

Preliminary Investigation

Ethan Allavarpu with Andy's Edits

10/27/2020

```
set.seed(110920)
library(tidyverse)
```

```
-- Attaching packages -----
v ggplot2 3.3.2      v purrr  0.3.4
v tibble  3.0.3      v dplyr  1.0.2
v tidyr   1.1.2      v stringr 1.4.0
v readr   1.3.1      v forcats 0.5.0
```

```
-- Conflicts -----
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
# sample <- read.csv("sample.csv", stringsAsFactors = TRUE)
# sample
```

```
training <- read.csv("training.csv", stringsAsFactors = TRUE)
sort(abs(cor(training)[ "class", ]), decreasing = TRUE)[2:19]
```

Broad_H4K20me1_percentage	Broad_H3K9ac_percentage	H4K20me1_width
0.5309561	0.4810581	0.4789003
BioGRID_log_degree	H3K79me2_height	H3K79me2_width
0.4764364	0.4719210	0.4709266
Broad_H3K4me2_percentage	Broad_H3K27ac_percentage	H4K20me1_height
0.4693101	0.4641367	0.4595187
Broad_H3K4me1_percentage	Broad_H3K79me2_percentage	Broad_H3K4me3_percentage
0.4580446	0.4571273	0.4315878
Broad_H3K36me3_percentage	H3K4me1_width	H3K36me3_width
0.4306293	0.4287817	0.4275987
pLOF_Zscore	N_LOF	H3K4me2_width
0.4227907	0.4212450	0.4080466

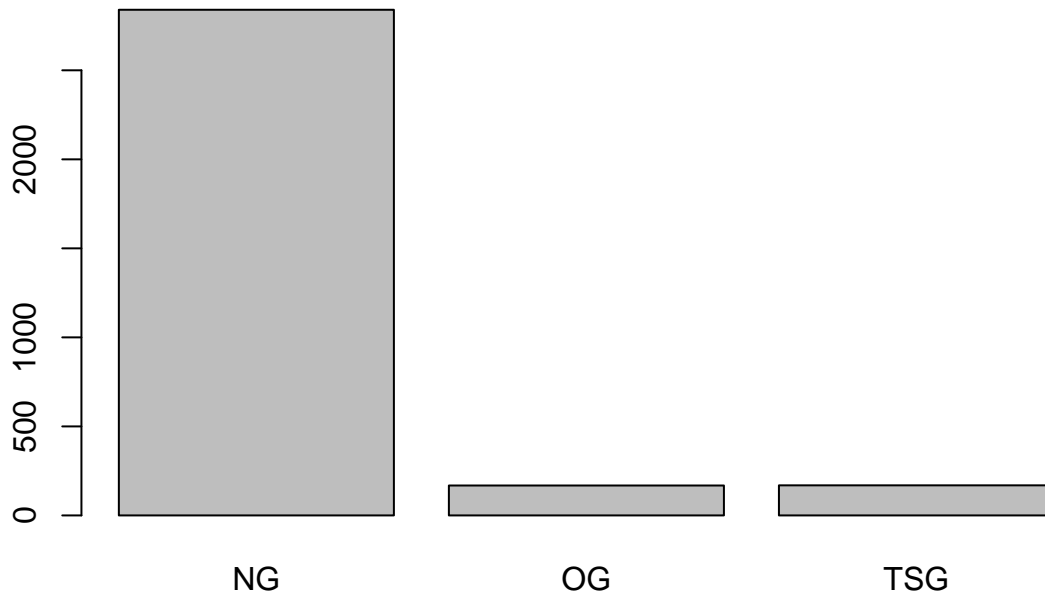
```
training$class <- factor(training$class)
levels(training$class) <- c("NG", "OG", "TSG")
dim(training)
```

```
[1] 3177 99
```

```
names(training)[c(1, 99)]
```

```
[1] "id" "class"
```

```
barplot(table(training$class))
```



```
table(training$class) / nrow(training)
```

```
      NG      OG      TSG
0.89392509 0.05288008 0.05319484
```

```
any(is.na(training))
```

```
[1] FALSE
```

```
# library(ggplot2)
# scatter <- function(var) {
#   ggplot(training, aes_string(var, "class")) +
#     geom_jitter(width = 0.05, height = 0.1, size = 0.1,
#               colour = rgb(0, 0, 0, alpha = 1 / 3))
# }
# scat_plot <- lapply(names(training)[-99], scatter)
# library(gridExtra)
# grid.arrange(grobs = scat_plot[1:20], ncol = 4)
# grid.arrange(grobs = scat_plot[21:40], ncol = 4)
# grid.arrange(grobs = scat_plot[41:60], ncol = 4)
# grid.arrange(grobs = scat_plot[61:80], ncol = 4)
# grid.arrange(grobs = scat_plot[81:98], ncol = 4)
```

```
sig <- logical(98)
names(sig) <- names(training)[-99]
k <- 1
diffs <- logical(98)
for (var in names(training)[-99]) {
  model <- aov(training[[var]] ~ factor(training$class))
  sig[k] <- summary(model)[[1]][1, 5]
  diffs[k] <- all(TukeyHSD(model)$`factor(training$class)`[, 4] < 0.05)
  k <- k + 1
}
head(sort(sig[diffs]), 15)
```

```
Broad_H4K20me1_percentage    Broad_H3K9ac_percentage    H4K20me1_width
```

2.605382e-232	8.993961e-184	3.222855e-181
H3K79me2_height	H3K79me2_width	Broad_H3K4me2_percentage
4.253756e-176	2.449130e-175	4.311054e-174
Broad_H3K79me2_percentage	Broad_H3K27ac_percentage	H4K20me1_height
1.188924e-168	2.345807e-168	1.791736e-164
Broad_H3K4me1_percentage	Broad_H3K36me3_percentage	H3K36me3_width
1.065370e-163	4.471080e-154	2.142472e-147
pLOF_Zscore	Broad_H3K4me3_percentage	H3K4me1_width
5.366468e-145	2.017738e-143	5.760242e-141

```
score <- function (conf_mat) {
  print(sum(diag(conf_mat) * c(1, 20, 20)))
  print(sum(diag(conf_mat) * c(1, 20, 20)) / sum(apply(conf_mat, 2, sum) * c(1, 20, 20)))
}
```

```
classify <- function(probs) {
  if (any(probs[2:3] > 0.05)) {
    subset <- probs[2:3]
    output <- which(subset == max(subset))
    if (length(output) > 1) {
      output <- sample(1:2, 1)
    }
  } else {
    output <- 0
  }
  output
}
```

```
vars <- training %>% dplyr::select(Broad_H3K9ac_percentage, N_LOF, pLOF_Zscore, Broad_H4K20me1_percentage,
                                H3K79me2_height,
                                N_Splice, LOF_T0_Total_Ratio, VEST_score, Missense_Entropy, BioGRID_log_deg)
vars$class <- factor(vars$class)
levels(vars$class) <- c("NG", "OG", "TSG")
cor_mtx = round(cor(vars[, names(vars) != "class"]), 2)
library(reshape2)
```

Attaching package: 'reshape2'

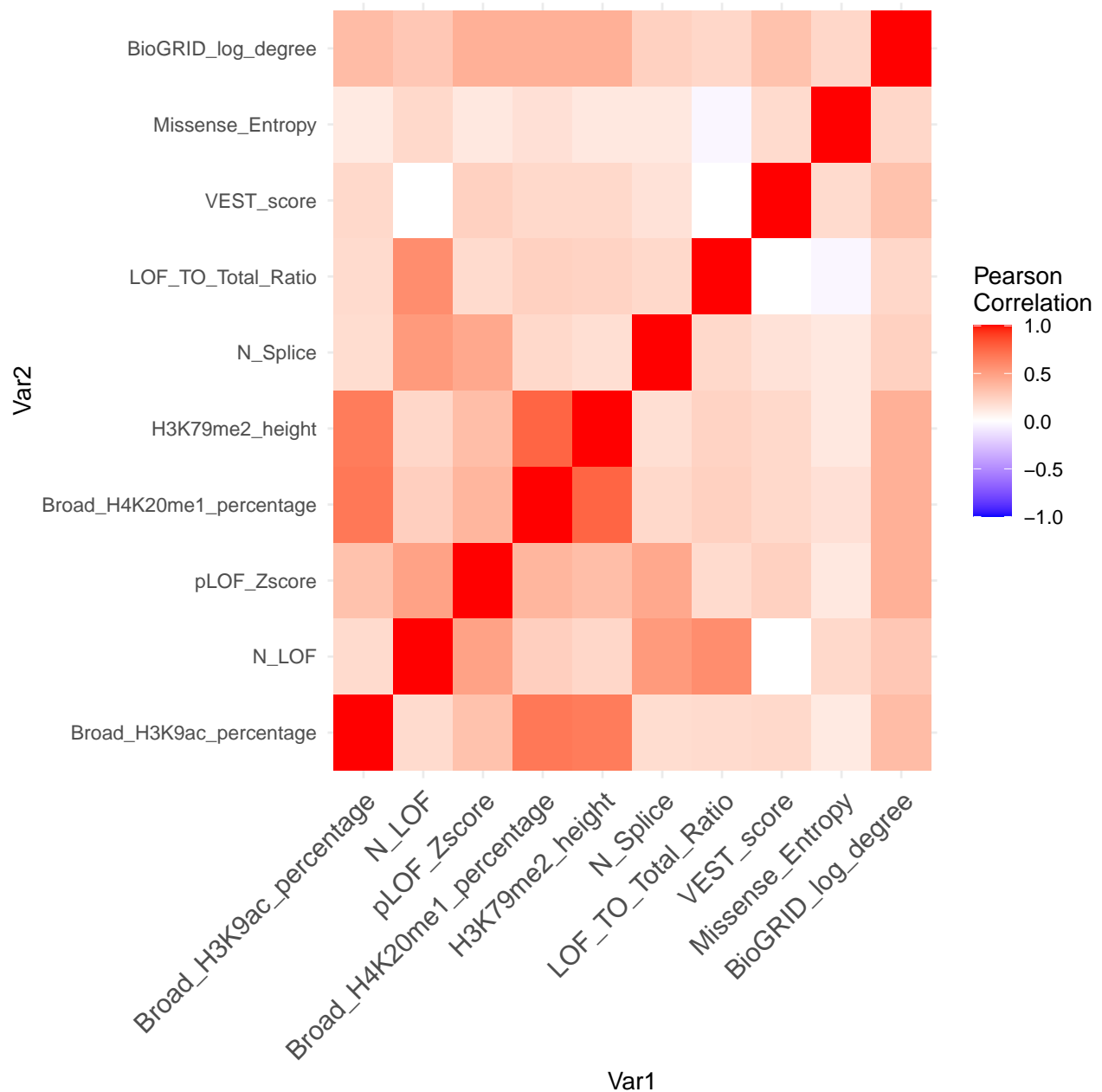
The following object is masked from 'package:tidyr':

smiths

```
#reshape it
melted_cor_mtx <- melt(cor_mtx)

#draw the heatmap
cor_heatmap = ggplot(data = melted_cor_mtx, aes(x=Var1, y=Var2, fill=value)) + geom_tile()
cor_heatmap = cor_heatmap +
scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1,1), space = "srgb") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1))

cor_heatmap
```



```
set.seed(nrow(training) + 421314)
library(caret)
```

Loading required package: lattice

Attaching package: 'caret'

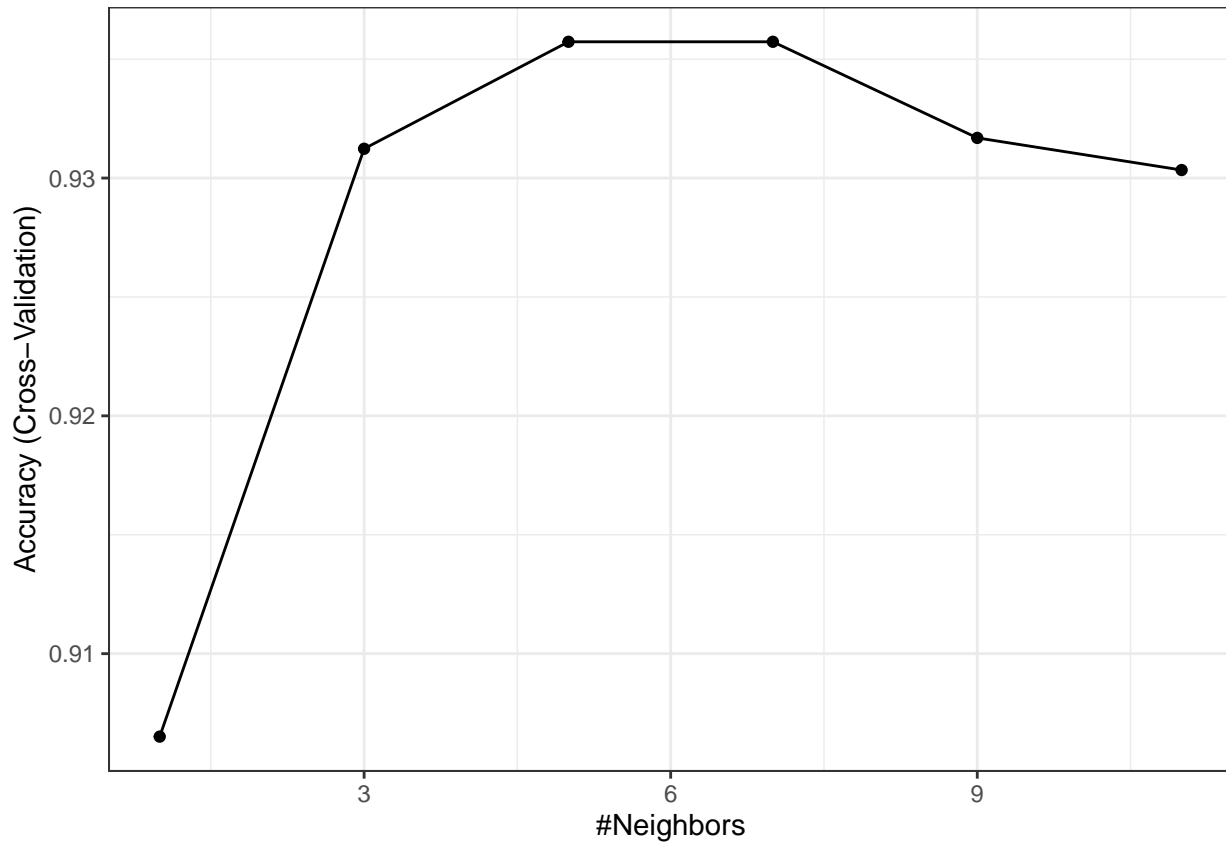
The following object is masked from 'package:purrr':

lift

```
vars_test <- createDataPartition(vars$class, p = 0.7,
                                list = FALSE)
vars_train <- vars[vars_test, ]
```

```
vars_test <- vars[-vars_test, ]
```

```
train_cont <- trainControl(method = "cv", number = 10, classProbs = TRUE, savePredictions = TRUE)
knn_ft <- train(class ~ ., data = vars_train, method = "knn", preProc = c("center", "scale"),
               trControl = train_cont, tuneGrid = expand.grid(k = seq(from = 1, to = 11, by = 2)))
ggplot(knn_ft) + theme_bw()
```



```
for (k in seq(from = 1, to = 11, by = 2)) {
  preds <- predict(knn_ft, newdata = vars_test, type = "prob")
  knn_mod <- table("pred"=unlist(apply(preds, 1, classify)), "obs" = vars_test$class)
  knn_mod
  score(knn_mod)
}
```

```
[1] 2063
[1] 0.723352
[1] 2083
[1] 0.7303647
[1] 2103
[1] 0.7373773
[1] 2103
[1] 0.7373773
[1] 2083
[1] 0.7303647
[1] 2083
[1] 0.7303647
```

```

train_cont <- trainControl(method = "cv", number = 10, classProbs = TRUE, savePredictions = TRUE)
knn_ft <- train(class ~ ., data = vars_train, method = "qda", preProc = c("center", "scale"),
               trControl = train_cont)
preds <- predict(knn_ft, newdata = vars_test, type = "prob")

qda_mod <- table("pred"=apply(preds, 1, classify), "obs" = vars_test$class)
qda_mod

```

	obs			
pred	NG	OG	TSG	
0	768	9	8	
1	16	28	3	
2	68	13	39	

```
score(qda_mod)
```

```

[1] 2108
[1] 0.7391304

```

```

train_cont <- trainControl(method = "cv", number = 10, classProbs = TRUE, savePredictions = TRUE)
knn_ft <- train(class ~ ., data = vars_train, method = "lda", preProc = c("center", "scale"),
               trControl = train_cont)
preds <- predict(knn_ft, newdata = vars_test, type = "prob")

lda_mod <- table("pred"=apply(preds, 1, classify), "obs" = vars_test$class)
lda_mod

```

	obs			
pred	NG	OG	TSG	
0	804	9	6	
1	19	30	7	
2	29	11	37	

```
score(lda_mod)
```

```

[1] 2144
[1] 0.7517532

```