Can Gene Expression Data Identify Patients With Inflammatory Bowel Disease?

A Consulting Project for Math 6627 (3/3)

Benjamin Smith

28 March 2022

Contents

Introduction

(Quoted from SSC website.)

Inflammatory bowel disease (IBD), which is comprised of the two disease entities of Crohn's disease (CD) and ulcerative colitis (UC), is an incurable gastrointestinal illness that results in chronic inflammation. IBD greatly affects patients' quality of life. Approximately 1.5 million people have IBD in the United States and Canada, where the rates are among the highest in the world.

There are currently no biomarkers for IBD, which could help to identify better treatments and individualize patient care. Such biomarkers could also be used to facilitate the development of clinical trials involving new medications. Recently, genome-wide association studies (GWAS) have significantly advanced our understanding about the importance of genetic susceptibility in IBD. Studies have identified a total of 201 IBD loci (Liu et al. 2015). However, these loci have yielded only a handful of candidate genes which often have small contributory effect in IBD.

The aim of this case study is to construct classifiers for IBD using global gene expression data based on these candidate genes. The research questions are:

1. Can data features (i.e., variables or probesets or genes) be used to cluster individuals into three biological groups (i.e., healthy individuals, CD patients, UC patients)?
2. Can data features (i.e., variables, probesets or genes) predict the disease state of individuals from three biological groups (i.e., healthy individuals, CD patients, UC patients)?

The Data

The data available consists of two datasets:

(Quoted from SSC website.)

1. Global gene expression data: Burczynski et al. (2006) generated genome-wide gene expression profiles for 41 healthy individuals (note that the processed data includes only 41 individuals although the original study included 42 individuals), 59 CD patients, and 26 UC patients using Affymetrix HG-U133A human GeneChip array. The GeneChip include approximate 22,000 probesets (each gene may have multiple probesets). The expression level of each probeset in each individual was quantified using MAS 5.0 software (we downloaded the processed data from ArrayExpress: E-GEOD-3365).
2. IBD candidate genes: IBD candidate genes implicated in the 201 IBD associated loci were evaluated using GRAIL (Gene Relationships across Implicated Loci) and DAPPLE (Disease Association Protein-Protein Link Evaluator) software tools. A total of 225 unique genes (see Supplementary Table 9 of Liu et al. 2015) were identified, and 185 of these 225 genes are on the Affymetrix HG-U133A human GeneChip array. These 185 candidate genes include 309 probesets.

Due to lack of domain knowledge and the ambiguity of the dataset, only the first data set will be used for this analysis. After the global gene expression data shaped properly for the analysis, there is no missing data present. With this taken care of, lets proceed with answering the two research questions.

Analysis

Clustering Individuals Into Three Biological Groups

As is the case with the assignments in this practicum, the ability to thoroughly analyze the data under time constraint presents its challenges. On the SSC website it mentions that Liu, van Sommeren, Huang, et al found through association analyses which identify 38 susceptibility loci for inflammatory bowel disease and highlight shared genetic risk across populations. To explain that in terms for that people with a non-genetics background - the result found by Liu, van Sommeren, Huang, et al identifies 38 specific locations on an subjects chromosomes which highlight shared genetic risk across populations.

An approach to cluster the data into three groups can be done through principal component analysis. Interestingly enough, $\sim 80\%$ of the variance in the data can be explained by 38 principal components as shown in the R output below:

A way to view the effectiveness of such a model would be to use 10-fold cross validation and see the mean-square error of prediction produced. From figures 1 and 2 the MSE produced from PCA regression using 38 components produces a MSE of around 0.43 and the spread between predicted and measured groups is quite large. From this it can be seen that the 38 principal components are useful for explaining the variance in the data, however as a method for classifying individuals it is quite poor.

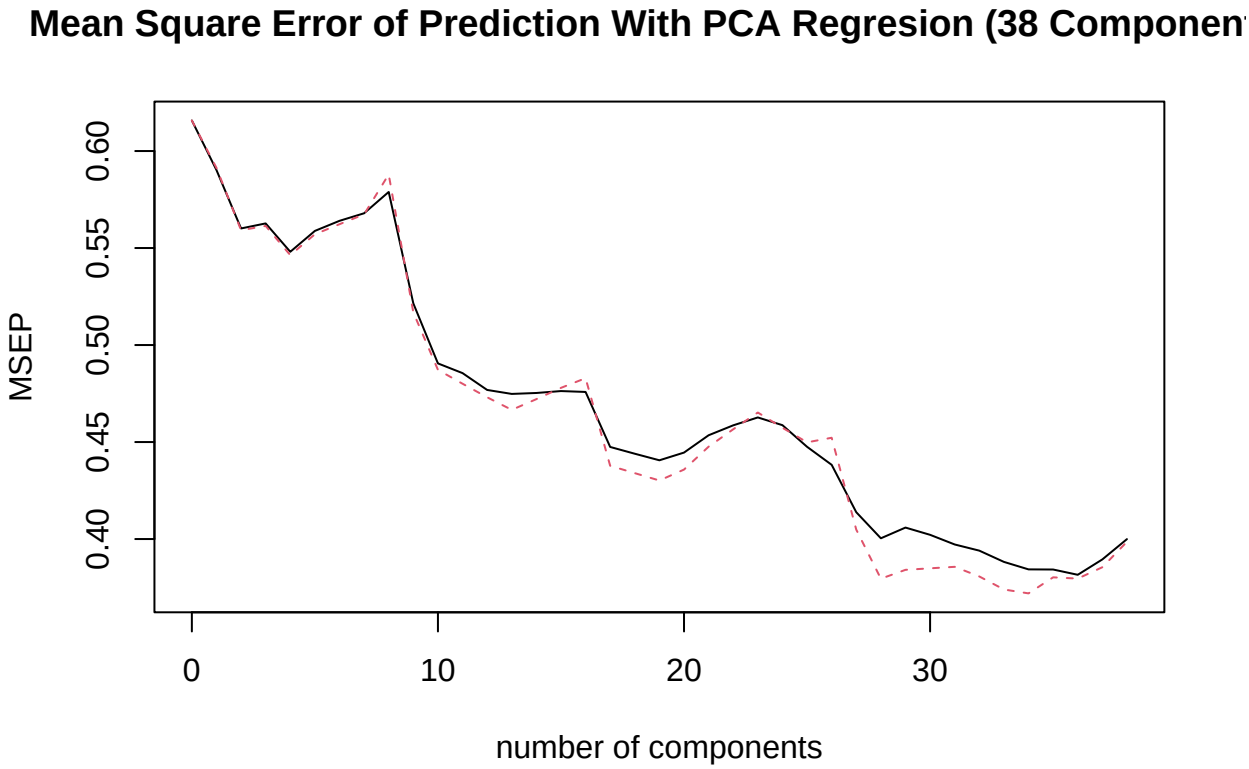


Figure 1: Mean Square Error of Prediction With PCA Regresion (38 Components)

While this method did prove to be a successful method for clustering data, it does demonstrate (and confirm) that the finding by Liu, van Sommeren, Huang, et al identifies loci which are susceptible to receiving the diseases in question, but are not useful as predictors in of themselves.

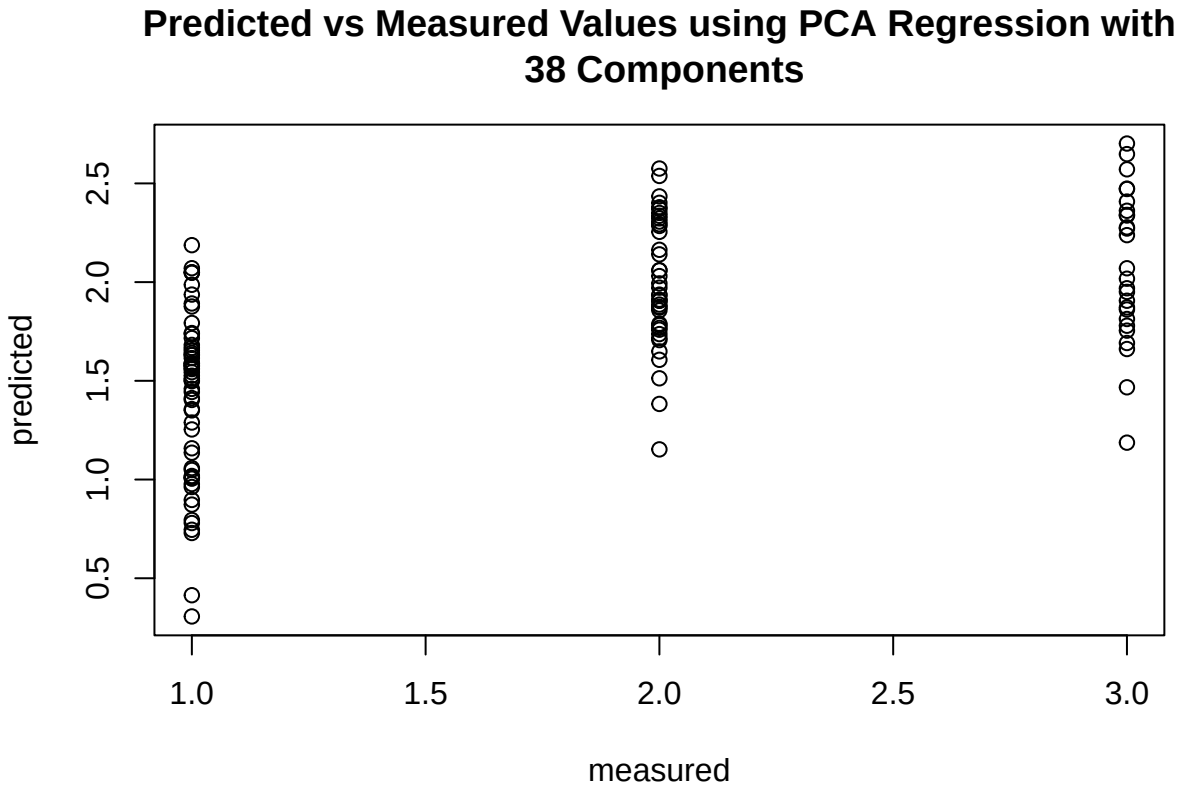


Figure 2: Predicted vs Measured Values using PCA Regression with 38 Components

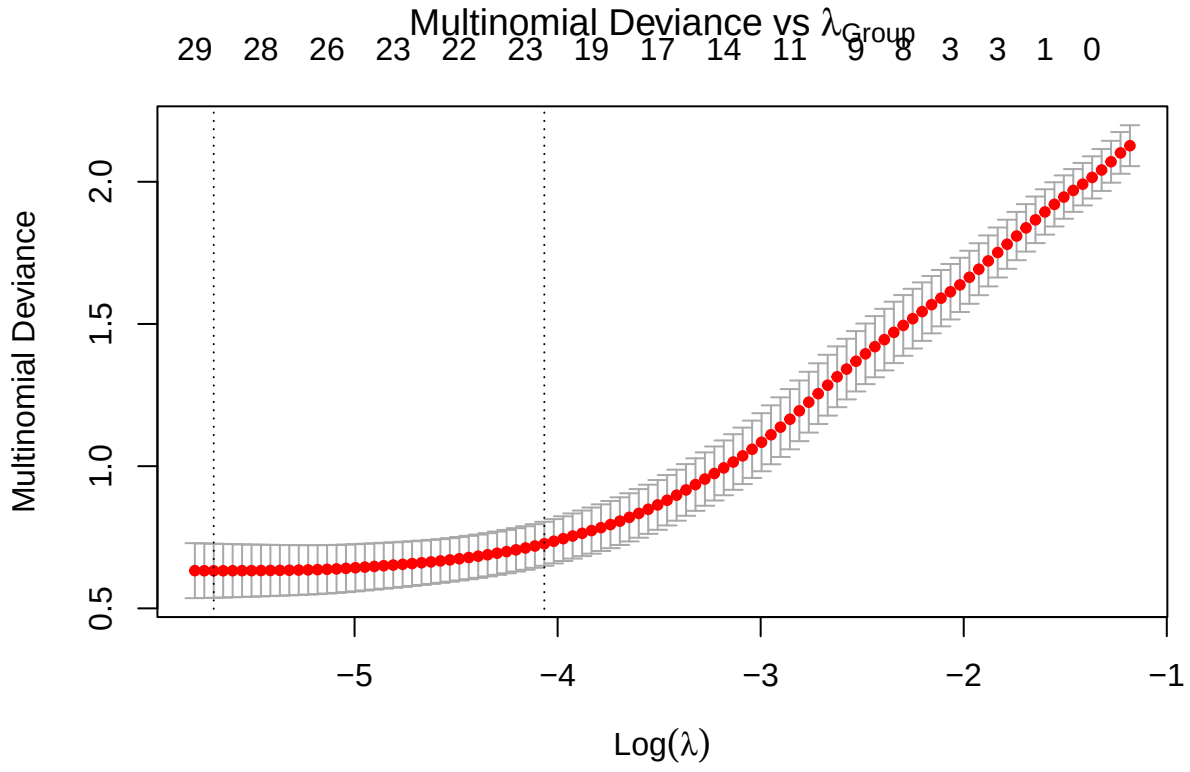
What Data Features Can Predict The Disease State From Three Biological Groups

With a focus on prediction, the use of multinomial regression with 10-fold cross validation is employed with a traditional machine learning approach where the data is split into training and testing sets with a 75%/25% training-testing split in the data.

```
set.seed(1)
dtt2 <- data.matrix(dtt)
y <- dtt2[,1]
X <- dtt2[,-1]

train = sample(seq(length(y)),75,replace=FALSE)
# 10 -fold crossvalidation
cv_model <- cv.glmnet(X, y, family='multinomial', alpha = 1)
best_lambda<- cv_model$lambda.min
```

```
plot(cv_model,main=expression("Multinomial Deviance vs "*lambda[Group]*""))
```




```
train_model <- glmnet(X[train,], y[train],family='multinomial', alpha = 1, lambda = best_lambda)

confusion.glmnet(train_model, newx = X[-train,], newy = y[-train])
```

```
##           True
```

```
## Predicted  1  2 3 Total
##     1     21  0 2    23
##     2      5 13 1    19
##     3      4  0 5     9
##     Total 30 13 8    51
##
##  Percent Correct:  0.7647
```

```
sparseEstimates<-data.frame(
       `Crohn's Disease`=train_model$beta$`1`[,],
       `Normal` =train_model$beta$`2`[,],
       `Ulcerative Colitis`=train_model$beta$`3`[,])

colnames(sparseEstimates)<-c("Crohn's Disease","Normal","Ulcerative Colitis")

sparseEstimates %>%
  filter(!(`Crohn's Disease`==0 &`Normal`==0 & `Ulcerative Colitis`==0)) %>%
  knitr::kable(caption = "Sparse Estimates for Multinomial Classification of Groups")
```

Table 1: Sparse Estimates for Multinomial Classification of Groups

	Crohn's Disease	Normal	Ulcerative Colitis
Age	-0.0097589	0.0000000	0.0000000
200930_s_at	0.0000000	0.0000000	-0.0448403
201783_s_at	0.0000000	0.0017043	0.0000000
202531_at	0.0000000	0.0000000	-0.0009233
202716_at	0.0000000	0.0000000	0.0200865
203236_s_at	-0.0104848	0.0000000	0.0006280
203320_at	0.0000000	0.0030645	0.0000000
204057_at	0.0000000	0.0067957	0.0000000
204213_at	-0.0035302	0.0000000	0.0000000
204417_at	0.0000000	0.0000000	0.0000171
204785_x_at	0.0019350	0.0000000	0.0000000
204902_s_at	0.0000000	0.0000000	-0.0211848
204906_at	0.0283967	0.0000000	0.0000000
205098_at	0.0000000	-0.0033583	0.0003681
205266_at	0.0349097	0.0000000	0.0000000
206035_at	0.0000000	0.0320196	0.0000000
206390_x_at	0.0018512	0.0000000	0.0000000
206828_at	0.0000000	0.0169354	0.0000000
207072_at	0.0000000	0.0018425	-0.0003017
207526_s_at	0.0000000	0.0000000	0.0587881
207535_s_at	0.0000000	0.0042939	0.0000000
207844_at	-0.0355289	0.0000000	0.0000000
208010_s_at	0.0000000	0.0000000	-0.0138359
208038_at	0.0085615	0.0000000	0.0000000
208304_at	0.0000000	0.0000000	0.0227305
208621_s_at	0.0000000	0.0000000	-0.0084096
209544_at	0.0172310	0.0000000	0.0000000
209545_s_at	0.0000000	0.0000000	-0.0024219
209575_at	0.0000000	-0.0072154	0.0000000
209664_x_at	0.0000000	0.0000000	0.0173999

	Crohn's Disease	Normal	Ulcerative Colitis
209782_s_at	0.0000000	0.0137451	0.0000000
209967_s_at	0.0000000	0.0000000	0.0063665
210133_at	0.0274815	0.0000000	0.0000000
210422_x_at	0.0000000	0.0000000	0.0016145
211153_s_at	0.0263287	0.0000000	0.0000000
211639_x_at	0.0000000	0.0000000	0.0333122
211856_x_at	-0.0120901	0.0000000	0.0000000
212501_at	0.0000000	0.0000000	0.0002571
212587_s_at	0.0000000	0.0000000	-0.0018010
212588_at	0.0000000	-0.0004283	0.0000000
212668_at	0.0000000	0.0426836	0.0000000
212912_at	0.0000000	-0.0154505	0.0000000
214228_x_at	0.0000000	0.0000000	-0.0038817
214467_at	0.0076350	0.0000000	0.0000000
215050_x_at	0.0000000	0.0000000	-0.0208561
215346_at	-0.0019466	0.0000000	0.0000000
216734_s_at	0.0000000	0.0000000	-0.0782045
216889_s_at	0.0000000	0.0204305	0.0000000
216986_s_at	0.0000000	-0.0429206	0.0000000
216987_at	-0.0172659	0.0000000	0.0000000
217473_x_at	0.0019883	0.0000000	0.0000000
217689_at	0.0000000	0.0000000	-0.0067085
218648_at	0.0000000	0.0000000	0.0026076
219209_at	0.0000000	0.0158086	-0.0117014
220704_at	0.0000000	0.0596789	0.0000000
221092_at	0.0000000	0.0000000	-0.0486084
221111_at	0.0000000	0.0000000	-0.0634202
221331_x_at	-0.0876114	0.0138713	0.0000000
221690_s_at	0.0000000	0.0058926	0.0000000
222292_at	0.0000000	-0.0247665	0.0000000

Conclusion

References

1. Statistical Society of Canada, Can Gene Expression Data Identify Patients With Inflammatory Bowel Disease? (2017). https://ssc.ca/en/meeting/annual/2017/case-study-2

2. Liu JZ, van Sommeren S, Huang H, et al. Association analyses identify 38 susceptibility loci for inflammatory bowel disease and highlight shared genetic risk across populations. Nat Genet. 47(9):979-86 (2015).

3. Ron LP, Natalie CT, Krystyna AZ, et al. Molecular classification of Crohn's disease and ulcerative colitis patients using transcriptional profiles in peripheral blood mononuclear cells. Michael E Burczynski, J Mol Diagn 8(1):51-61 (2006).

4. Ambroise C, McLachlan GJ. Selection bias in gene extraction on the basis of microarray gene-expression data. Proc Natl Acad Sci USA. 99(10):6562-6 (2002).

5. Dupuy A, Simon RM. Critical review of published microarray studies for cancer outcome and guidelines on statistical analysis and reporting. J Natl Cancer Inst. 99(2):147-57 (2007).

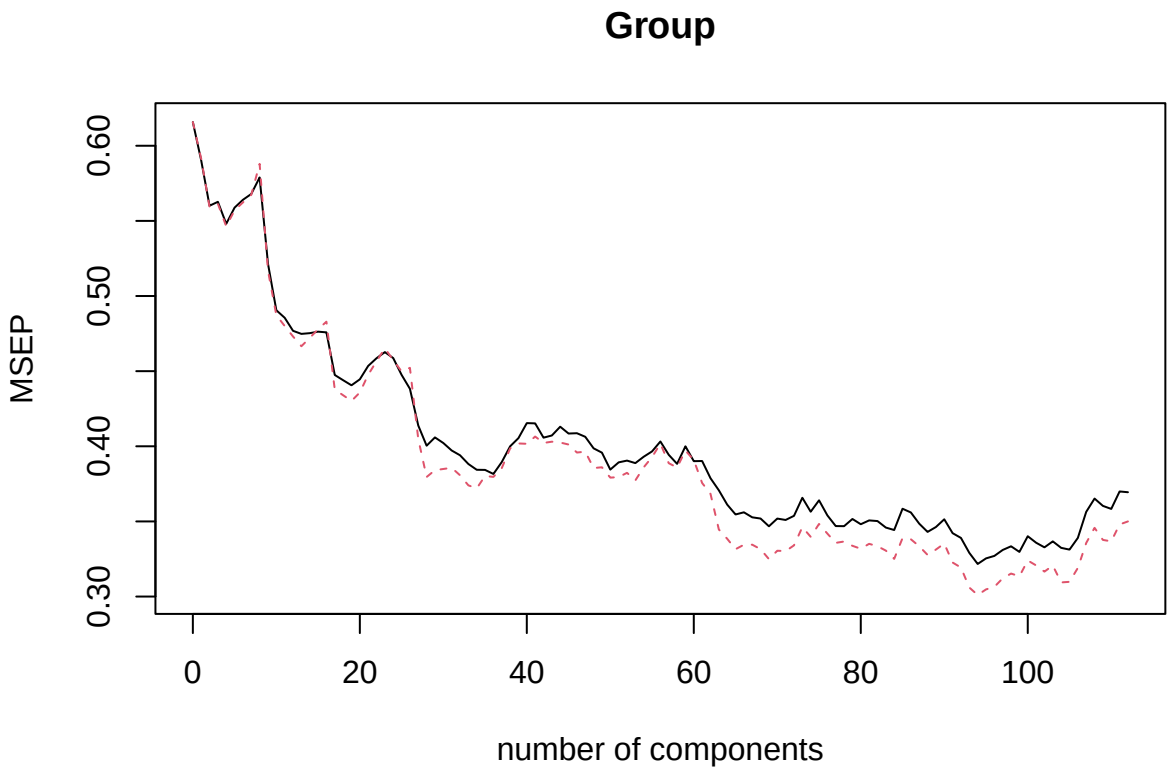
6. Cross-Validated, How to use R prcomp results for prediction? https://stats.stackexchange.com/questions/72839/how-to-use-r-prcomp-results-for-prediction

Code Appendix

```
library(pls)
library(caret)
set.seed(1)

pcr_model <- pcr(Group~ .,
                 data = data.matrix(dtt) %>% as.data.frame.matrix(),
                 scale = TRUE,
                 validation = "CV")

validationplot(pcr_model, val.type="MSEP")
```



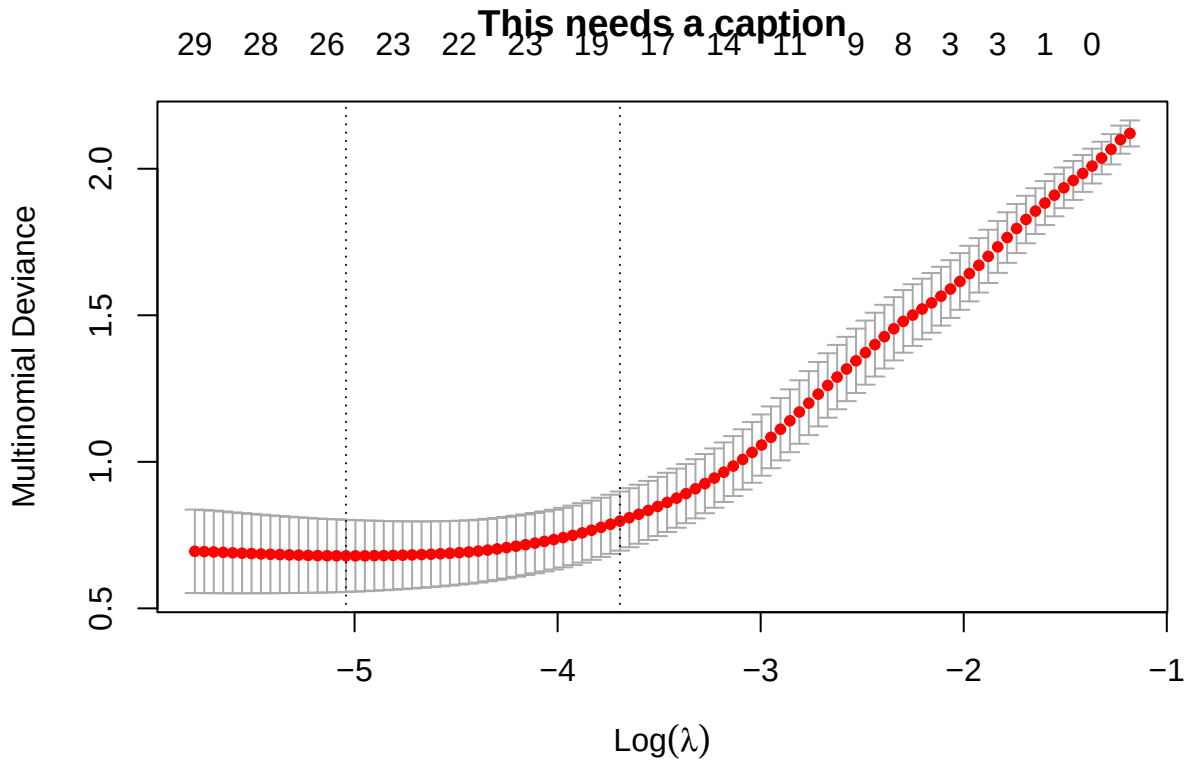
What Data Features Can Predict The Disease State From Three Biological Groups

[MULTINOMIAL REGRESSION]

```
dtt2 <- data.matrix(dtt)
y <- dtt2[,1]
X <- dtt2[,-1]

# 10 -fold crossvalidation
cv_model <- cv.glmnet(X, y, family='multinomial', alpha = 1)
best_lambda<- cv_model$lambda.min
```

```
plot(cv_model,main="This needs a caption")
```




```
best_model <- glmnet(X, y, alpha = 1, lambda = best_lambda)


sparseEstimates<- as.matrix(coef(best_model),rownames) %>%
                  as.data.frame.matrix() %>%
                  filter(s0!=0)

sparseEstimates[order(-abs(sparseEstimates$s0)),,drop=FALSE] %>%  knitr::kable(caption="Sparse Estimates
```

Table 2: Sparse Estimates

	s0
(Intercept)	2.5349003

	s0
209857_s_at	0.0309373
212466_at	-0.0298132
207257_at	-0.0238532
207533_at	-0.0196933
221092_at	-0.0193123
206211_at	0.0154410
Ethnicity	-0.0144825
221111_at	-0.0114593
210145_at	-0.0099183
221085_at	0.0098706
204906_at	-0.0095116
216901_s_at	0.0094640
211269_s_at	0.0082719
214832_at	-0.0080394
203650_at	-0.0073348
216889_s_at	-0.0072014
Age	0.0069898
215050_x_at	-0.0069519
207526_s_at	0.0066874
205455_at	0.0061097
215458_s_at	-0.0057218
210643_at	0.0056785
206336_at	-0.0050236
211639_x_at	0.0049945
207433_at	-0.0045661
214228_x_at	-0.0039927
210354_at	0.0039901
207535_s_at	0.0033583
207952_at	-0.0032641
217299_s_at	0.0031490
204896_s_at	-0.0031414
212458_at	0.0030800
209544_at	-0.0030495
209664_x_at	0.0029548
211105_s_at	-0.0028953
206419_at	0.0028479
203236_s_at	0.0027602
213137_s_at	-0.0026552
202820_at	-0.0025196
207538_at	0.0025044
210162_s_at	-0.0024579
211856_x_at	0.0024395
210001_s_at	-0.0023764
212666_at	0.0023494
208621_s_at	-0.0022490
212486_s_at	0.0018416
206983_at	0.0018168
205469_s_at	-0.0016613
208602_x_at	0.0014153
220704_at	-0.0013198
202716_at	0.0012859
214467_at	-0.0012571

	s0
209782_s_at	0.0011523
213450_s_at	0.0011173
207375_s_at	0.0010452
207844_at	0.0010149
201460_at	0.0009740
208351_s_at	-0.0009457
207630_s_at	0.0009296
221331_x_at	0.0009143
207536_s_at	-0.0009112
200930_s_at	0.0008746
220320_at	0.0008683
202682_s_at	0.0008160
208304_at	0.0007768
204362_at	-0.0006803
206072_at	-0.0006174
209967_s_at	0.0005610
203717_at	-0.0004903
211861_x_at	-0.0004815
204563_at	0.0004261
201095_at	-0.0003756
207238_s_at	-0.0003754
205842_s_at	0.0003317
207072_at	-0.0003197
206246_at	-0.0003074
212195_at	0.0002970
202018_s_at	-0.0002837
215346_at	0.0002626
208196_x_at	0.0002146
202531_at	-0.0002007
211939_x_at	0.0001497
200931_s_at	-0.0001302
200704_at	-0.0001232
206485_at	0.0001041
208075_s_at	0.0000972
206390_x_at	-0.0000967
203320_at	0.0000911
210422_x_at	0.0000869
202681_at	-0.0000802
216986_s_at	0.0000801
210423_s_at	-0.0000768
209545_s_at	-0.0000753
212501_at	0.0000734
201041_s_at	-0.0000697
204420_at	-0.0000622
205039_s_at	0.0000548
217916_s_at	-0.0000484
203111_s_at	0.0000449
202644_s_at	-0.0000371
213136_at	-0.0000040

```
library(mclogit)
```

```
## Warning: package 'mclogit' was built under R version 4.1.3
```

```
library(nnet)
library(memisc)
```

```
## Warning: package 'memisc' was built under R version 4.1.3

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

##
## Attaching package: 'memisc'

## The following object is masked from 'package:Matrix':
##
##     as.array

## The following objects are masked from 'package:dplyr':
##
##     collect, recode, rename, syms

## The following object is masked from 'package:purrr':
##
##     %@%

## The following object is masked from 'package:tibble':
##
##     view

## The following object is masked from 'package:ggplot2':
##
##     syms

## The following objects are masked from 'package:stats':
##
##     contr.sum, contr.treatment, contrasts

## The following object is masked from 'package:base':
##
##     as.array
```

```r
mod <- multinom(Group ~ .,data=dtt)
```

```
## # weights:  951 (632 variable)
## initial  value 138.425148
## iter  10 value 28.983852
## iter  20 value 15.715806
## iter  30 value 13.291647
## iter  40 value 11.443507
## iter  50 value 10.861357
## iter  60 value 10.409941
## iter  70 value 10.140307
## iter  80 value 10.041742
## iter  90 value 9.977845
## iter 100 value 9.938310
## final  value 9.938310
## stopped after 100 iterations
```

Conclusion

References

1. Statistical Society of Canada, Can Gene Expression Data Identify Patients With Inflammatory Bowel Disease? (2017). https://ssc.ca/en/meeting/annual/2017/case-study-2

2. Liu JZ, van Sommeren S, Huang H, et al. Association analyses identify 38 susceptibility loci for inflammatory bowel disease and highlight shared genetic risk across populations. Nat Genet. 47(9):979-86 (2015).

3. Ron LP, Natalie CT, Krystyna AZ, et al. Molecular classification of Crohn's disease and ulcerative colitis patients using transcriptional profiles in peripheral blood mononuclear cells. Michael E Burczynski, J Mol Diagn 8(1):51-61 (2006).

4. Ambroise C, McLachlan GJ. Selection bias in gene extraction on the basis of microarray gene-expression data. Proc Natl Acad Sci USA. 99(10):6562-6 (2002).

5. Dupuy A, Simon RM. Critical review of published microarray studies for cancer outcome and guidelines on statistical analysis and reporting. J Natl Cancer Inst. 99(2):147-57 (2007).

Code Appendix