HarvardX PH125.9x Capstone CYO (Choose Your Own) project - Early autoimmune biomarker for multiple sclerosis

Kicheol Kim

Feb. 5. 2022

Introduction

Multiple sclerosis (MS) is a neurodegenerative autoimmune disease of the central nervous system caused by demyelination. The international multiple sclerosis genetics consortium (IMSGC) identified identified more than 200 independent risk loci in multiple sclerosis through genome-wide association studies. Further studies shown that most of risk loci harbor genes expressed in immune cells such as T cells. Multiple sclerosis have 4 different type of disease course according to the National MS Society. CIS (clinically isolated syndrome) is a first episode of inflammation and demyelination. CIS may or may not go on to develop MS. RRMS (relapsing-remitting MS) is the most common disease course. RRMS experiences neurological attack (relapse/exacerbation) followed by periods of recovery (remission). SPMS (secondary progressive MS) follows initial RR courses.Some RRMS patients will transition to secondary progressive course which is progressive worsening of neurologic function. PPMS (primary progressive MS) is progressive worsening of neurologic function from the onset of the disease.

Kim et al. (2021; PMID 33374005) published post-translational modification in CD4+ T cells is critical in MS pathogenesis. This publication provides CD4+ T cell transcriptome dataset from early diagnosed multiple sclerosis patients. Therefore, in this analysis, I used differential expression analysis and machine learning techniques to identify potential early biomarker for autoimmune response in MS.

The dataset (gene expression) has been downloaded from GEO (Accession No. GSE137143). This dataset includes 3 different cell types (CD4+ T cell, CD8+ T cells, CD14+ monocytes). Since I have interested in CD4+ T cells, I kept only CD4+ cells and removed CD8+ and CD14+ cells.

```
## [1] 138  14
```

```
##                 geoID                                 title age.at.exam
## 13311d-CD4 GSM4071522           Healthy controls, 13311d-CD4          35
## 43213b-CD4 GSM4071526 Treatment naïve MS patients, 43213b-CD4          34
## 46913b-CD4 GSM4071529 Treatment naïve MS patients, 46913b-CD4          72
## 47413a-CD4 GSM4071535 Treatment naïve MS patients, 47413a-CD4          44
## 47513a-CD4 GSM4071538 Treatment naïve MS patients, 47513a-CD4          63
## 47713b-CD4 GSM4071541 Treatment naïve MS patients, 47713b-CD4          24
##               cell.type disease.duration          disease.state
## 13311d-CD4 CD4+ T cells               NA        Healthy control
## 43213b-CD4 CD4+ T cells                0 Multiple sclerosis (MS)
## 46913b-CD4 CD4+ T cells               28 Multiple sclerosis (MS)
## 47413a-CD4 CD4+ T cells                2 Multiple sclerosis (MS)
## 47513a-CD4 CD4+ T cells                2 Multiple sclerosis (MS)
## 47713b-CD4 CD4+ T cells                0 Multiple sclerosis (MS)
##           disease.subtypes edss gender disease.state2   sampleID cell.type2
## 13311d-CD4               NA   NA      M             HC 13311d-CD4        CD4
## 43213b-CD4               RR  2.0      F             MS 43213b-CD4        CD4
## 46913b-CD4               SP  2.5      F             MS 46913b-CD4        CD4
## 47413a-CD4               PP  4.0      M             MS 47413a-CD4        CD4
## 47513a-CD4               PP  2.0      F             MS 47513a-CD4        CD4
## 47713b-CD4               RR  2.5      M             MS 47713b-CD4        CD4
##            disease.subtype2 AgeAtExamGrp
## 13311d-CD4               HC        30_40
## 43213b-CD4               RR        30_40
## 46913b-CD4              PMS        60_90
## 47413a-CD4              PMS        40_50
## 47513a-CD4              PMS        60_90
## 47713b-CD4               RR        10_30
```

Methods and Analysis

1. Download dataset

I have downloaded RSEM outputs from GEO (Acc. No. GSE137143) and decompressed downloaded file. The count matrix created using tximport package.

2. Normalization and differential expression analysis using DESeq2

The gene expression counts were normalized using DESeq2 and transformed by variance stabilization transformation for machine learning model building.

Here, I used differential expression analysis model design like this in the DESeq2:

$$\sim diseaseSubtype + gender + age - 1$$

I would like to compare in disease, and gender and age are covariates. I used all dataset for normalization and dispersion calculation. Since There are 3 different disease courses and healthy controls, I added '- 1' variable to get individual disease subtype and compare between single disease subtype and healthy control.

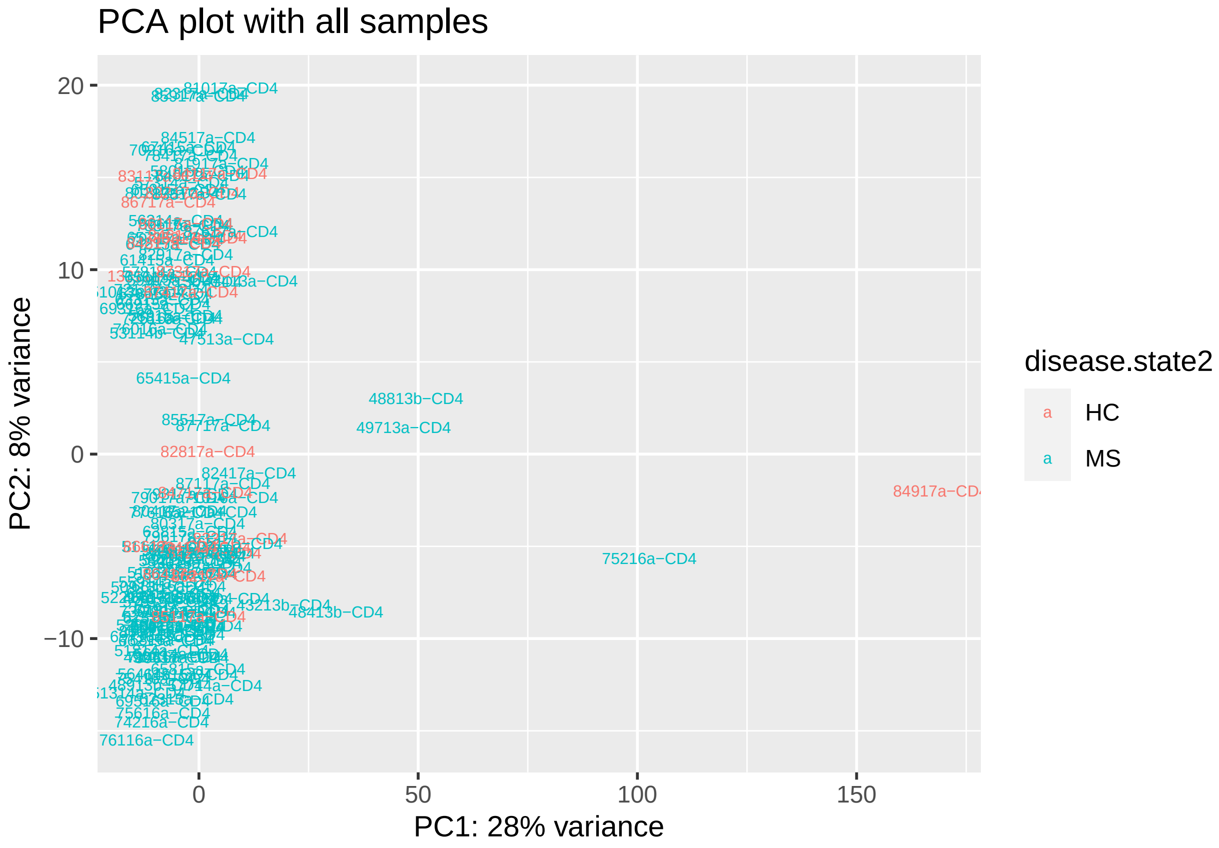
```
## using counts and average transcript lengths from tximport

## estimating size factors

## using 'avgTxLength' from assays(dds), correcting for library size

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
## -- replacing outliers and refitting for 932 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
```

```
## estimating dispersions
```

```
## fitting model and testing
```

PCA plot and outlier samples PCA plot showed 2 outlier samples in CD4+ T cell samples. Therefore, I removed these outlier samples from the DESeq2 object. Then count data transformed by variance stabilization transformation for further machine learning analysis.

Gene expression of immune cells are affected by gender. In the PCA plot, I observed 2 major clusters that separated by gender. Therefore, gender effect is important in the analysis of immune cells, also critical to use gender as a covariate.



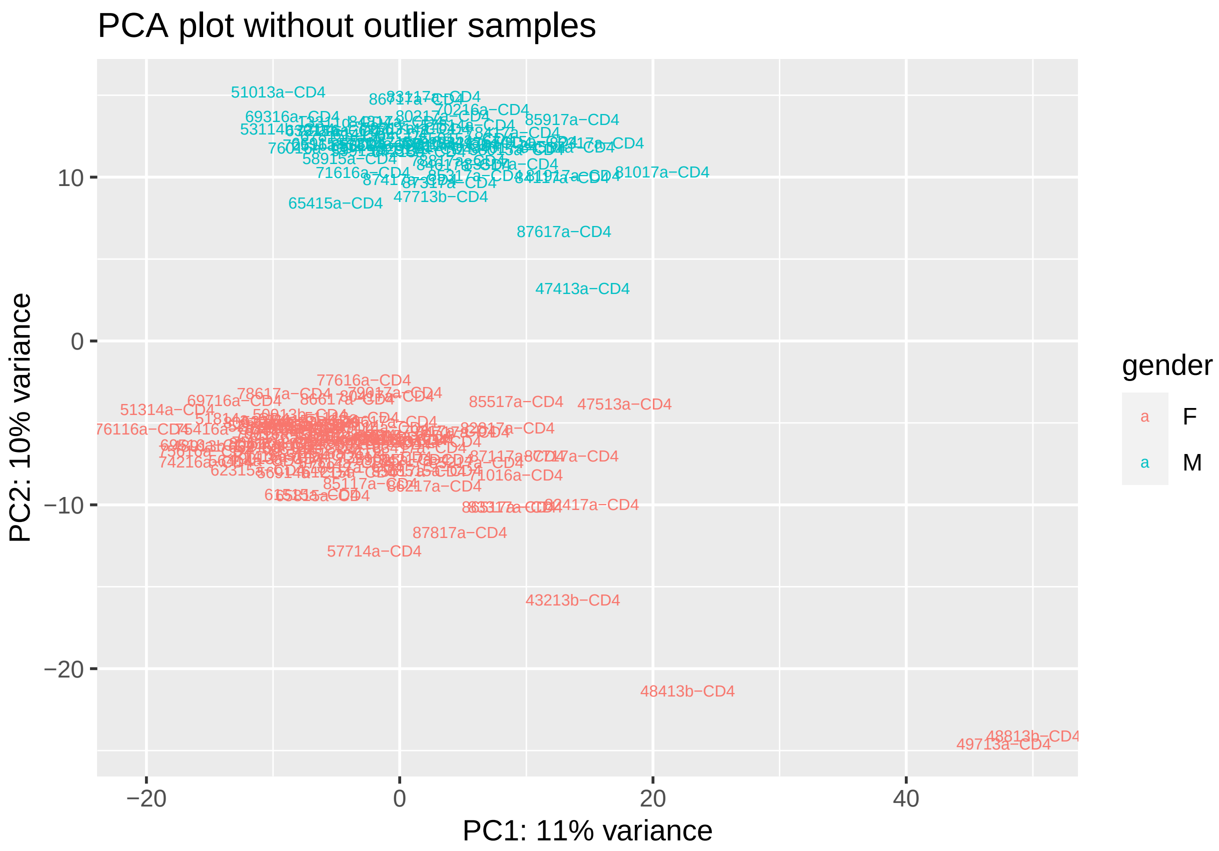
```
## using pre-existing normalization factors
```

```
## estimating dispersions
```

```
## found already estimated dispersions, replacing these
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 179 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```



3. Prepare dataset for machine learning

Although some CIS patient may not develop to MS, CIS is beginning of the MS disease course. In addition, there was almost no difference between CIS and RRMS in previous analysis (results does not shown here). Therefore, I performed differential expression analysis between CIS and healthy control to remove noise then used top significant genes for model building.

Next, because I want to compare between CIS and healthy controls, I excluded other disease course samples. Then selected samples were randomly split into training (80%) and test (20%) samples.

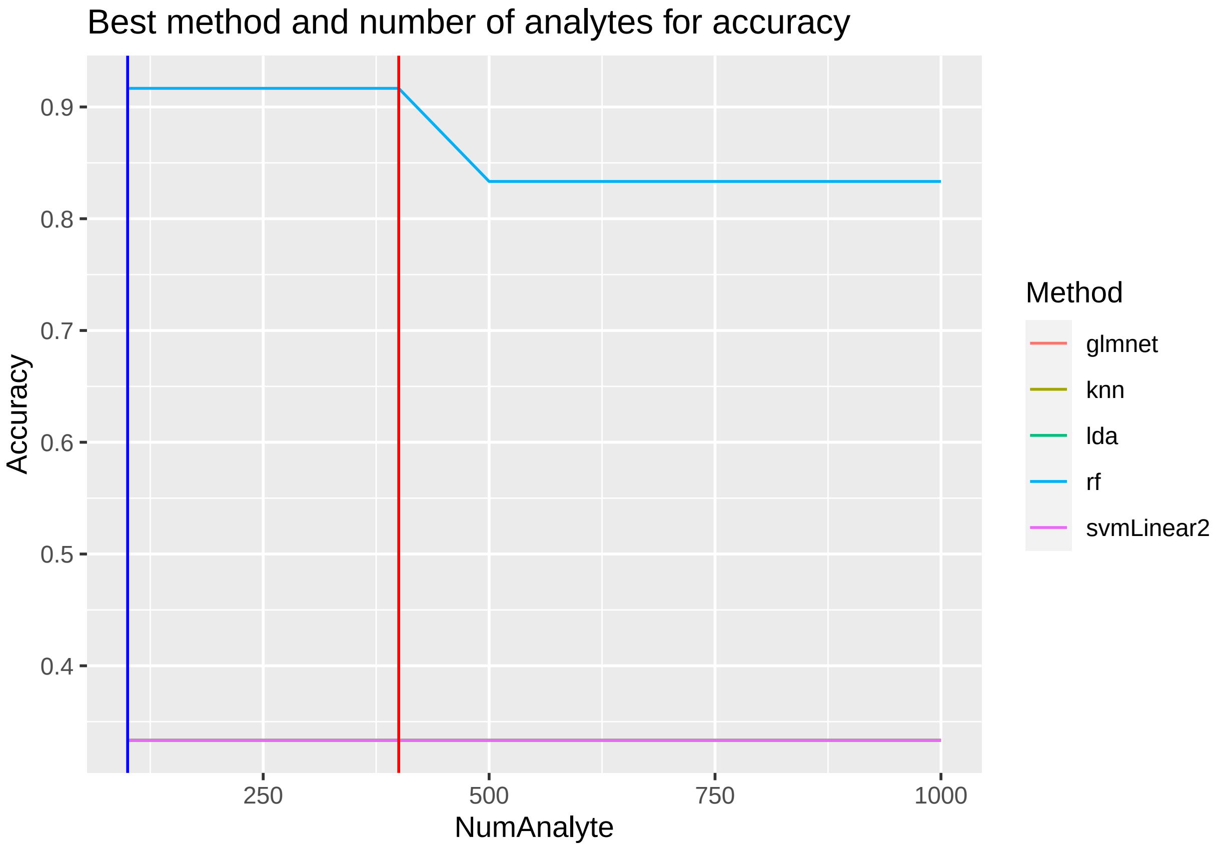
```
## [1] 38281    59
```

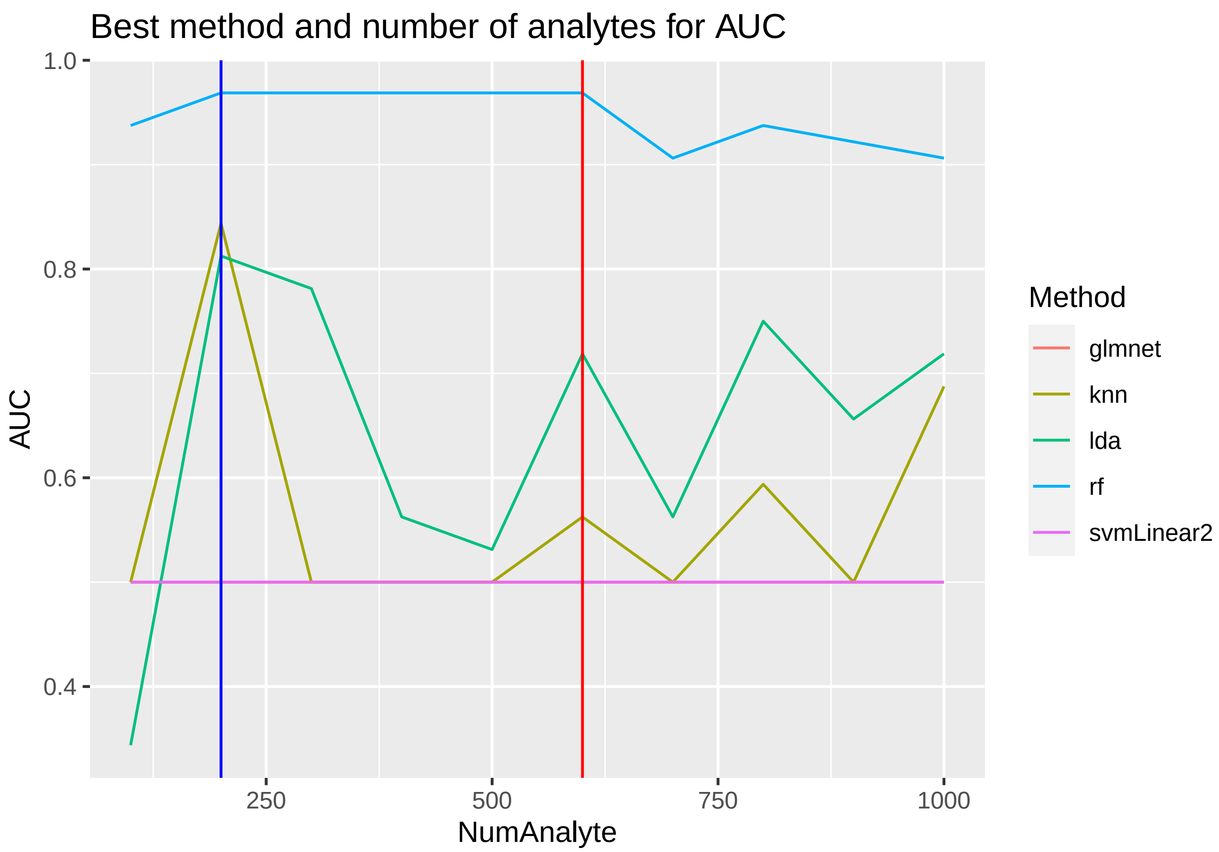
I created a function to summarize modeling results of machine learning.

```r
## function to retrieve model fitting results
fit_output <- function(fit, test_set, test_set_meta, model_name, gene_n){
  confusionMatrix <- confusionMatrix(predict(fit, test_set), test_set_meta$disease.subtype2)
  confusionMatrix

  pred_for_roc <- as.data.frame(predict(fit, test_set, type="prob"))
  pred_for_roc$predict <- names(pred_for_roc)[1:2][apply(pred_for_roc[,1:2], 1, which.max)]
  pred_for_roc$observed <- test_set_meta$disease.subtype2

  roc_obj <- roc(pred_for_roc$observed, as.numeric(pred_for_roc$CIS))

  ml_res <- data.frame(Method = model_name, NumAnalyte = gene_n,
                       AUC = auc(roc_obj),
                       Accuracy = confusionMatrix$overall["Accuracy"],
                       Sensitivity = confusionMatrix$byClass["Sensitivity"],
                       Specificity = confusionMatrix$byClass["Specificity"],
                       row.names = NULL)

  return(ml_res)
}
```

4. Try multiple modeling methods with various number of genes

I have run 5 different modeling methods (lda, rf, glmnet, knn, svmlinear) with 10 different number of analytes (100 genes ~ 1000 genes).


Best method and number of analytes for accuracy
Accuracy
0.9
0.8
0.7
0.6
0.5
0.4
250
500
750
1000
NumAnalyte
Method
glmnet
knn
lda
rf
svmLinear2



```
##         Method NumAnalyte      AUC  Accuracy Sensitivity Specificity
## 1           rf        200 0.968750 0.9166667           1        0.75
## 2           rf        300 0.968750 0.9166667           1        0.75
## 3           rf        400 0.968750 0.9166667           1        0.75
## 4           rf        100 0.937500 0.9166667           1        0.75
## 5           rf        500 0.968750 0.8333333           1        0.50
## 6           rf        600 0.968750 0.8333333           1        0.50
## 7           rf        800 0.937500 0.8333333           1        0.50
## 8           rf        900 0.921875 0.8333333           1        0.50
## 9           rf        700 0.906250 0.8333333           1        0.50
## 10          rf       1000 0.906250 0.8333333           1        0.50
## 11         knn        200 0.843750 0.3333333           0        1.00
## 12         lda        200 0.812500 0.3333333           0        1.00
## 13         lda        300 0.781250 0.3333333           0        1.00
## 14         lda        800 0.750000 0.3333333           0        1.00
## 15         lda        600 0.718750 0.3333333           0        1.00
## 16         lda       1000 0.718750 0.3333333           0        1.00
## 17         knn       1000 0.687500 0.3333333           0        1.00
## 18         lda        900 0.656250 0.3333333           0        1.00
## 19         knn        800 0.593750 0.3333333           0        1.00
## 20         lda        400 0.562500 0.3333333           0        1.00
## 21         knn        600 0.562500 0.3333333           0        1.00
## 22         lda        700 0.562500 0.3333333           0        1.00
## 23         lda        500 0.531250 0.3333333           0        1.00
## 24      glmnet        100 0.500000 0.3333333           0        1.00
## 25         knn        100 0.500000 0.3333333           0        1.00
## 26 svmLinear2        100 0.500000 0.3333333           0         1.00
## 27     glmnet        200 0.500000 0.3333333           0         1.00
## 28 svmLinear2        200 0.500000 0.3333333           0         1.00
## 29     glmnet        300 0.500000 0.3333333           0         1.00
## 30        knn        300 0.500000 0.3333333           0         1.00
## 31 svmLinear2        300 0.500000 0.3333333           0         1.00
## 32     glmnet        400 0.500000 0.3333333           0         1.00
## 33        knn        400 0.500000 0.3333333           0         1.00
## 34 svmLinear2        400 0.500000 0.3333333           0         1.00
## 35     glmnet        500 0.500000 0.3333333           0         1.00
## 36        knn        500 0.500000 0.3333333           0         1.00
## 37 svmLinear2        500 0.500000 0.3333333           0         1.00
## 38     glmnet        600 0.500000 0.3333333           0         1.00
## 39 svmLinear2        600 0.500000 0.3333333           0         1.00
## 40     glmnet        700 0.500000 0.3333333           0         1.00
## 41        knn        700 0.500000 0.3333333           0         1.00
## 42 svmLinear2        700 0.500000 0.3333333           0         1.00
## 43     glmnet        800 0.500000 0.3333333           0         1.00
## 44 svmLinear2        800 0.500000 0.3333333           0         1.00
## 45     glmnet        900 0.500000 0.3333333           0         1.00
## 46        knn        900 0.500000 0.3333333           0         1.00
## 47 svmLinear2        900 0.500000 0.3333333           0         1.00
## 48     glmnet       1000 0.500000 0.3333333           0         1.00
## 49 svmLinear2       1000 0.500000 0.3333333           0         1.00
## 50        lda        100 0.343750 0.3333333           0         1.00
```

5. Model building with best model and number of genes

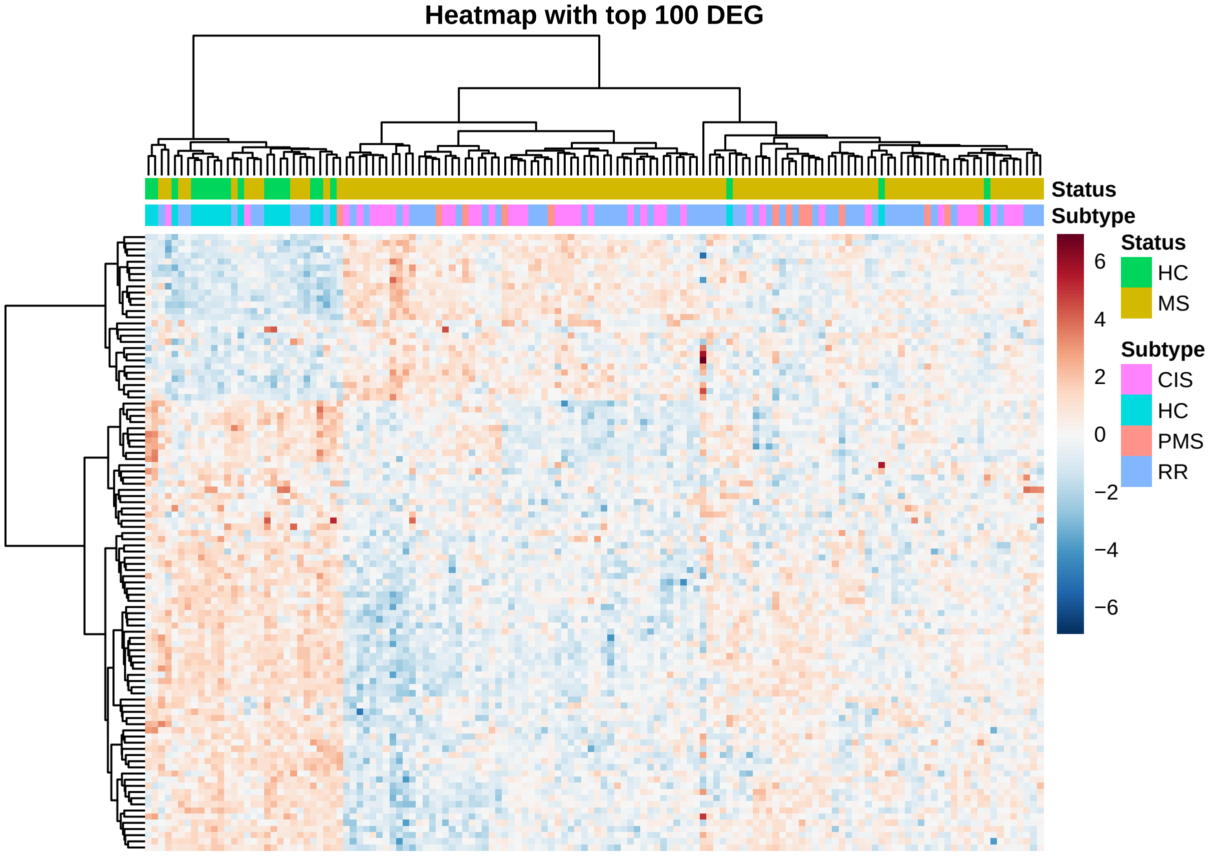
Based on previous test modeling, random forest method with 100 genes showed best performance. Therefore, I repeat modeling using random forest with 100 top statistically significant genes. In the current random forest run, I used 5-fold cross validation in the modeling because data size is small. The random split cannot reflect heterogeneity of human samples with small dataset.

Results

1. Results of differential expression analysis

In the differential expression analysis, I've found 1044 statistically significant genes (adjusted p-value < 0.1; 415 up-regulated and 629 down-regulated in MS). The heatmap shows top 100 differentially expressed genes (DEG). I observed separation between healthy controls and MS.

```
##
## out of 38280 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 415, 1.1%
## LFC < 0 (down)     : 629, 1.6%
## outliers [1]       : 44, 0.11%
## low counts [2]     : 17785, 46%
## (mean count < 4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
## [1] 100
```



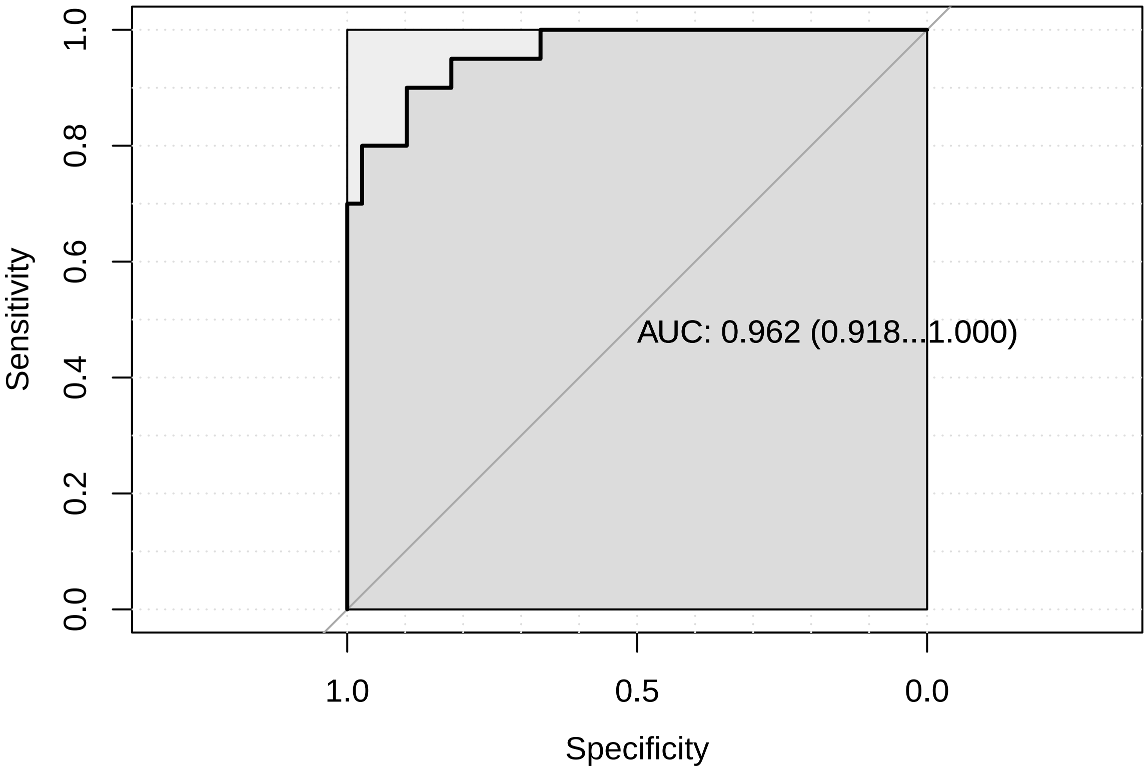
2. Results of random forest modeling

In the modeling using random forest with 100 genes, **accuracy was 0.8983051** and **AUC was 0.9615385**.

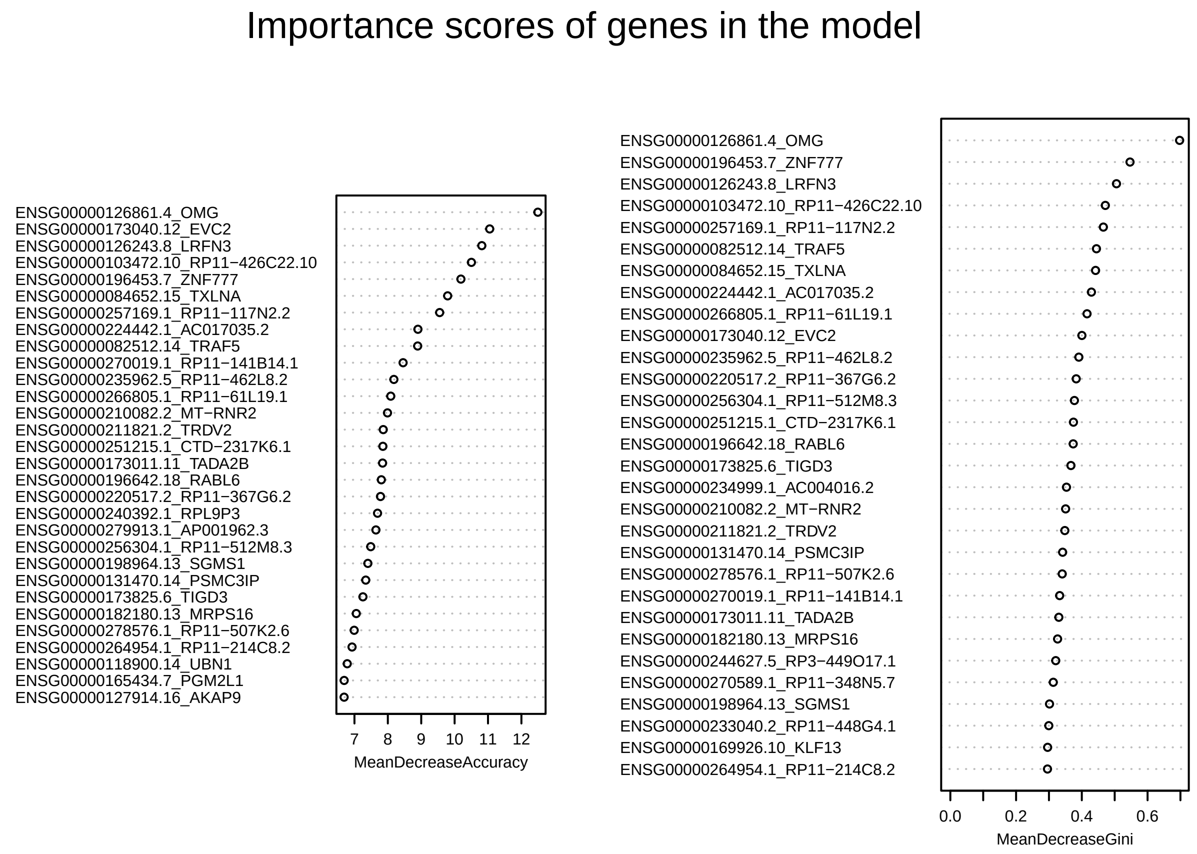
Results details are below:

```
##   Method NumAnalyte       AUC  Accuracy Sensitivity Specificity
## 1     RF        100 0.9615385 0.8983051   0.9487179         0.8
```

```
## function (data, ...)
## {
##     UseMethod("confusionMatrix")
## }
## <bytecode: 0x7fe2564798b8>
## <environment: namespace:caret>
```

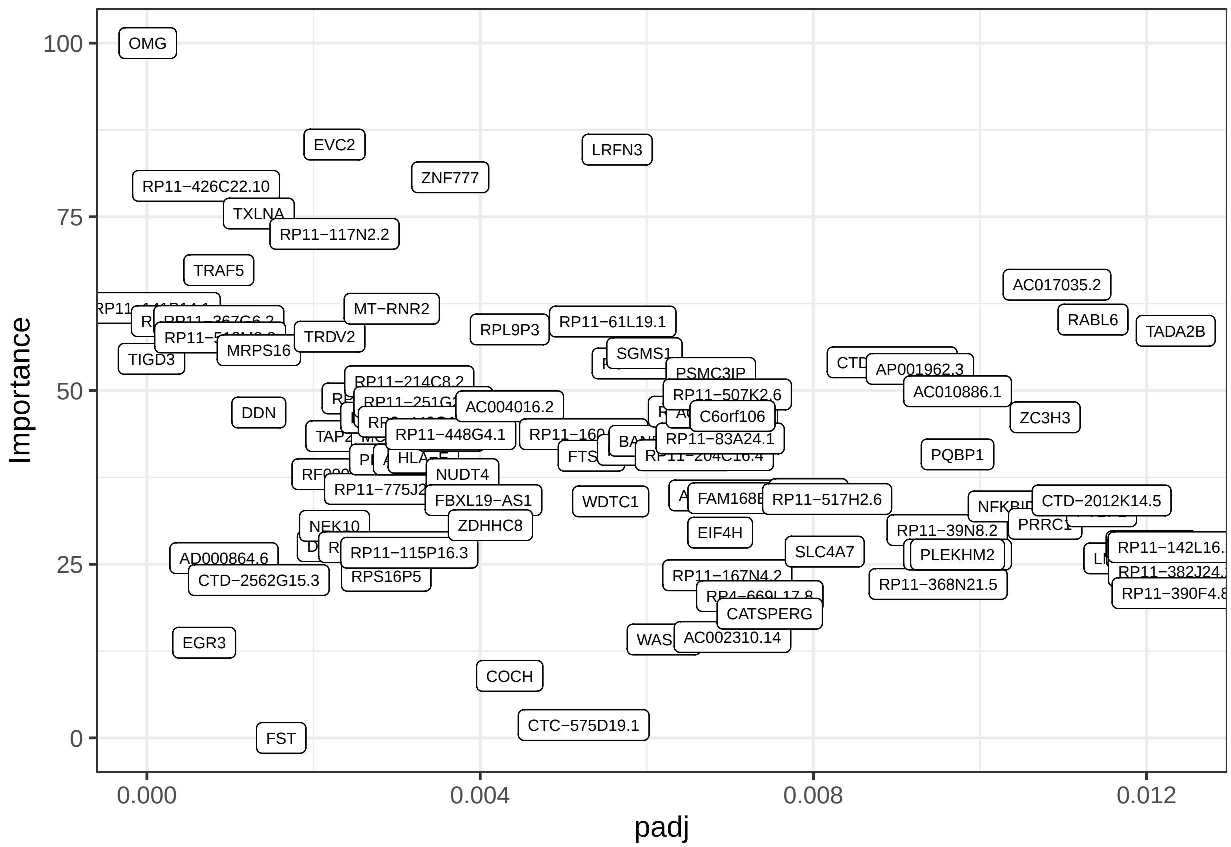


```
##
## Call:
## roc.default(response = pred_for_roc$obs, predictor = as.numeric(pred_for_roc$CIS),     ci = TRUE, plo
##
## Data: as.numeric(pred_for_roc$CIS) in 39 controls (pred_for_roc$obs CIS) > 20 cases (pred_for_roc$obs
## Area under the curve: 0.9615
## 95% CI: 0.9181-1 (DeLong)
```

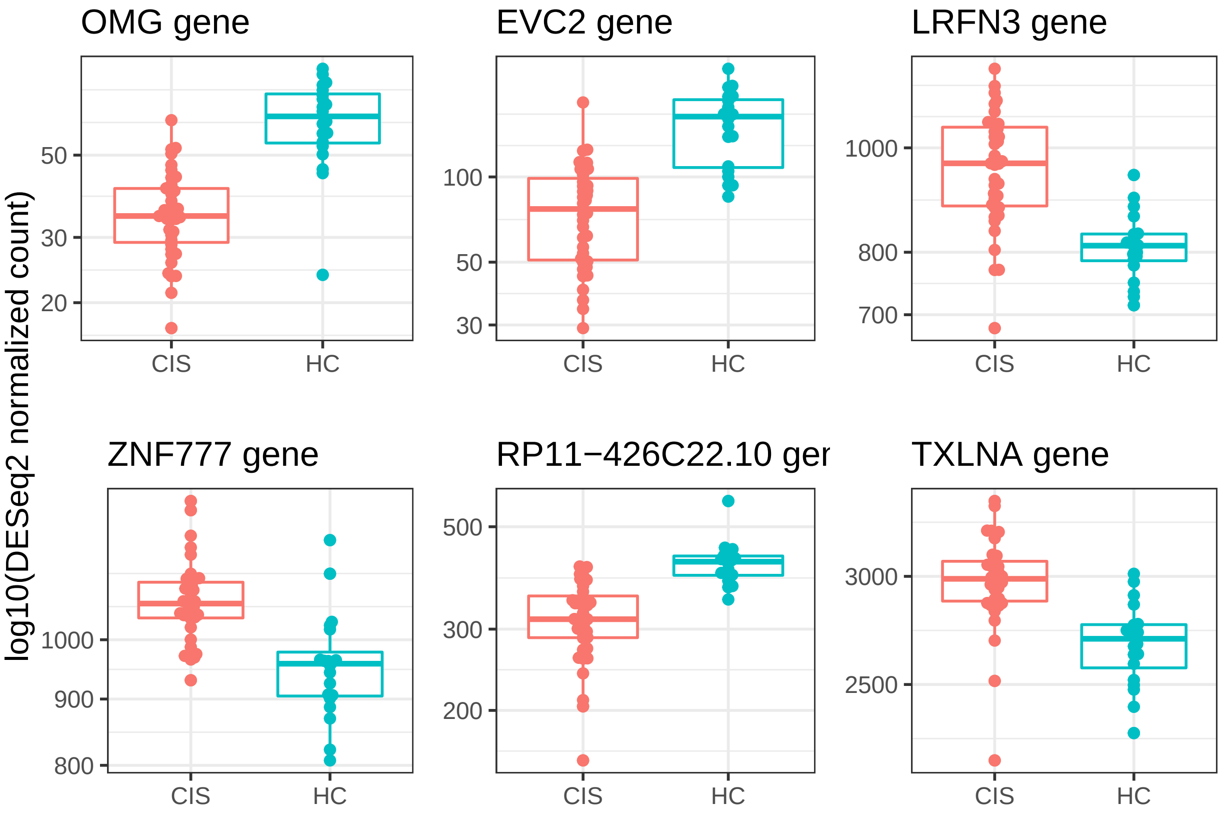


3. Comparison between differentially expressed genes (DEG) and random forest results

I compared between adjusted p-value from differential expresison analysis and imporance score from random forest. This plot indicates differentially expression result and random forest modeling are not consistent. Some genes are shows very low adjusted p-value from differential expression analysis but not well performed in random forest, or vice versa.



Last, these boxplots are examples for expression of top 6 gene that selected based on random forest importance score.



Conclusion

Multiple sclerosis (MS) is autoimmune condition of the central nervous system. In this study, I analyzed peripheral CD4+ T cells using differential expression analysis method and machine learning to identify early autoimmune response biomarker for MS. I found 1044 differential expressed genes in CIS and used statistically significant genes for model building using the random forest method. The final model achieved **AUC 0.9615385** and **accuracy 0.8983051** using 100 genes.

The genes identified in this study are potential early biomarker for multiple sclerosis that could detect neurological attack from blood. In addition, pathways associated with these genes in immune cells might be potential biomarker or therapeutic target for autoimmunity in CD4+ T cells in MS.

As shown in the PCA plot, the immune cells have strong gender effect. Therefore, I selected top differential expressed genes that tested with gender and age as a covariate in the current study. However, the performance could improve with adjustment of gender effect in the model building.