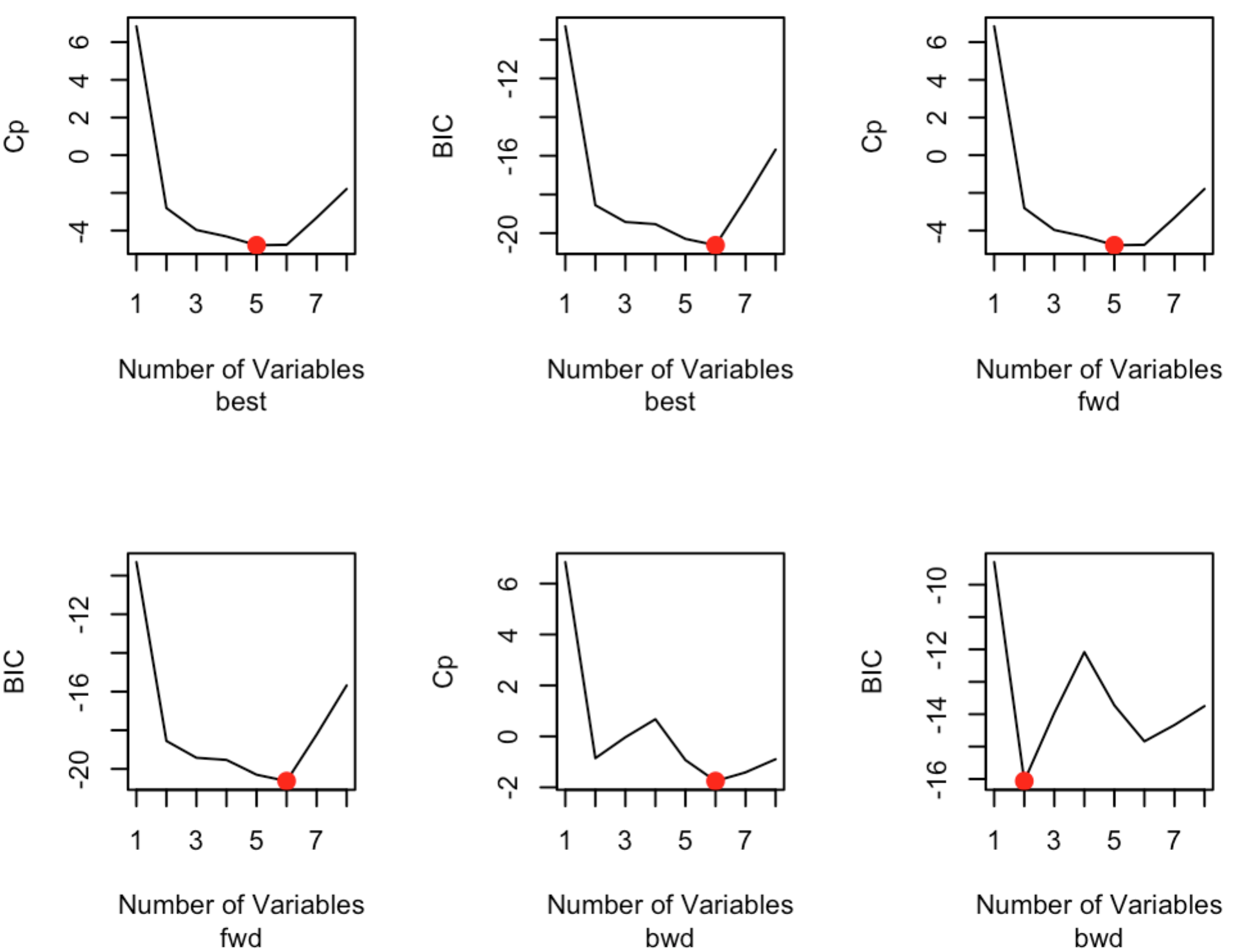
We first split the data into training data and test data. We used the training data to obtain the best model for each method, and used the test data to get the test MSE. Since we plan to do cross-validation to obtain the test MSE, the splitting procedures will be repeated nfold (e.g. 5 or 10) times and the training set will be different each time. Below we reported results from one of the training set (when nfold = 10, set.seed(1), first split) just to show examples. The results are different every time when we have different training set.

Each of the three methods gave 8 models with 1-8 variables. An example from the best subset method is given below (each method gave different 8 set of models). “\*” indicate the variable is included in such a model.



Then, we used Cp, BIC and CV to select the best model.

* Evaluation of the three models using Cp and BIC

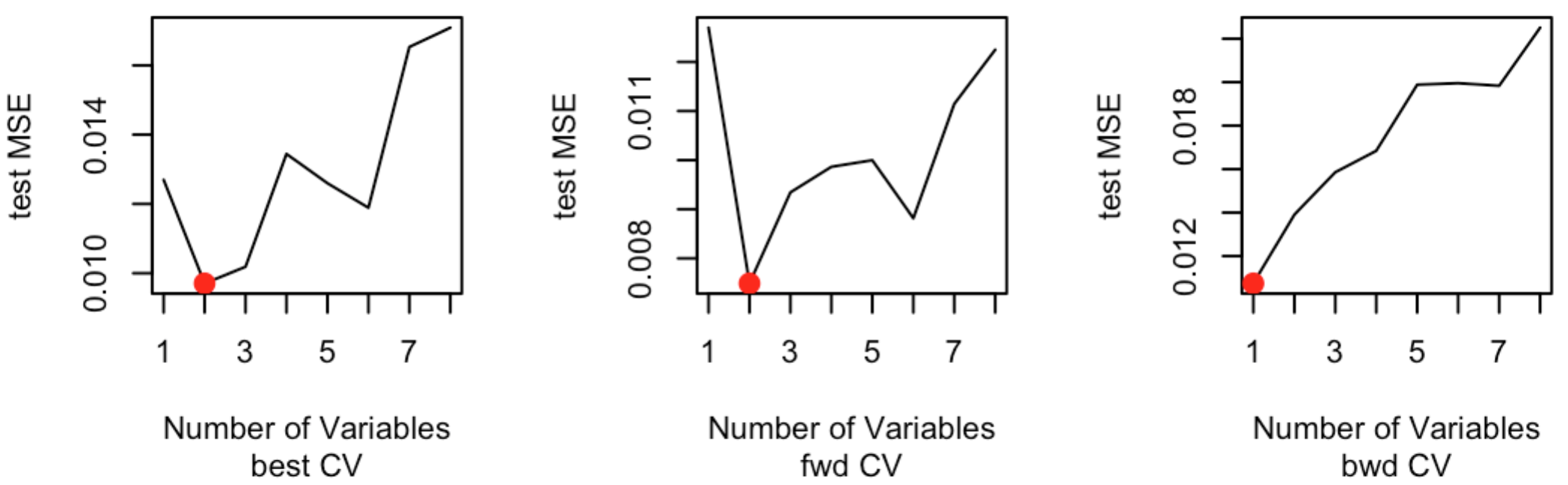


*Figure 2. Cp and BIC for models selected by the three methods*

The best models are selected when Cp/BIC is the lowest. The x-axis is the number of variables, which indicate the 8 models for each method. The red dots are the indices of the selected best model using corresponding model-selection and evaluation method. As shown from the graph, with this particular training set, Cp and BIC gave different best models for each method (but can be the same with some training sets).

* Evaluation of the three models using 10-fold cross-validation[[1]](#footnote-1)

This CV was conducted to select the best model within each method. The best models selected by CV are as below:



*Figure 3. cross-validation test MSE for models selected by the three methods*

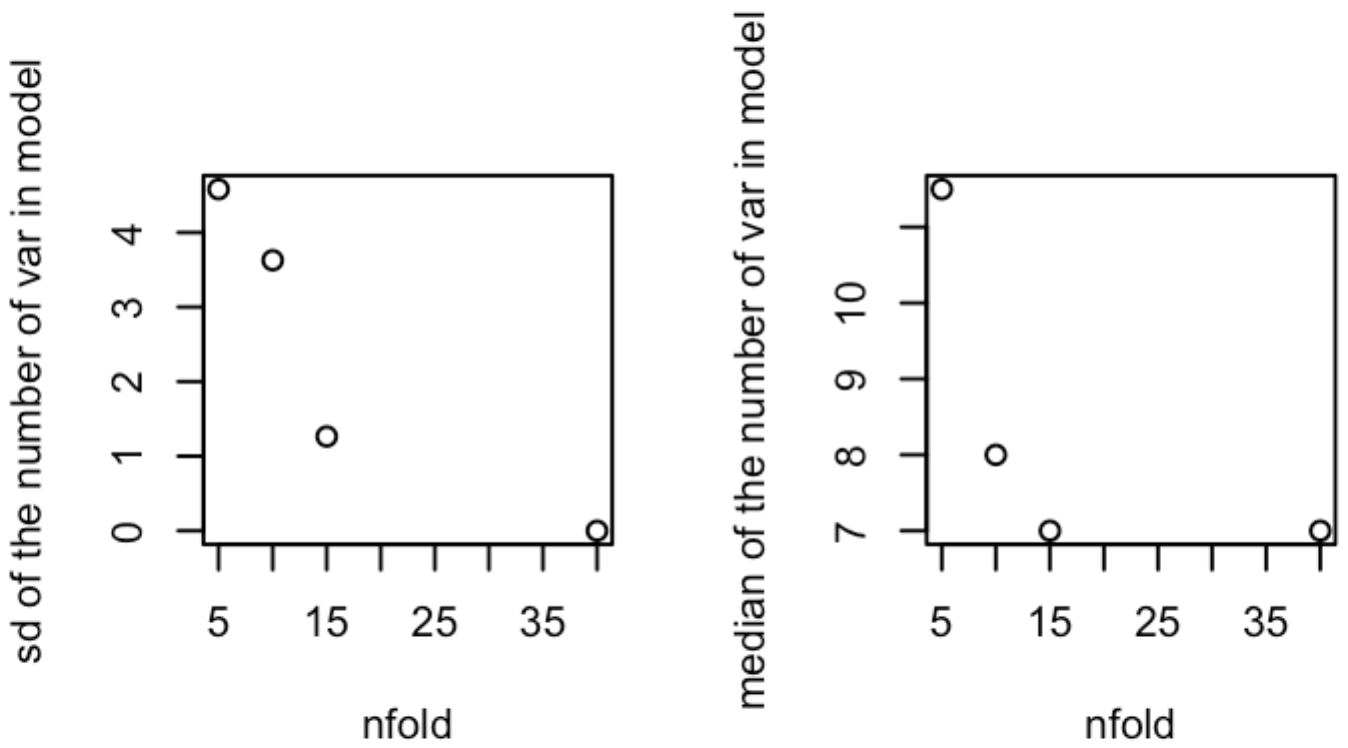
The y-axis is the test MSE using 10-fold CV, with the red dots being the lowest MSE and therefore the best models. Again, for each method, different best models were given using CV, and they are also different from those selected using BIC and Cp (but it seems that CV tend to give smaller model).

1. Lasso method:

Within each training set, we used 10 fold cross-validation to obtain the best lambda that gives the lowest test MSE, and used the best lambda to get the best model for this training set (variables with non-zero coefficients).

For this particular training set, the best lambda is 0.02462796 (set.seed(1)); when applying the best lambda to the whole training set, it produced a best model, which contains 6 variables.

We noticed that when set.seed are different, the models produced are different and contain different variables. We did a brief check on the influence of the number of folds on the produced best model. We used the number of selected variable (*n*) as a simple indicator of whether the final models are the same. We obtained the standard deviation and median of *n* by setting set.seed different. The results are present below:



*Figure 4. sd and median of the number of variables in the best models using different set.seed(n)*

For this training set (the generalization, however, is a little limited), as nfold increases, variation decreases; also the median becomes more stable. Therefore when nfold is bigger, the best models given by this method should be more stable. In the model comparison in the next section, we set nfold =10 (instead of 5) for both CV in model selection and obtaining the test MSE to compare methods.

1. Compare methods:

The above part provides examples of model selection within a given training set. The model will be tested on a test set (the rest of the set) to obtain a test MSE. In 10-fold CV, this procedure will be repeated 10 times and the MSE will be averaged to get a final test MSE. The one that has the lowest test MSE should be the best method and provide the best model

Since there may still be some variation, we used different seeds (seed 1-20) to get values for each model and obtain the mean and variation. The one that yields the lowest test MSE (and preferably lowest variation) should provide the best model.

The results are summarized below:

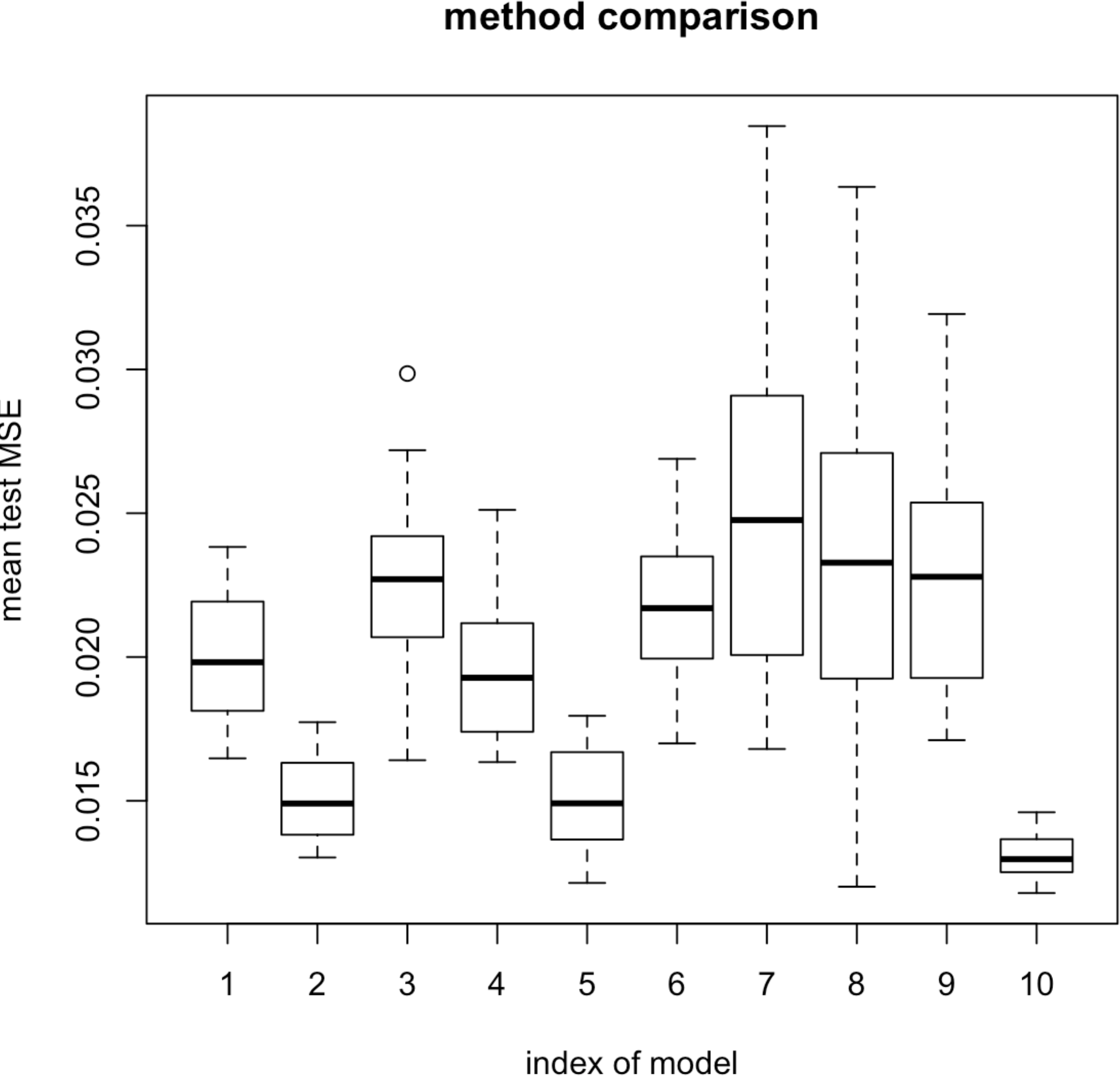


Figure 5. Test MSE of all the 10 methods using different seeds

In the graphs above, the x-axis are the indices for the methods, they are:

1-3: Best subset, forward, backward + Cp

4-6: Best subset, forward, backward + BIC

7-9: Best subset, forward, backward + CV

10: Lasso + CV

The y-axis is the test MSE. As can been seen, the #10 model, that is, Lasso (using CV as the evaluation method), provides consistently the lowest MSE with lowest variation, therefore stably give the best model. Besides Lasso, the #2 and #5 model, that is, forward stepwise method (using either Cp or BIC as the evaluation method), also produce results among the lowest and can potential give good models.

The model generated by the three method (#10, #2 and #5) are:

(2). Mapk1 ~ Akt2 + Rik + Pik3r3 + Rac1 (Best/Forward BIC)

(5). Mapk1 ~ Akt2 + Rik + Pik3r3 + Pik3r1 + Rac1 (Best/Forward Cp)

**(10). Mapk1 ~ Rik + Pik3cd + Pik3r3 + Rac1 + Nfat5 (Lasso)**

Although #2 gives a simpler model, it does not have as good performance as that of Lasso. Therefore, we decided on Lasso to provide the best model[[2]](#footnote-2).

We then compare Lasso compare with random forest and Ridge. Different from Lasso and other methods motioned above, these two methods do not do well on variable selection. We do the comparison for the sake of examining whether a simple model produced by Lasso can have as good as, if not better, performance as more complex models.

We also tried random forest method to create a model. The random forest builds a number of decision trees, every single time a predictor from the full set is selected to split in the tree.

We creates 800 decision trees for random forest in our estimation. In order to compare with other methods, we also use ten-fold cross validation to train our model of random forest.

We can observe the importance for each predictor to interpret the result from the random forest. The way to understand the importance of each predictor is to capture the variance base on %IncMSE and IncNodePurity.

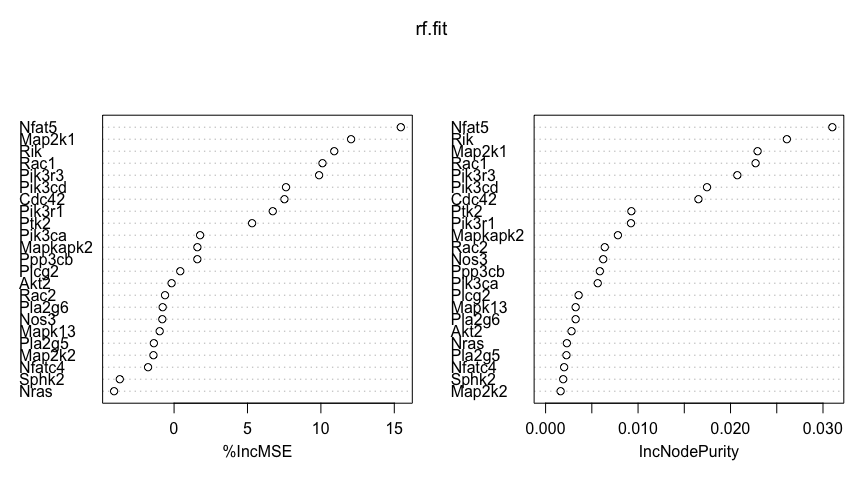


Figure 6. %incMSE and incNodePurity of results of random forest

For the %IncMSE, the larger the mse of prediction(which is estimated with out-of-bag-CV) if a result of predictor is permuted, the more importance of predictor is.

For the IncNodePurity, it interprets the importance of predictor in another way. The most important predictor, that split the data, could achieve the highest increase in node purities.

According to our results, the Nfat5, Map2k1, Rik, Rac1, Pik3r3 are the five most important predictors to the model, which is very similar to the result that our Lasso method create.

The testing error is around 0.010, which also achieve a good performance for the prediction.

Random Forest VS the Ridge VS the Lasso

The procedure of cross-validation: Set seed from 1 to 50, and with each seed randomly shuffle dataset to perform one time cross-validation. Use boxplot to show means, variances and outliers of different methods.

Total lasso: 0.0134601980207286

Total ridge: 0.0103132458276357

Total random forest 0.010975346957304

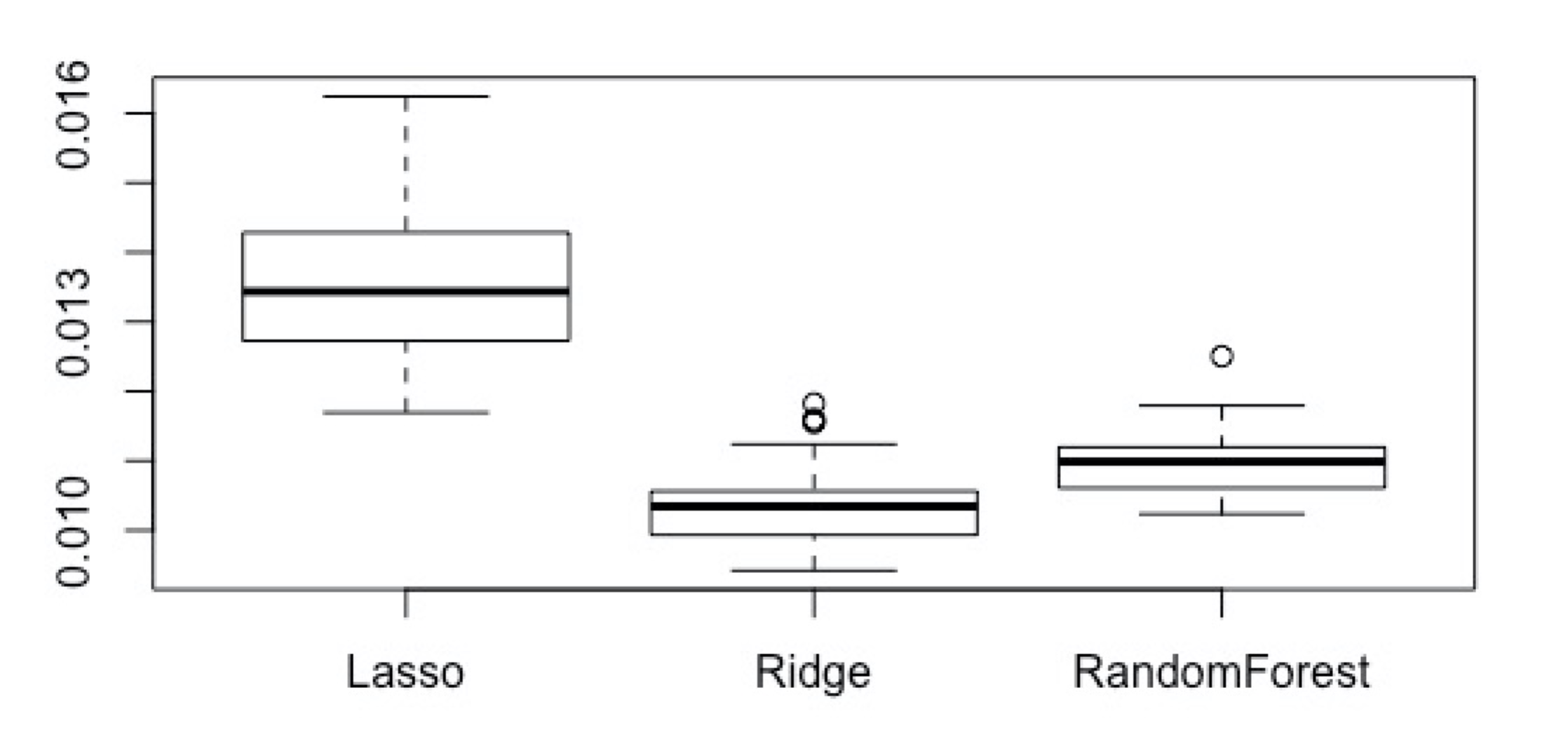


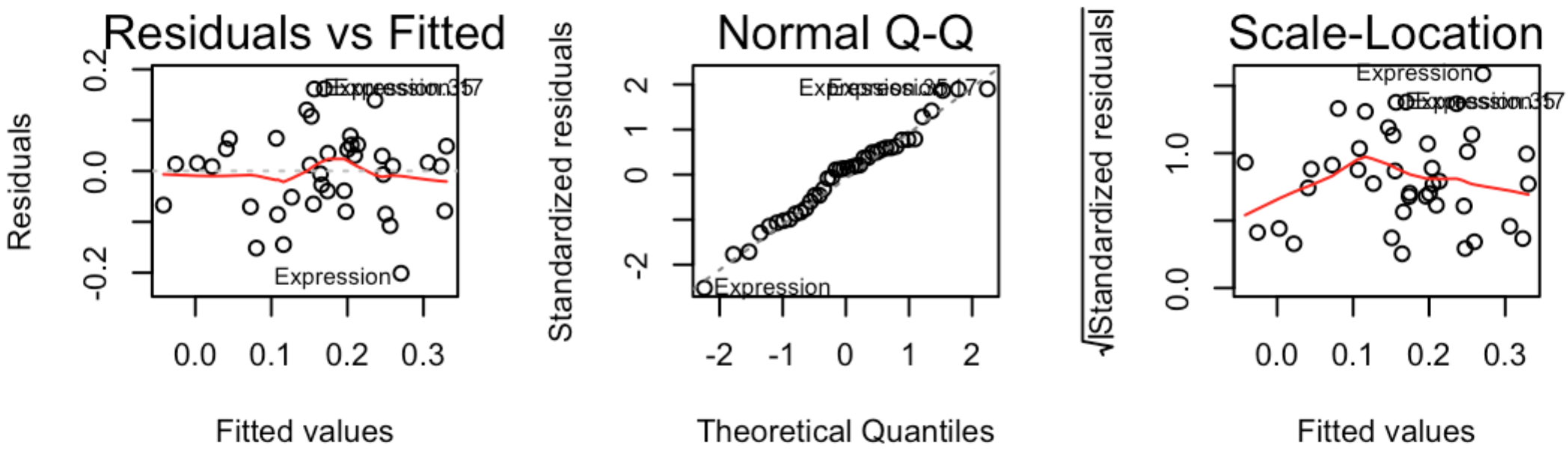
Figure 7. test MSE of Lasso, Ridge and random forest

The model of the Ridge performs the best among the models, then the random forest, then the Lasso. However, the first two models are also more complicated. The Ridge doesn’t perform the variable selection. Although the random forest prune insignificant variables, the model is still close to the full mode. In contrast, the Lasso has the feature of variable selection, so its model is more interpretable. Therefore, in order to obtain a simple and easily interpreted model, we decided to use the Lasso for this dataset.

We learned from the random forest and the Ridge that models with high computation complexity could create good prediction. Hence, if heavy computation is affordable, models which require high computation could be a good choice.

1. Check linear assumptions:

At last, we also checked some assumptions of linear models, such as whether the relationship between the responses and predictors is linear and whether there are outliers that may distort the results (some outliers are shown in the boxplots). By plotting the model, we can see that 1) there is no apparent pattern in residuals vs. fitted values; 2) the QQ plot of standardized residuals suggests a normal distribution; 3) there is no obvious outliers (no standardized residual exceeds 3). Therefore the model satisfied the linear assumptions.

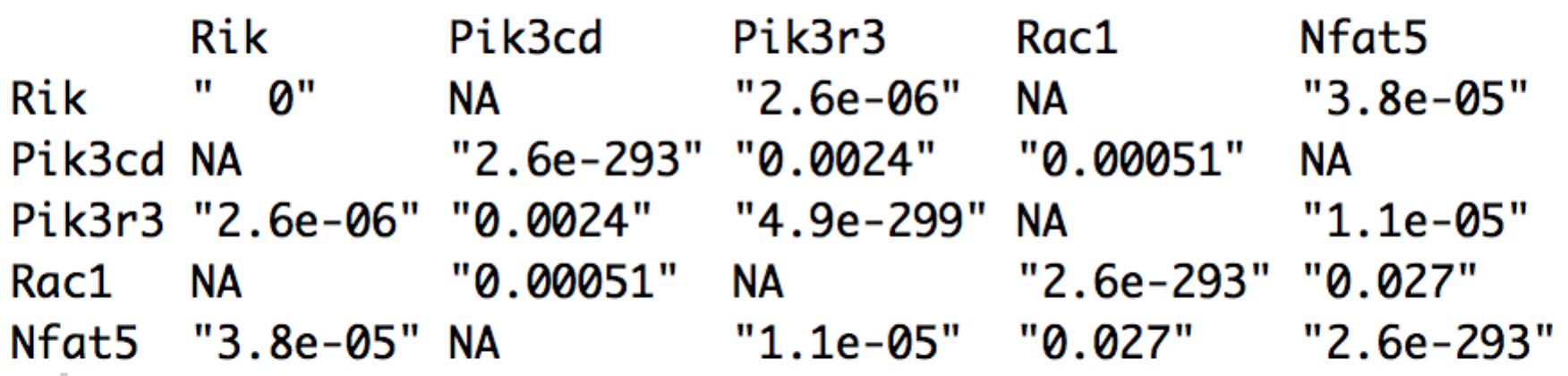


*Figure 8. Check the assumptions of linear models*

To summary, we used several methods to generate the best model and among them, Lasso provides a best model that has the lowest test MSE while not sacrificing simplicity and interpretability. Although comparing to methods that generate more complex models such as random forest and ridge, Lasso has slightly higher test MSE, it is good enough and is simple. The final model is:

**Mapk1 = -0.132 + 0.106 \* Rik – 0.032 \* Pik3cd + 0.164 \* Pik3r3 + 0.132 \* Rac1 + 0.051 \* Nfat5**

1. We did not use validation approach set because the sample size is small, the training and test set would be 20 if the validation set approach were used, which would yield results (the number of variables) with large variation (tested by using different set.seed(n)). The number of folds is set to be 10 instead of 5, because when k=5, the results also have larger variation, while the variation is small when k = 10 (always models with 1-3 variables are selected) [↑](#footnote-ref-1)
2. The #10 model does have a problem of collinearity. As can be shown below, many variables are correlated with other variables (NA indicates p > 0.05). However, although some variables are correlated, each individual variables still provide extra information that other variables do not include, and we lack a reliable method to determine whether deleting one or two variables in this model will produce a much worse model, we retain the model as it is produced by Lasso. Another option is to use one of the models generated by other methods, like #2 or #5 (#2 is simpler). However, we do want to focus more on the test MSE, and #10 model is simple enough; so we still stick to Lasso and the model it produces.

    [↑](#footnote-ref-2)