Exploratory Analysis: Why are there so many zeroes???

Varan Nimar

```
#Data cleaning from Ethan's analysis

training <- read.csv("training.csv", stringsAsFactors = TRUE)
training$class <- factor(training$class)
levels(training$class) <- c("NG", "OG", "TSG")

outlier <- function(data) {
  low <- mean(data) - 3 * sd(data)
  high <- mean(data) + 3 * sd(data)
  which(data < low | data > high)
}
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)
outlier_index <- sort(table(unlist(lapply(training[,-99], outlier))), decreasing = TRUE)
outlier_index[1:100]
```

```
##
##  915 1280 2918  517 1914 2182 3052 1173 2215 3049  259  740 1749 1979 2998  417
##   24   24   24   22   22   22   20   19   19   19   18   18   18   18   18   17
##  441  806 2297  422  635 1258 1570 2278 2518 2729   80  150 2694  169  276  341
##   17   17   17   16   16   16   16   16   16   16   15   15   15   14   14   14
## 1528 1556 1726 1809 1911 1955 2071 2624 2641 3120 3142   73  277  364  751 1244
##   14   14   14   14   14   14   14   14   14   14   14   13   13   13   13   13
## 1330 2329 2787  343 1138 1171 1188 1372 1460 2031 2251 2968 2983 3166  352  634
##   13   13   13   12   12   12   12   12   12   12   12   12   12   12   11   11
##  907  923 1096 1858 2636  588 1137 1317 1463 1561 1740 1991 2487 2540 2555 2621
##   11   11   11   11   11   10   10   10   10   10   10   10   10   10   10   10
## 2815 3029   74  144  657  789  857 1267 1610 1932 2022 2093 2142 2534 2666 2721
##   10   10    9    9    9    9    9    9    9    9    9    9    9    9    9    9
## 2848 2900 3027  155
##    9    9    9    8
```

```
training <- training[-as.numeric(names(outlier_index)[1:50]),]
training <- training[-which(training$Missense_TO_Silent_Ratio > 100), ]
training <- training[-which(training$Missense_KB_Ratio > 2000), ]
training <- training[-which(training$LOF_TO_Silent_Ratio > 5), ]
training <- training[-which(training$Gene_expression_Z_score > 4), ]
training <- training[-which(training$dN_to_dS_ratio > 5),]
training <- training[-which(training$Silent_KB_Ratio > 200), ]
training <- training[-which(training$Lost_start_and_stop_fraction > 0.2),]
```

```
#Arguably easier to look at numbers max/min and see if anything stands out-- turns out it's not, just 0

# A LOT of observations have '0' for many variables. Is that meaningful? It could be.
#Let's see how many there are
numeric_training <- training[,-99]

n_zeroes <- rep(NA, nrow(numeric_training))

for(i in seq_len(nrow(numeric_training))){
  row_i_zeroes <- 0
  for(j in seq_len(ncol(numeric_training))){
    if(round(numeric_training[i,j], digits = 5) == 0){
      row_i_zeroes <- row_i_zeroes + 1
    }
  }
  n_zeroes[i] <- row_i_zeroes
}

head(sort(n_zeroes, decreasing = TRUE), n = 500)
```

```
##   [1] 88 88 85 82 81 81 81 80 79 77 77 75 74 74 71 68 68 67 67 67 66 66 65 64 64
##  [26] 64 64 63 63 62 62 62 62 62 61 61 61 61 60 59 59 59 58 58 58 57 57 57 57 57
##  [51] 57 57 57 57 56 56 56 56 55 55 55 55 55 55 54 53 53 52 52 52 52 52 52 51 51
##  [76] 51 51 51 51 50 50 50 49 49 49 49 49 49 49 48 48 48 48 48 48 48 48 47 47 47
## [101] 47 47 47 47 47 47 47 47 46 46 46 46 45 45 45 45 45 45 45 45 45 45 45 45 45
## [126] 45 44 44 44 44 44 44 44 44 44 44 44 43 43 43 43 43 43 43 43 43 43 43 43 42
## [151] 42 42 42 42 42 42 42 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 40 40 40
## [176] 40 40 40 40 40 40 40 40 40 40 40 40 40 40 39 39 39 39 39 39 39 39 39 39 39
## [201] 39 39 39 39 39 39 39 39 38 38 38 38 38 38 38 38 38 38 38 38 38 38 38 38 38
## [226] 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 36
## [251] 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 35 35
## [276] 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35
## [301] 35 35 35 35 35 35 35 35 35 35 35 35 35 35 34 34 34 34 34 34 34 34 34 34 34
## [326] 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34
## [351] 34 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33
## [376] 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 32 32 32 32
## [401] 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
## [426] 32 32 32 32 32 32 32 32 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
## [451] 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31
## [476] 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 30 30 30 30 30 30 30
```

```
tail(sort(n_zeroes, decreasing = TRUE), n = 500)
```

```
##   [1] 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
```

```
##  [26] 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
##  [51] 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
##  [76] 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 10
## [101] 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## [126] 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## [151] 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## [176] 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## [201] 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
## [226] 10 10 10 10 10 10 10 10 10 10 10 10 10  9  9  9  9  9  9  9  9  9  9  9  9
## [251]  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9
## [276]  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9
## [301]  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9
## [326]  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  8  8  8  8  8  8
## [351]  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8
## [376]  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8
## [401]  8  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
## [426]  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
## [451]  7  7  7  7  7  7  7  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
## [476]  6  6  6  6  6  6  6  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  4  4  4  4  3
```

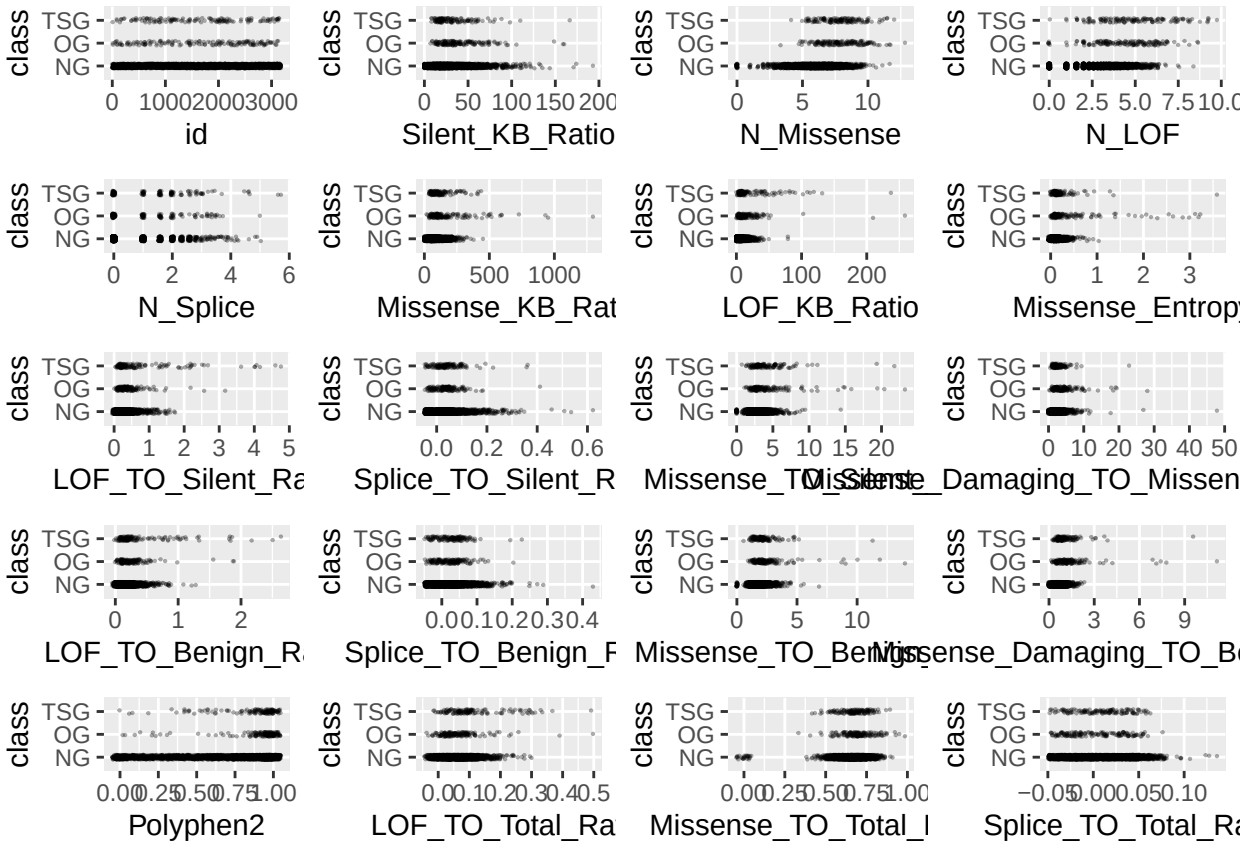
I think we should cross-reference which variables we're using for our final model, and see if there are any trends in genes that have all 0's for those variables. If not, maybe we can remove them?

Otherwise I think we should definitely test throwing out observations with 80+ zeroes and see how that affects our accuracy, could very well be dragging it down.

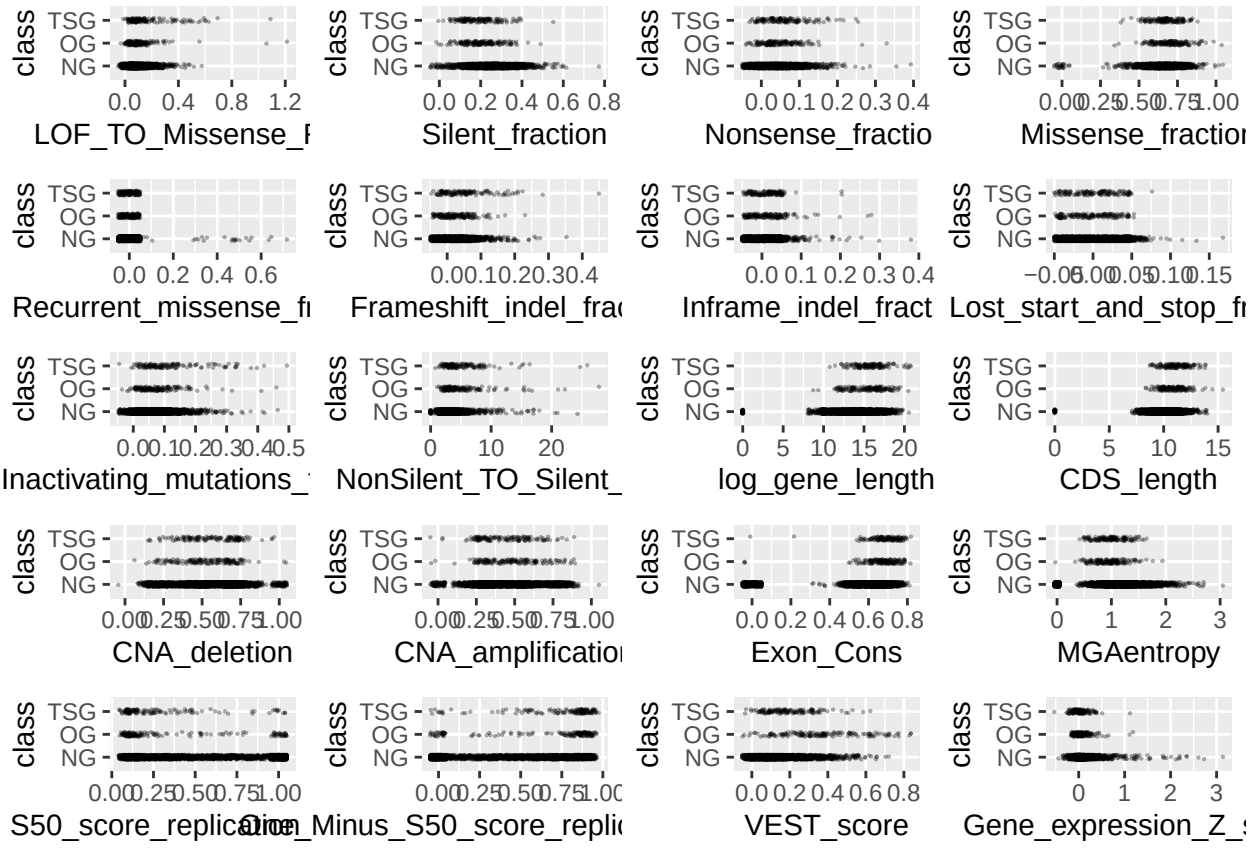
It's a bit difficult to tell what transformations would be appropriate, especially with so many zeroes. If we plan on a log transformation we'll have to account for that (possibly take `log(x+1)`). Or do a different kind of transformation (square root or something of the sort). Those seem like the popular ways of dealing with data skewed by zeroes.

I had some trouble getting Box Cox to work on our LDA model, but it might be worth keeping at it to check for the effectiveness of polynomial/log transforms, the latter of which we may have to check manually.

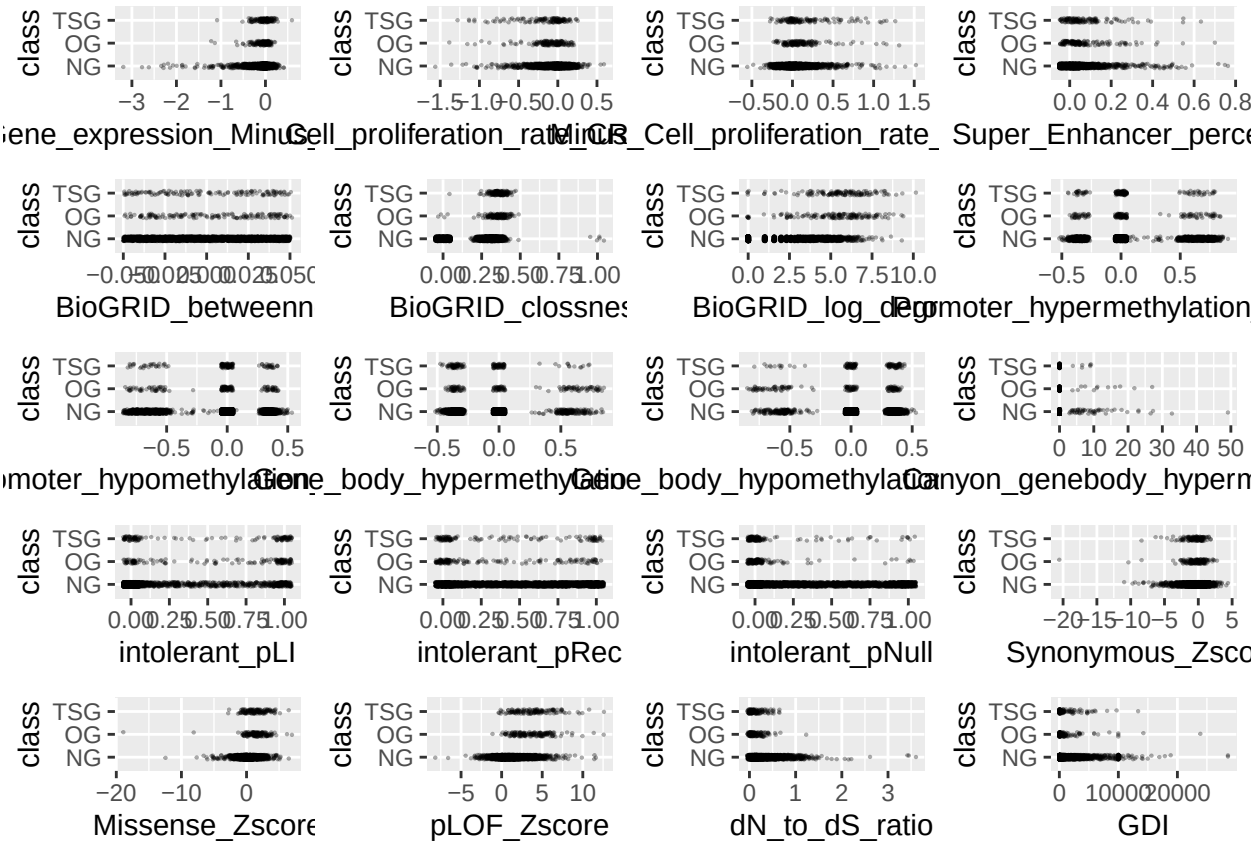
```
# Code graciously provided by Ethan Allavarpu
library(ggplot2)
scatter <- function(var) {
  ggplot(training, aes_string(var, "class")) +
    geom_jitter(width = 0.05, height = 0.1, size = 0.1,
                colour = rgb(0, 0, 0, alpha = 1 / 3))
}
scat_plot <- lapply(names(training)[-99], scatter)
library(gridExtra)
grid.arrange(grobs = scat_plot[1:20], ncol = 4)
```



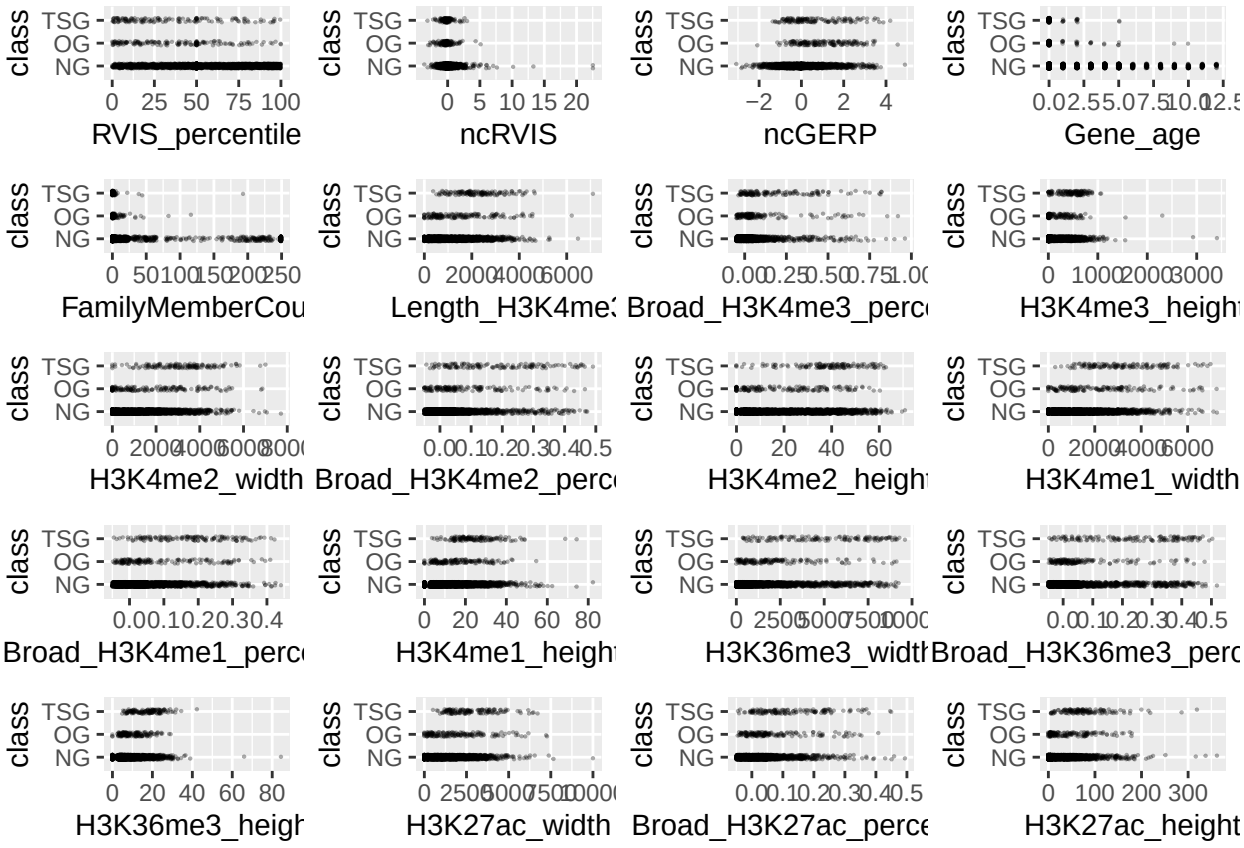

```
grid.arrange(grobs = scat_plot[21:40], ncol = 4)
```



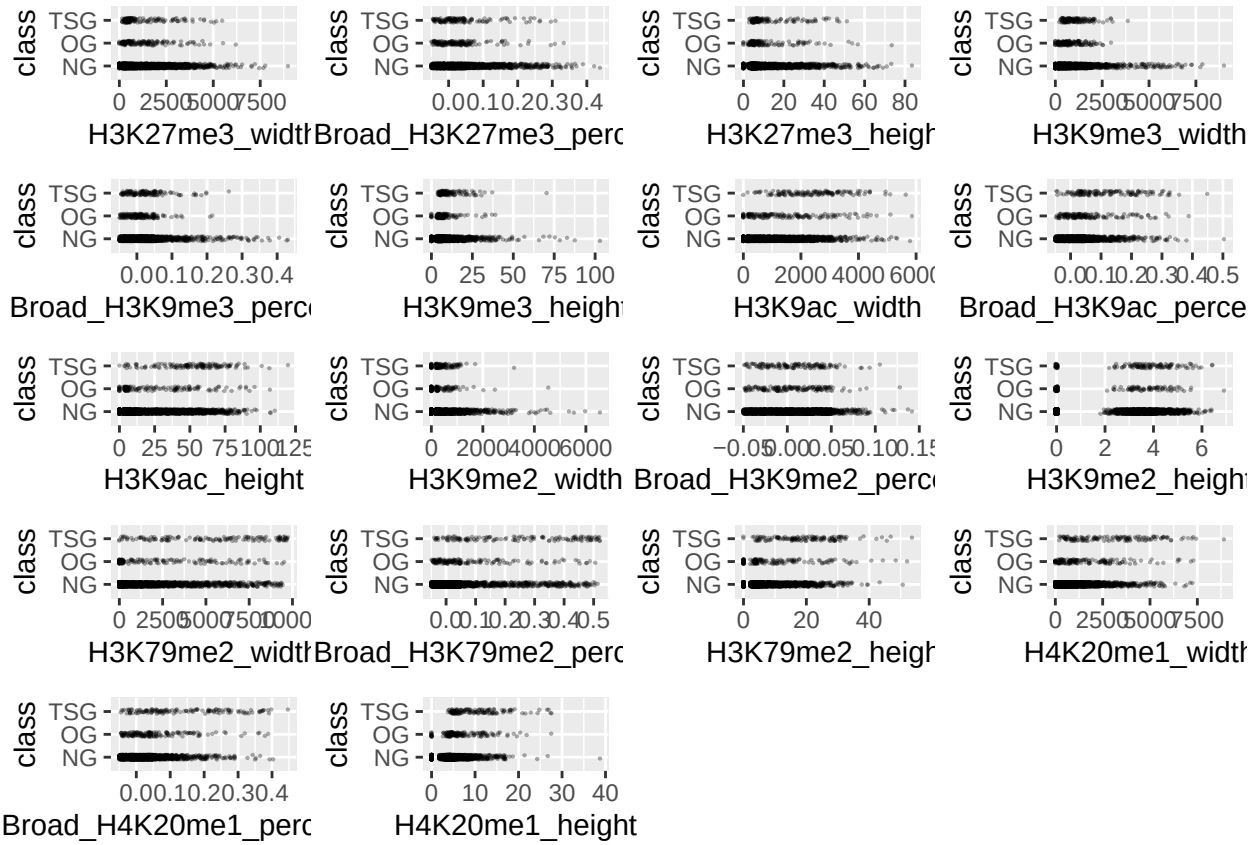

```
grid.arrange(grobs = scat_plot[41:60], ncol = 4)
```




```
grid.arrange(grobs = scat_plot[61:80], ncol = 4)
```




```
grid.arrange(grobs = scat_plot[81:98], ncol = 4)
```


class
TSG
OG
NG
H3K27me3_width
Broad_H3K27me3_perc
H3K27me3_heigh
H3K9me3_width
Broad_H3K9me3_perc
H3K9me3_height
H3K9ac_width
Broad_H3K9ac_perce
H3K9ac_height
H3K9me2_width
Broad_H3K9me2_perc
H3K9me2_height
H3K79me2_width
Broad_H3K79me2_perc
H3K79me2_heigh
H4K20me1_width
Broad_H4K20me1_perc
H4K20me1_height