Model 14 Predictions

Ethan Allavarpu (UID: 405287603)

11/02/2020

CREATE YOUR OWN COPY OF THE FILE IF YOU WANT TO CHANGE THINGS!!!!

Transforming and Cleaning the Data

```
training <- read.csv("training.csv", stringsAsFactors = TRUE)
training$class <- factor(training$class)
levels(training$class) <- c("NG", "OG", "TSG")
# Create an outlier function
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)
outliers <- table(unlist(lapply(training[,-99], outlier)))
outlier_index <- sort(outliers, decreasing = TRUE)
# Remove 50 most common outliers (i.e. outliers across multiple predictors)
training <- training[-as.numeric(names(outlier_index)[1:50]),]

# The below variables had outstanding outliers which we chose to remove
sort(training$Missense_TO_Silent_Ratio, decreasing = TRUE)[1:10]
```

```
 [1] 384.98658 172.91420 135.59623  71.09712  23.21809  21.81193  20.37791
 [8]  19.42402  19.38769  15.84808
```

```
training <- training[-which(training$Missense_TO_Silent_Ratio > 100), ]
sort(training$Missense_KB_Ratio, decreasing = TRUE)[1:10]
```

```
 [1] 2063.9413 1296.6625 1060.0601  952.3810  931.4227  726.8519  594.7603
```

```
 [8]  593.3610  581.5085  516.8084
```

```
training <- training[-which(training$Missense_KB_Ratio > 2000), ]
sort(training$LOF_TO_Silent_Ratio, decreasing = TRUE)[1:10]
```

```
 [1] 81.177835  9.030120  6.470238  5.582840  4.741460  4.558252  4.176630
 [8]  4.058140  4.039062  4.021930
```

```
training <- training[-which(training$LOF_TO_Silent_Ratio > 5), ]
sort(training$Gene_expression_Z_score, decreasing = TRUE)[1:10]
```

```
 [1] 19.720  9.210  7.080  6.883  6.590  6.280  5.321  5.316  3.161  2.767
```

```
training <- training[-which(training$Gene_expression_Z_score > 4), ]
sort(training$dN_to_dS_ratio, decreasing = TRUE)[1:10]
```

```
 [1] 20.950  3.649  3.446  3.372  2.574  2.194  2.183  2.102  1.921  1.744
```

```
training <- training[-which(training$dN_to_dS_ratio > 5),]
sort(training$Silent_KB_Ratio, decreasing = TRUE)[1:10]
```

```
 [1] 474.4745 193.1684 174.0558 171.0362 166.4971 160.2273 158.7697 148.5800
 [9] 143.6782 135.2657
```

```
training <- training[-which(training$Silent_KB_Ratio > 200), ]
sort(training$Lost_start_and_stop_fraction, decreasing = TRUE)[1:10]
```

```
 [1] 0.333 0.167 0.118 0.087 0.074 0.071 0.071 0.068 0.067 0.067
```

```
training <- training[-which(training$Lost_start_and_stop_fraction > 0.2),]
sort(training$Synonymous_Zscore, decreasing = FALSE)[1:10]
```

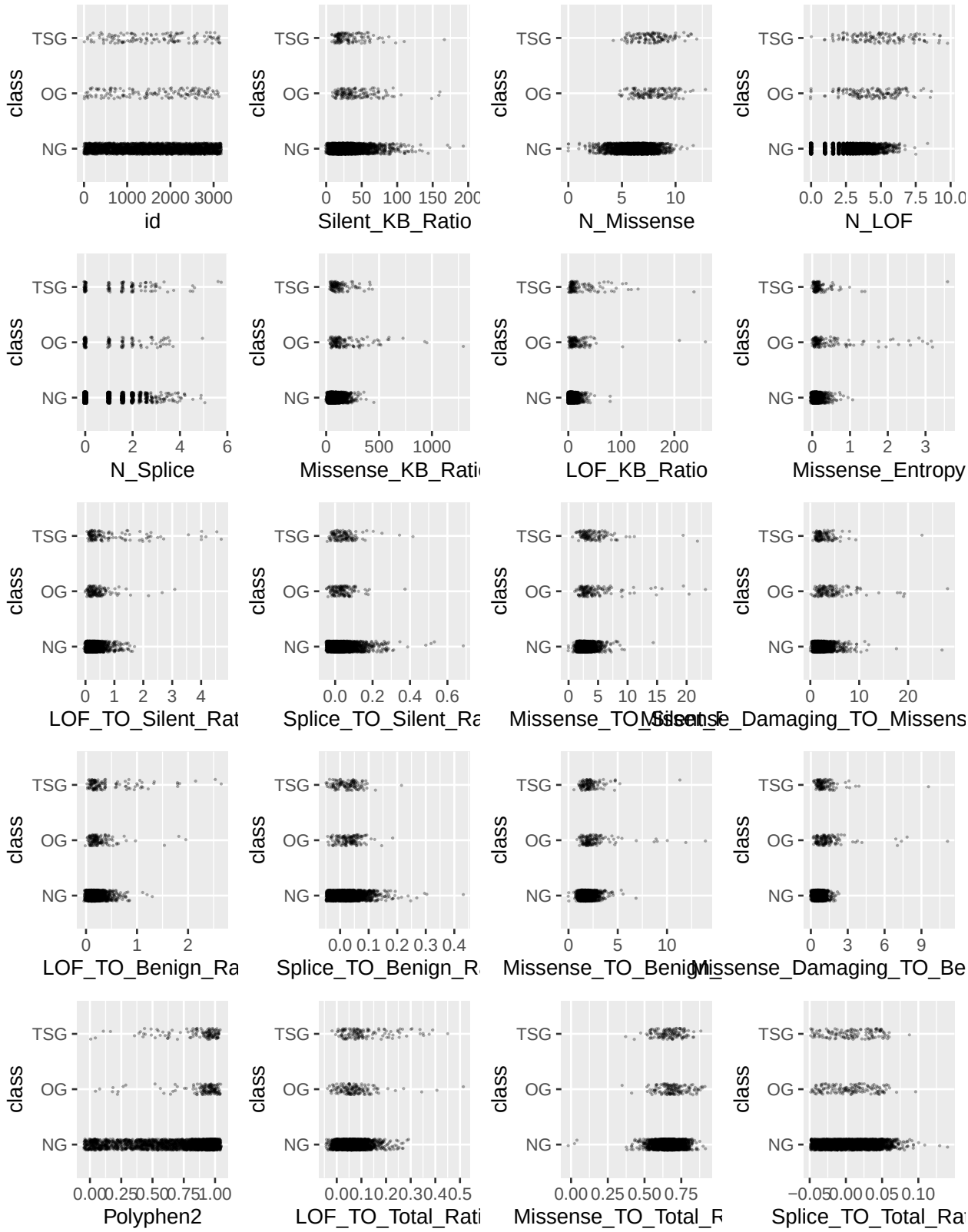
```
 [1] -20.5110 -10.9780 -10.2960  -9.7346  -9.3720  -8.8090  -8.4062  -8.3918
 [9]  -8.1076  -8.1076
```

```
training <- training[-which(training$Synonymous_Zscore < -15), ]
numeric_training <- training[,-99]

# Lots of zeroes for some observations, so removed those observations with quite a few zeroes across pr
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
}
training <- training[n_zeroes <= ncol(training) / 2, ]
# Remove 1 of 2 variables that have r value of 1 or -1 (i.e. perfectly correlated)
perf_corr <- which(abs(cor(training[,-99])) == 1, arr.ind = TRUE)
which(perf_corr[,1] != perf_corr[,2])
```

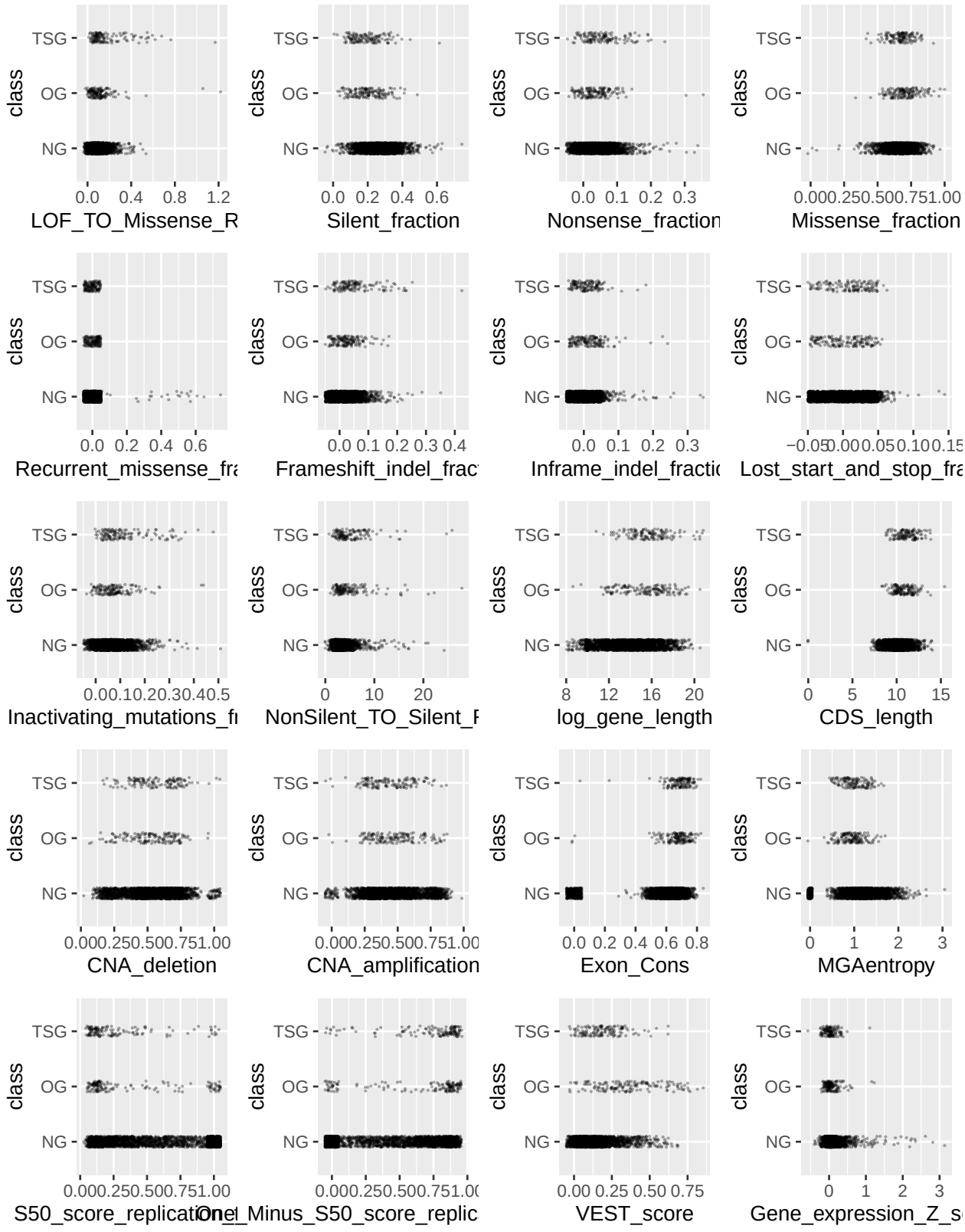
```
Minus_Cell_proliferation_rate_CRISPR_KD       Cell_proliferation_rate_CRISPR_KD
                                    43                                      44
    Gene_body_hypomethylation_in_cancer    Gene_body_hypermethylation_in_cancer
                                    53                                      54
```

```
training <- training[, -43]
```

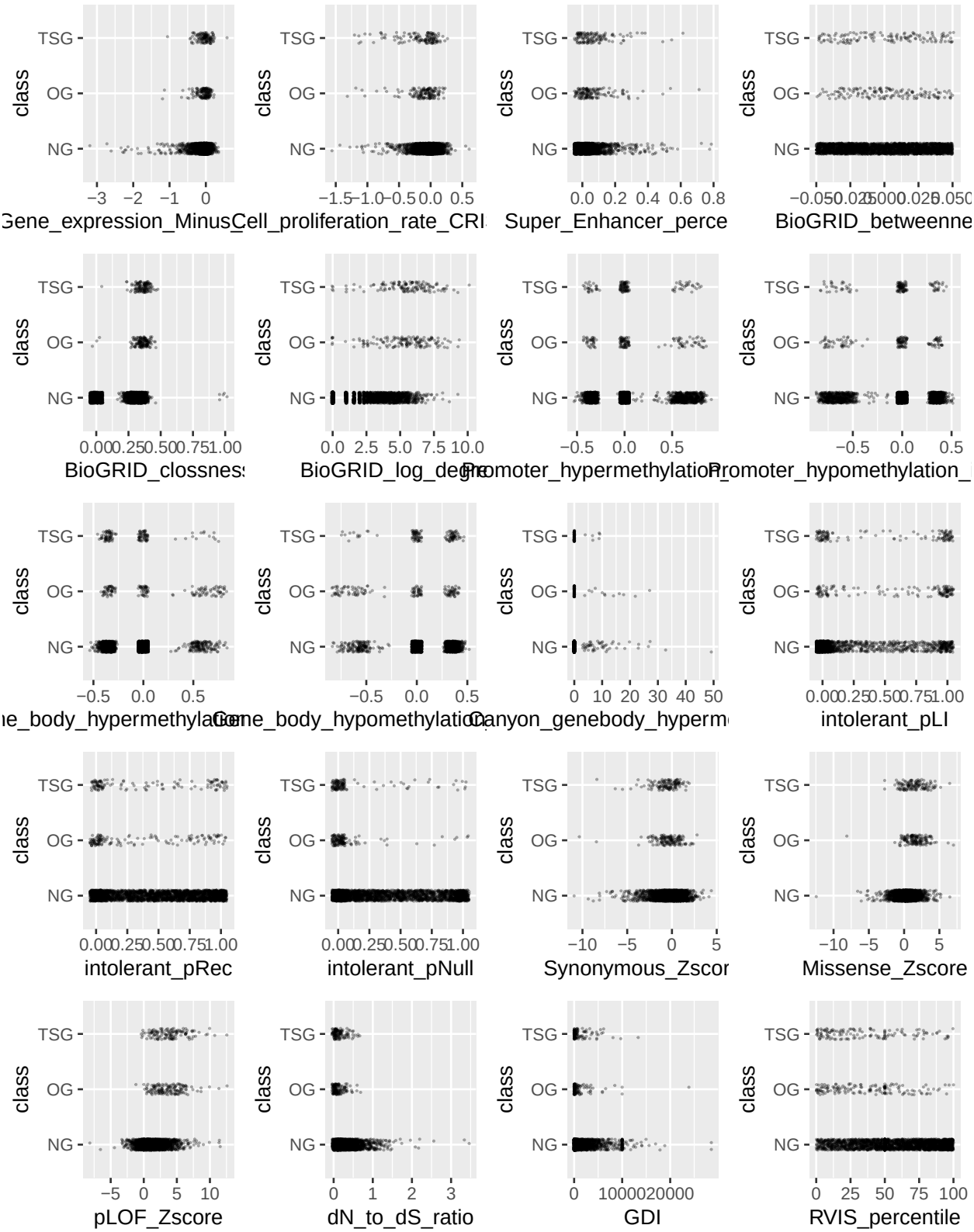
```
library(ggplot2)
scatter <- function(var) {
  ggplot(training, aes_string(var, "class")) +
    geom_jitter(width = 0.05, height = 0.1, size = 0.1,
                colour = rgb(0, 0, 0, alpha = 1 / 3))
}
scat_plot <- lapply(names(training)[-98], scatter)
library(gridExtra)
grid.arrange(grobs = scat_plot[1:20],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



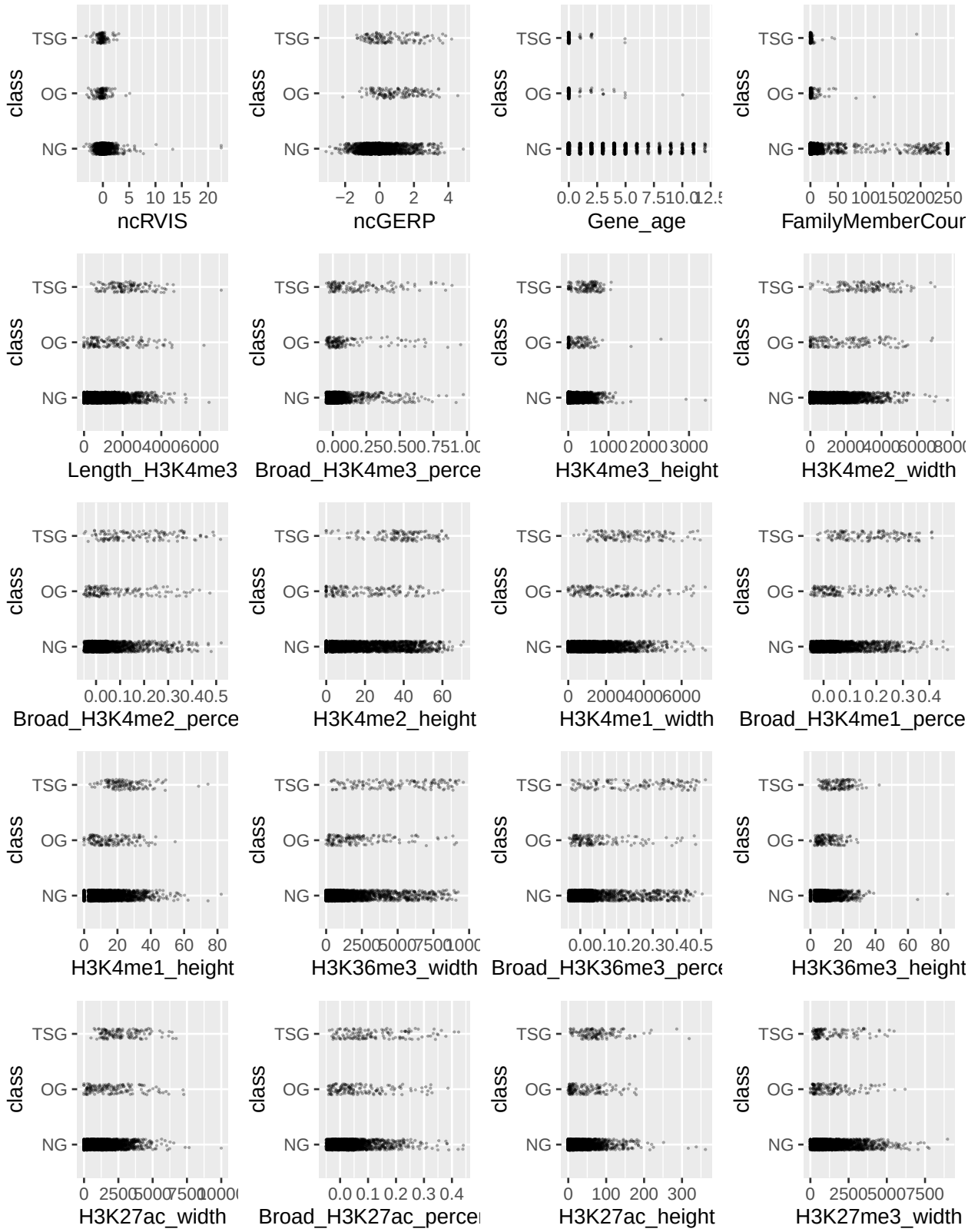

```
grid.arrange(grobs = scat_plot[21:40], ncol = 4)
```



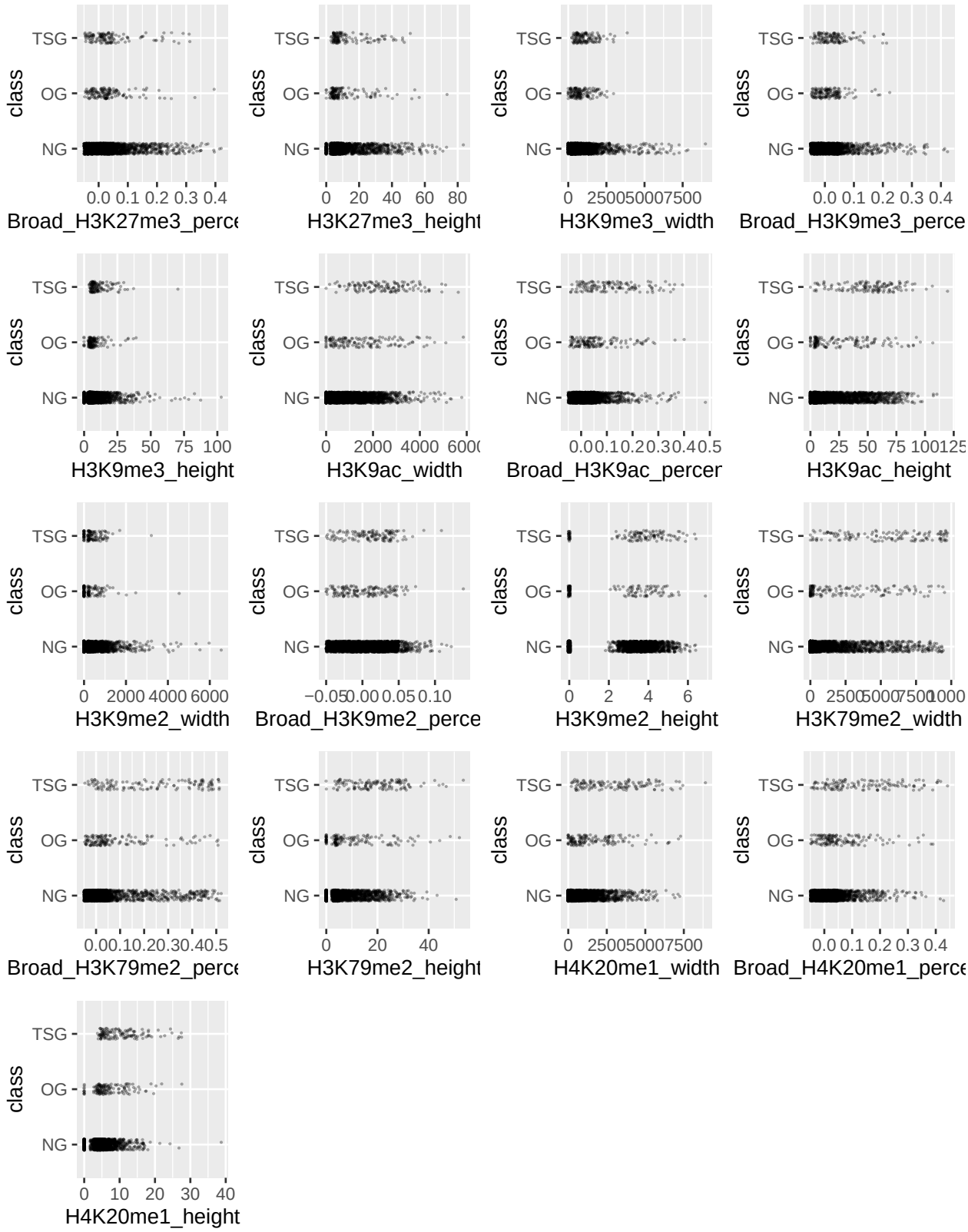

```
grid.arrange(grobs = scat_plot[41:60], ncol = 4)
```




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




```
grid.arrange(grobs = scat_plot[81:97], ncol = 4)
```




```
library(dplyr)
```

```
Attaching package: 'dplyr'

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

    combine

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
#function to calculate wca
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)))
}

# Set new threshold to account for unbalanced data
classify <- function(probs, k) {
  if (any(probs[2:3] > k)) {
    subset <- probs[2:3]
    output <- which(subset == max(subset))
    if (length(output) > 1) {
        output <- sample(1:2, 1)
    }
  } else {
    output <- 0
  }
  output
}
```

Multinom (Logistic Regression)

```
library(dplyr)
library(caret)
```

```
Loading required package: lattice
```

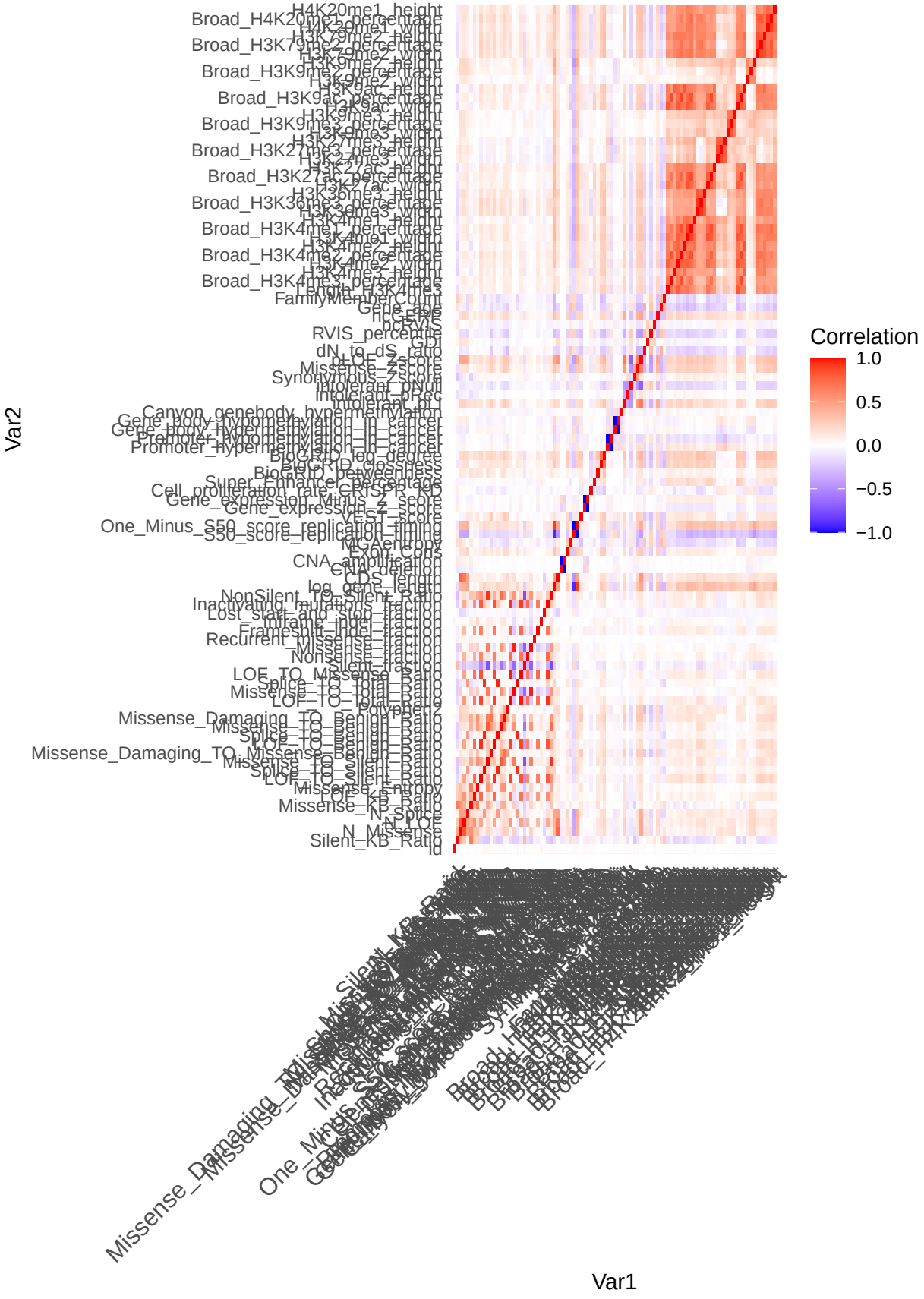
```
set.seed(12)
library(dplyr)
sig <- logical(97)
names(sig) <- names(training)[-98]
k <- 1
diffs <- logical(97)
for (var in names(training)[-98]) {
  model <- aov(training[[var]] ~ factor(training$class))
  sig[k] <- summary(model)[[1]][1, 5]
  diffs[k] <- all(TukeyHSD(model)$`factor(training$class)`[, 4] < 0.05 / 97)
  k <- k + 1
}
sort(sig[diffs])
```

```
                Broad_H4K20me1_percentage
                             3.975365e-146
                                VEST_score
                             1.518621e-121
                  Broad_H3K9ac_percentage
                             1.014012e-110
                           H3K79me2_height
                             1.806683e-110
                            H3K79me2_width
                             8.109050e-110
                          Missense_Entropy
                             1.652872e-109
                Broad_H3K79me2_percentage
                             6.487163e-107
       Missense_Damaging_TO_Benign_Ratio
                             2.174392e-106
                 Broad_H3K4me2_percentage
                             1.693160e-105
                           H4K20me1_width
                             9.802812e-105
                Broad_H3K36me3_percentage
                             1.138945e-103
                           H3K36me3_width
                             6.404467e-103
                          H4K20me1_height
                             1.762519e-101
                     LOF_TO_Silent_Ratio
                             6.478273e-101
                 Broad_H3K27ac_percentage
                              7.193535e-97
                 Broad_H3K4me1_percentage
                              1.431918e-95
                              LOF_KB_Ratio
                              5.656587e-93
```

```
                          Missense_KB_Ratio
                                6.106427e-89
                             H3K4me1_width
                                8.592826e-80
                           LOF_TO_Benign_Ratio
                                2.045965e-78
                      Broad_H3K4me3_percentage
                                4.120957e-78
                            H3K36me3_height
                                3.587724e-75
                             H3K4me2_width
                                1.258079e-73
                             H3K9ac_width
                                1.323555e-71
Missense_Damaging_TO_Missense_Benign_Ratio
                                3.272695e-68
                             H3K9ac_height
                                2.252510e-65
                      Missense_TO_Benign_Ratio
                                7.295266e-64
                             H3K27ac_width
                                3.349174e-63
                            Length_H3K4me3
                                1.431847e-60
                             H3K27ac_height
                                3.227318e-58
                          LOF_TO_Total_Ratio
                                1.021068e-47
                         LOF_TO_Missense_Ratio
                                2.322931e-42
                             H3K4me1_height
                                8.251224e-42
                    Frameshift_indel_fraction
                                5.738075e-34
```

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

# sig_terms <- names(sort(sig[diffs]))
# combo_terms <- table(c(terms, sig_terms)) == 2
# final_terms <- names(combo_terms[combo_terms])
# term_mat <- cor(training %>% select(all_of(final_terms)))
# ind_terms <- final_terms[-c(1, 3, 4, 5, 10, 11, 13, 14, 15, 17)]

vars <- training
cor_mtx <- round(cor(vars[, names(vars) != "class"]), 2)
library(reshape2)
melted_cor_mtx <- melt(cor_mtx)
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 = "Lab", name="Correlation") +
  theme_minimal() +
```

```
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1))
cor_heatmap
```


Var2
Var1
Correlation
1.0
0.5
0.0
−0.5
−1.0

```r
which(abs(cor_mtx) > 0.75, arr.ind = TRUE)
```

```
                                          row col
id                                          1   1
Silent_KB_Ratio                             2   2
N_Missense                                  3   3
N_LOF                                       4   3
CDS_length                                 32   3
N_Missense                                  3   4
N_LOF                                       4   4
N_Splice                                    5   5
Missense_KB_Ratio                           6   6
LOF_KB_Ratio                                7   7
Missense_Entropy                            8   8
LOF_TO_Silent_Ratio                         9   9
LOF_TO_Benign_Ratio                        13   9
LOF_TO_Total_Ratio                         18   9
LOF_TO_Missense_Ratio                      21   9
Inactivating_mutations_fraction            29   9
Splice_TO_Silent_Ratio                     10  10
Splice_TO_Benign_Ratio                     14  10
Splice_TO_Total_Ratio                      20  10
Missense_TO_Silent_Ratio                   11  11
Missense_TO_Benign_Ratio                   15  11
NonSilent_TO_Silent_Ratio                  30  11
Missense_Damaging_TO_Missense_Benign_Ratio 12  12
LOF_TO_Silent_Ratio                         9  13
LOF_TO_Benign_Ratio                        13  13
LOF_TO_Total_Ratio                         18  13
LOF_TO_Missense_Ratio                      21  13
Inactivating_mutations_fraction            29  13
Splice_TO_Silent_Ratio                     10  14
Splice_TO_Benign_Ratio                     14  14
Splice_TO_Total_Ratio                      20  14
Missense_TO_Silent_Ratio                   11  15
Missense_TO_Benign_Ratio                   15  15
Missense_Damaging_TO_Benign_Ratio          16  16
Polyphen2                                  17  17
LOF_TO_Silent_Ratio                         9  18
LOF_TO_Benign_Ratio                        13  18
LOF_TO_Total_Ratio                         18  18
LOF_TO_Missense_Ratio                      21  18
Inactivating_mutations_fraction            29  18
Missense_TO_Total_Ratio                    19  19
Missense_fraction                          24  19
Splice_TO_Silent_Ratio                     10  20
Splice_TO_Benign_Ratio                     14  20
Splice_TO_Total_Ratio                      20  20
LOF_TO_Silent_Ratio                         9  21
LOF_TO_Benign_Ratio                        13  21
LOF_TO_Total_Ratio                         18  21
LOF_TO_Missense_Ratio                      21  21
Inactivating_mutations_fraction            29  21
Silent_fraction                            22  22
```

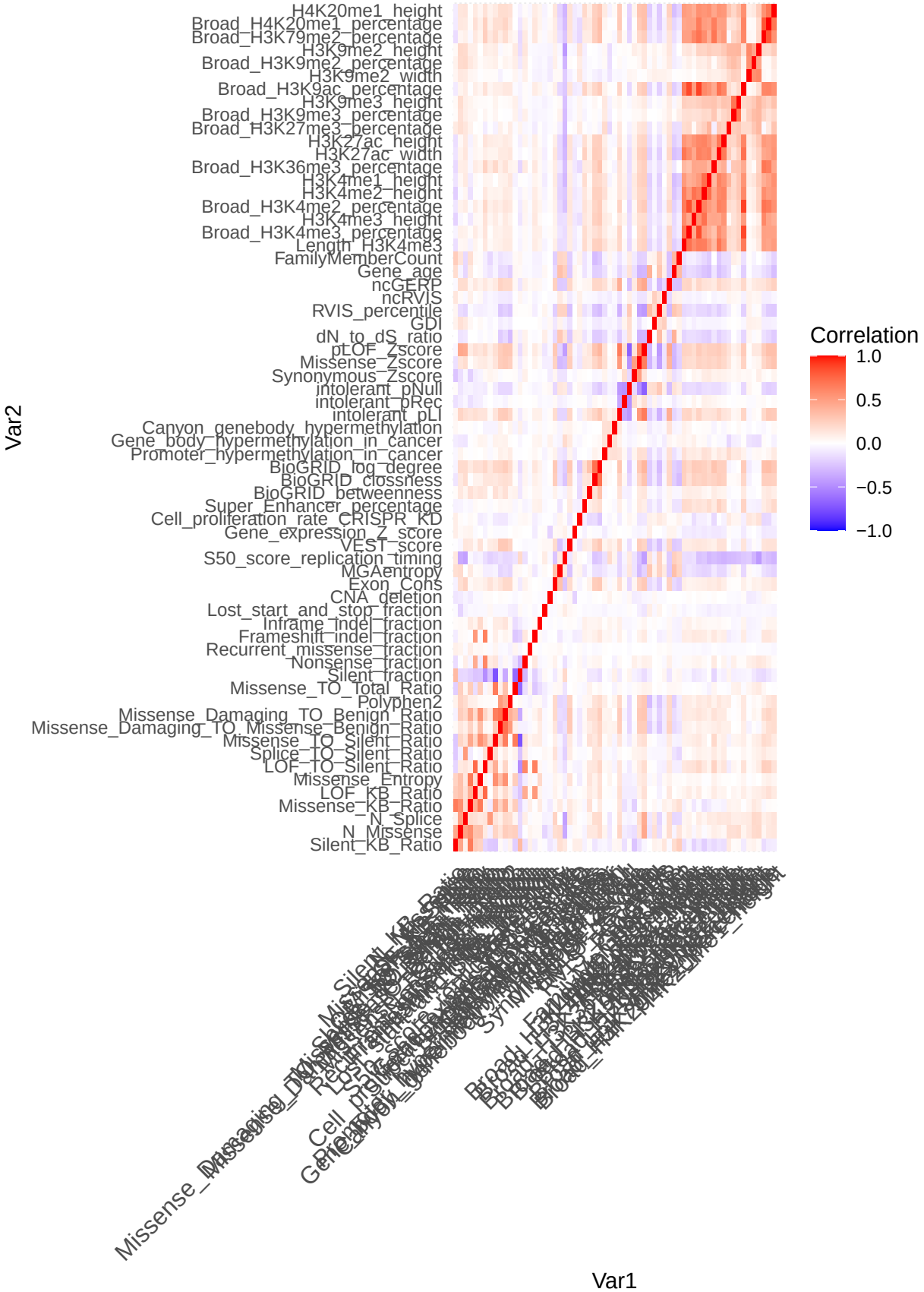
NonSilent_TO_Silent_Ratio	30	22
Nonsense_fraction	23	23
Missense_TO_Total_Ratio	19	24
Missense_fraction	24	24
Recurrent_missense_fraction	25	25
Frameshift_indel_fraction	26	26
Inframe_indel_fraction	27	27
Lost_start_and_stop_fraction	28	28
LOF_TO_Silent_Ratio	9	29
LOF_TO_Benign_Ratio	13	29
LOF_TO_Total_Ratio	18	29
LOF_TO_Missense_Ratio	21	29
Inactivating_mutations_fraction	29	29
Missense_TO_Silent_Ratio	11	30
Silent_fraction	22	30
NonSilent_TO_Silent_Ratio	30	30
log_gene_length	31	31
S50_score_replication_timing	37	31
One_Minus_S50_score_replication_timing	38	31
N_Missense	3	32
CDS_length	32	32
CNA_deletion	33	33
CNA_amplification	34	33
CNA_deletion	33	34
CNA_amplification	34	34
Exon_Cons	35	35
MGAentropy	36	36
log_gene_length	31	37
S50_score_replication_timing	37	37
One_Minus_S50_score_replication_timing	38	37
log_gene_length	31	38
S50_score_replication_timing	37	38
One_Minus_S50_score_replication_timing	38	38
VEST_score	39	39
Gene_expression_Z_score	40	40
Gene_expression_Minus_Z_score	41	40
Gene_expression_Z_score	40	41
Gene_expression_Minus_Z_score	41	41
Cell_proliferation_rate_CRISPR_KD	42	42
Super_Enhancer_percentage	43	43
BioGRID_betweenness	44	44
BioGRID_clossness	45	45
BioGRID_log_degree	46	46
Promoter_hypermethylation_in_cancer	47	47
Promoter_hypomethylation_in_cancer	48	47
Promoter_hypermethylation_in_cancer	47	48
Promoter_hypomethylation_in_cancer	48	48
Gene_body_hypermethylation_in_cancer	49	49
Gene_body_hypomethylation_in_cancer	50	49
Gene_body_hypermethylation_in_cancer	49	50
Gene_body_hypomethylation_in_cancer	50	50
Canyon_genebody_hypermethylation	51	51
intolerant_pLI	52	52
intolerant_pRec	53	53

intolerant_pNull	54	54
Synonymous_Zscore	55	55
Missense_Zscore	56	56
pLOF_Zscore	57	57
dN_to_dS_ratio	58	58
GDI	59	59
RVIS_percentile	60	60
ncRVIS	61	61
ncGERP	62	62
Gene_age	63	63
FamilyMemberCount	64	64
Length_H3K4me3	65	65
H3K4me2_width	68	65
H3K9ac_width	86	65
Broad_H3K4me3_percentage	66	66
Broad_H3K4me2_percentage	69	66
Broad_H3K9ac_percentage	87	66
H3K4me3_height	67	67
H3K9ac_height	88	67
Length_H3K4me3	65	68
H3K4me2_width	68	68
Broad_H3K4me2_percentage	69	68
H3K4me2_height	70	68
H3K4me1_width	71	68
H3K9ac_width	86	68
Broad_H3K4me3_percentage	66	69
H3K4me2_width	68	69
Broad_H3K4me2_percentage	69	69
Broad_H3K4me1_percentage	72	69
Broad_H3K27ac_percentage	78	69
Broad_H3K9ac_percentage	87	69
H3K4me2_width	68	70
H3K4me2_height	70	70
H3K9ac_height	88	70
H3K4me2_width	68	71
H3K4me1_width	71	71
Broad_H3K4me1_percentage	72	71
H3K27ac_width	77	71
Broad_H3K27ac_percentage	78	71
H3K9ac_width	86	71
Broad_H3K9ac_percentage	87	71
Broad_H3K4me2_percentage	69	72
H3K4me1_width	71	72
Broad_H3K4me1_percentage	72	72
Broad_H3K27ac_percentage	78	72
Broad_H3K9ac_percentage	87	72
H3K4me1_height	73	73
H3K36me3_width	74	74
Broad_H3K36me3_percentage	75	74
H3K36me3_height	76	74
H3K36me3_width	74	75
Broad_H3K36me3_percentage	75	75
H3K36me3_height	76	75
H3K36me3_width	74	76

Broad_H3K36me3_percentage	75	76
H3K36me3_height	76	76
H3K4me1_width	71	77
H3K27ac_width	77	77
Broad_H3K27ac_percentage	78	77
H3K9ac_width	86	77
Broad_H3K4me2_percentage	69	78
H3K4me1_width	71	78
Broad_H3K4me1_percentage	72	78
H3K27ac_width	77	78
Broad_H3K27ac_percentage	78	78
Broad_H3K9ac_percentage	87	78
H3K27ac_height	79	79
H3K27me3_width	80	80
Broad_H3K27me3_percentage	81	80
H3K27me3_height	82	80
H3K27me3_width	80	81
Broad_H3K27me3_percentage	81	81
H3K27me3_height	82	81
H3K27me3_width	80	82
Broad_H3K27me3_percentage	81	82
H3K27me3_height	82	82
H3K9me3_width	83	83
Broad_H3K9me3_percentage	84	83
H3K9me3_width	83	84
Broad_H3K9me3_percentage	84	84
H3K9me3_height	85	85
Length_H3K4me3	65	86
H3K4me2_width	68	86
H3K4me1_width	71	86
H3K27ac_width	77	86
H3K9ac_width	86	86
Broad_H3K9ac_percentage	87	86
Broad_H3K4me3_percentage	66	87
Broad_H3K4me2_percentage	69	87
H3K4me1_width	71	87
Broad_H3K4me1_percentage	72	87
Broad_H3K27ac_percentage	78	87
H3K9ac_width	86	87
Broad_H3K9ac_percentage	87	87
H3K4me3_height	67	88
H3K4me2_height	70	88
H3K9ac_height	88	88
H3K9me2_width	89	89
Broad_H3K9me2_percentage	90	90
H3K9me2_height	91	91
H3K79me2_width	92	92
Broad_H3K79me2_percentage	93	92
H3K79me2_height	94	92
Broad_H4K20me1_percentage	96	92
H3K79me2_width	92	93
Broad_H3K79me2_percentage	93	93
H3K79me2_height	94	93
Broad_H4K20me1_percentage	96	93

```
H3K79me2_width                          92  94
Broad_H3K79me2_percentage               93  94
H3K79me2_height                         94  94
H4K20me1_width                          95  95
Broad_H4K20me1_percentage               96  95
H3K79me2_width                          92  96
Broad_H3K79me2_percentage               93  96
H4K20me1_width                          95  96
Broad_H4K20me1_percentage               96  96
H4K20me1_height                         97  97
```

```r
# Eliminate all but one of a set of variables highly correlated with one another
vars <- vars[, -c(1, 4, 32, 13, 18, 21, 29, 14, 20, 15, 30,
                  15, 24, 31, 38, 34, 41, 48, 50, 68, 86, 88,
                  71, 72, 78, 74, 76, 80, 82, 83, 92, 94, 95)]
cor_mtx <- round(cor(vars[, names(vars) != "class"]), 2)
library(reshape2)
melted_cor_mtx <- melt(cor_mtx)
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 = "Lab", name="Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1))
cor_heatmap
```


H4K20me1_height
Broad_H4K20me1_percentage
Broad_H3K79me2_percentage
H3K9me2_height
Broad_H3K9me2_percentage
H3K9me2_width
Broad_H3K9ac_percentage
H3K9me3_height
Broad_H3K9me3_percentage
Broad_H3K27me3_percentage
H3K27ac_height
H3K27ac_width
Broad_H3K36me3_percentage
H3K4me1_height
H3K4me2_height
Broad_H3K4me2_percentage
H3K4me3_height
Broad_H3K4me3_percentage
Length_H3K4me3
FamilyMemberCount
Gene_age
ncGERP
ncRVIS
RVIS_percentile
GDI
dN_to_dS_ratio
pLOF_Zscore
Missense_Zscore
Synonymous_Zscore
intolerant_pNull
intolerant_pRec
intolerant_pLI
Canyon_genebody_hypermethylation
Gene_body_hypermethylation_in_cancer
Promoter_hypermethylation_in_cancer
BioGRID_log_degree
BioGRID_closness
BioGRID_betweenness
Super_Enhancer_percentage
Cell_proliferation_rate_CRISPR_KD
Gene_expression_Z_score
VEST_score
S50_score_replication_timing
MGAentropy
Exon_Cons
CNA_deletion
Lost_start_and_stop_fraction
Inframe_Indel_fraction
Frameshift_Indel_fraction
Recurrent_missense_fraction
Nonsense_fraction
Silent_fraction
Missense_TO_Total_Ratio
Polyphen2
Missense_Damaging_TO_Benign_Ratio
Missense_Damaging_TO_Missense_Benign_Ratio
Missense_TO_Silent_Ratio
Splice_TO_Silent_Ratio
LOF_TO_Silent_Ratio
Missense_Entropy
LOF_KB_Ratio
Missense_KB_Ratio
N_Splice
N_Missense
Silent_KB_Ratio
Var2
Var1
Correlation
1.0
0.5
0.0
−0.5
−1.0

```r
which(abs(cor_mtx) > 0.6, arr.ind = TRUE)
```

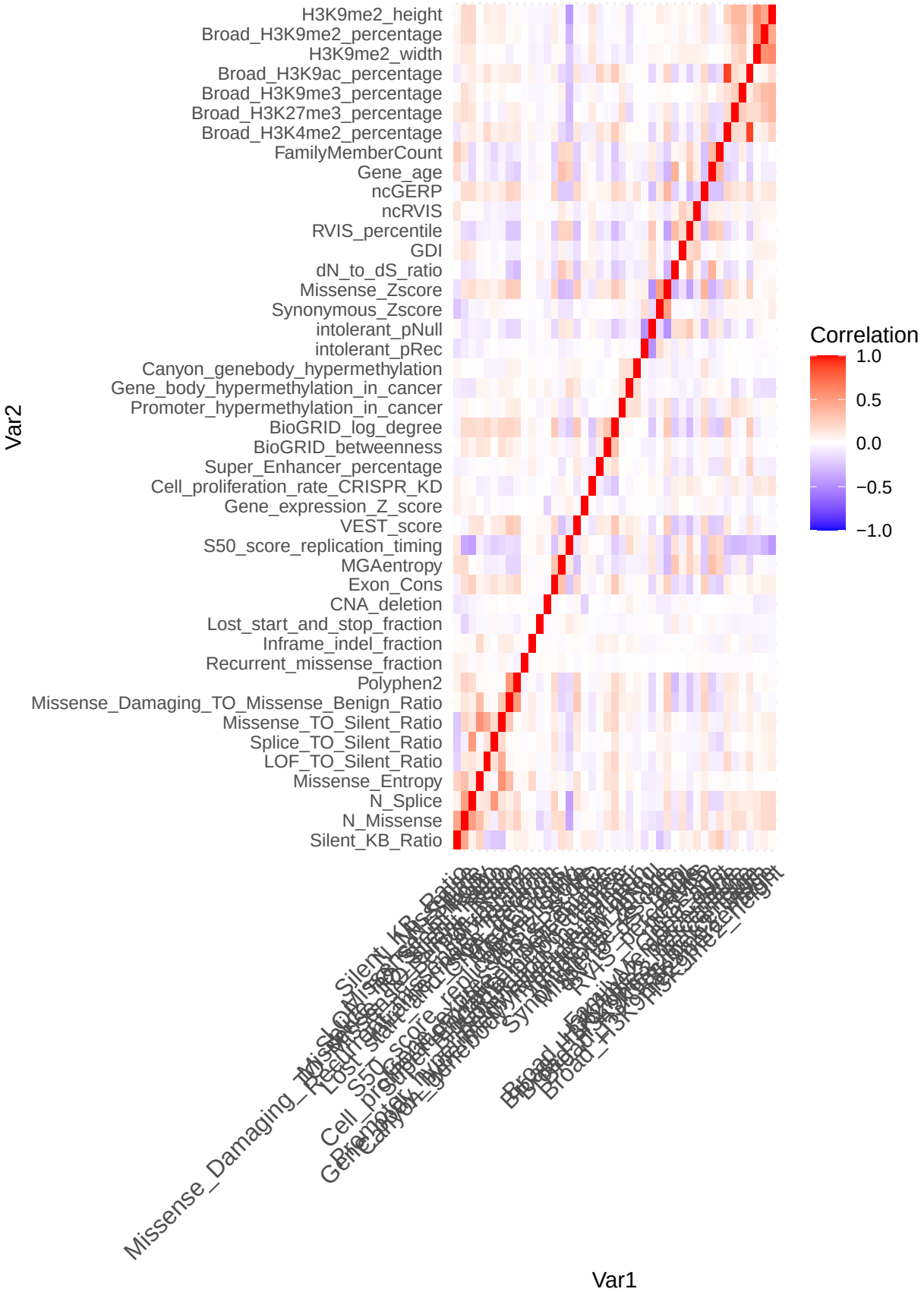
```
                                          row col
Silent_KB_Ratio                             1   1
Missense_KB_Ratio                           4   1
N_Missense                                  2   2
N_Splice                                    3   3
Silent_KB_Ratio                             1   4
Missense_KB_Ratio                           4   4
Missense_Entropy                            6   4
LOF_KB_Ratio                                5   5
LOF_TO_Silent_Ratio                         7   5
Missense_KB_Ratio                           4   6
Missense_Entropy                            6   6
LOF_KB_Ratio                                5   7
LOF_TO_Silent_Ratio                         7   7
Nonsense_fraction                          15   7
Frameshift_indel_fraction                  17   7
Splice_TO_Silent_Ratio                      8   8
Missense_TO_Silent_Ratio                    9   9
Missense_Damaging_TO_Benign_Ratio          11   9
Missense_TO_Total_Ratio                    13   9
Silent_fraction                            14   9
Missense_Damaging_TO_Missense_Benign_Ratio 10  10
Missense_Damaging_TO_Benign_Ratio          11  10
Missense_TO_Silent_Ratio                    9  11
Missense_Damaging_TO_Missense_Benign_Ratio 10  11
Missense_Damaging_TO_Benign_Ratio          11  11
Polyphen2                                  12  12
Missense_TO_Silent_Ratio                    9  13
Missense_TO_Total_Ratio                    13  13
Silent_fraction                            14  13
Missense_TO_Silent_Ratio                    9  14
Missense_TO_Total_Ratio                    13  14
Silent_fraction                            14  14
LOF_TO_Silent_Ratio                         7  15
Nonsense_fraction                          15  15
Recurrent_missense_fraction                16  16
LOF_TO_Silent_Ratio                         7  17
Frameshift_indel_fraction                  17  17
Inframe_indel_fraction                     18  18
Lost_start_and_stop_fraction               19  19
CNA_deletion                               20  20
Exon_Cons                                  21  21
MGAentropy                                 22  22
S50_score_replication_timing               23  23
VEST_score                                 24  24
Gene_expression_Z_score                    25  25
Cell_proliferation_rate_CRISPR_KD          26  26
Super_Enhancer_percentage                  27  27
BioGRID_betweenness                        28  28
BioGRID_clossness                          29  29
BioGRID_log_degree                         30  29
BioGRID_clossness                          29  30
```

BioGRID_log_degree	30	30
Promoter_hypermethylation_in_cancer	31	31
Gene_body_hypermethylation_in_cancer	32	32
Canyon_genebody_hypermethylation	33	33
intolerant_pLI	34	34
pLOF_Zscore	39	34
intolerant_pRec	35	35
intolerant_pNull	36	36
pLOF_Zscore	39	36
Synonymous_Zscore	37	37
Missense_Zscore	38	38
pLOF_Zscore	39	38
intolerant_pLI	34	39
intolerant_pNull	36	39
Missense_Zscore	38	39
pLOF_Zscore	39	39
dN_to_dS_ratio	40	40
GDI	41	41
RVIS_percentile	42	42
ncRVIS	43	43
ncGERP	44	44
Gene_age	45	45
FamilyMemberCount	46	46
Length_H3K4me3	47	47
Broad_H3K4me3_percentage	48	47
Broad_H3K4me2_percentage	50	47
H3K4me2_height	51	47
H3K27ac_width	54	47
Broad_H3K9ac_percentage	59	47
Length_H3K4me3	47	48
Broad_H3K4me3_percentage	48	48
Broad_H3K4me2_percentage	50	48
Broad_H3K9ac_percentage	59	48
H3K4me3_height	49	49
H3K4me2_height	51	49
Broad_H3K79me2_percentage	63	49
Length_H3K4me3	47	50
Broad_H3K4me3_percentage	48	50
Broad_H3K4me2_percentage	50	50
H3K4me2_height	51	50
H3K27ac_width	54	50
H3K27ac_height	55	50
Broad_H3K9ac_percentage	59	50
Broad_H3K79me2_percentage	63	50
Broad_H4K20me1_percentage	64	50
Length_H3K4me3	47	51
H3K4me3_height	49	51
Broad_H3K4me2_percentage	50	51
H3K4me2_height	51	51
H3K4me1_height	52	51
H3K27ac_height	55	51
H3K4me2_height	51	52
H3K4me1_height	52	52
H3K27ac_width	54	52

```
Broad_H3K36me3_percentage                53  53
Broad_H3K79me2_percentage                63  53
Broad_H4K20me1_percentage                64  53
Length_H3K4me3                           47  54
Broad_H3K4me2_percentage                 50  54
H3K4me1_height                           52  54
H3K27ac_width                            54  54
H3K27ac_height                           55  54
Broad_H3K9ac_percentage                  59  54
Broad_H3K4me2_percentage                 50  55
H3K4me2_height                           51  55
H3K27ac_width                            54  55
H3K27ac_height                           55  55
Broad_H3K9ac_percentage                  59  55
Broad_H3K27me3_percentage                56  56
Broad_H3K9me3_percentage                 57  57
H3K9me3_height                           58  57
Broad_H3K9me3_percentage                 57  58
H3K9me3_height                           58  58
Length_H3K4me3                           47  59
Broad_H3K4me3_percentage                 48  59
Broad_H3K4me2_percentage                 50  59
H3K27ac_width                            54  59
H3K27ac_height                           55  59
Broad_H3K9ac_percentage                  59  59
Broad_H3K79me2_percentage                63  59
Broad_H4K20me1_percentage                64  59
H3K9me2_width                            60  60
Broad_H3K9me2_percentage                 61  61
H3K9me2_height                           62  62
H3K4me3_height                           49  63
Broad_H3K4me2_percentage                 50  63
Broad_H3K36me3_percentage                53  63
Broad_H3K9ac_percentage                  59  63
Broad_H3K79me2_percentage                63  63
Broad_H4K20me1_percentage                64  63
Broad_H3K4me2_percentage                 50  64
Broad_H3K36me3_percentage                53  64
Broad_H3K9ac_percentage                  59  64
Broad_H3K79me2_percentage                63  64
Broad_H4K20me1_percentage                64  64
H4K20me1_height                          65  64
Broad_H4K20me1_percentage                64  65
H4K20me1_height                          65  65
```

```r
# Eliminate all but one of a set of variables highly correlated with one another
vars <- vars[, -c(4, 5, 15, 17, 11, 13, 14, 29, 39, 47,
                  48, 51, 49, 54, 55, 52, 58, 63, 65,
                  34, 64, 53)]

cor_mtx <- round(cor(vars[, names(vars) != "class"]), 2)
library(reshape2)
melted_cor_mtx <- melt(cor_mtx)
cor_heatmap <- ggplot(data = melted_cor_mtx, aes(x = Var1, y = Var2, fill = value)) + geom_tile()
```

```r
cor_heatmap <- cor_heatmap +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab", name="Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1))
cor_heatmap
```


H3K9me2_height
Broad_H3K9me2_percentage
H3K9me2_width
Broad_H3K9ac_percentage
Broad_H3K9me3_percentage
Broad_H3K27me3_percentage
Broad_H3K4me2_percentage
FamilyMemberCount
Gene_age
ncGERP
ncRVIS
RVIS_percentile
GDI
dN_to_dS_ratio
Missense_Zscore
Synonymous_Zscore
intolerant_pNull
intolerant_pRec
Canyon_genebody_hypermethylation
Gene_body_hypermethylation_in_cancer
Promoter_hypermethylation_in_cancer
BioGRID_log_degree
BioGRID_betweenness
Super_Enhancer_percentage
Cell_proliferation_rate_CRISPR_KD
Gene_expression_Z_score
VEST_score
S50_score_replication_timing
MGAentropy
Exon_Cons
CNA_deletion
Lost_start_and_stop_fraction
Inframe_indel_fraction
Recurrent_missense_fraction
Polyphen2
Missense_Damaging_TO_Missense_Benign_Ratio
Missense_TO_Silent_Ratio
Splice_TO_Silent_Ratio
LOF_TO_Silent_Ratio
Missense_Entropy
N_Splice
N_Missense
Silent_KB_Ratio
Var2
Var1
Correlation
1.0
0.5
0.0
−0.5
−1.0

```r
set.seed(12)
mn <- nnet::multinom(class ~ ., data = vars, model = TRUE)
```

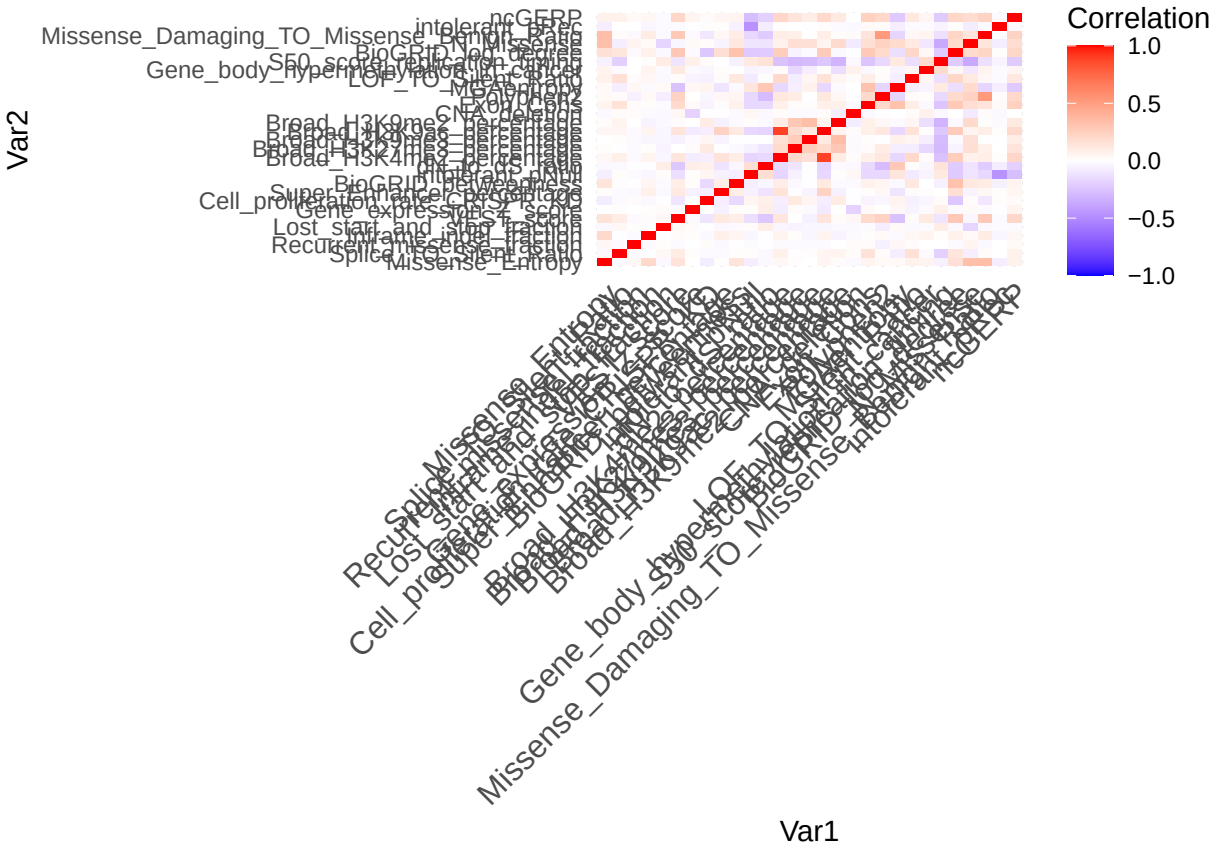
```
# weights:  135 (88 variable)
initial  value 3323.302173
iter  10 value 2200.558341
iter  20 value 1805.402799
iter  30 value 1277.085182
iter  40 value 574.167012
iter  50 value 391.767327
iter  60 value 386.643100
iter  70 value 385.103013
iter  80 value 384.497863
iter  90 value 384.220123
iter 100 value 384.101004
final  value 384.101004
stopped after 100 iterations
```

```r
tidymn <- broom::tidy(mn) %>% arrange(p.value)
tidymn <- tidymn %>% filter(p.value < 0.05 / nrow(tidymn))
terms <- unique(tidymn$term)[-1]
terms
```

```
 [1] "Missense_Entropy"
 [2] "Splice_TO_Silent_Ratio"
 [3] "Recurrent_missense_fraction"
 [4] "Inframe_indel_fraction"
 [5] "Lost_start_and_stop_fraction"
 [6] "VEST_score"
 [7] "Gene_expression_Z_score"
 [8] "Cell_proliferation_rate_CRISPR_KD"
 [9] "Super_Enhancer_percentage"
[10] "BioGRID_betweenness"
[11] "intolerant_pNull"
[12] "dN_to_dS_ratio"
[13] "Broad_H3K4me2_percentage"
[14] "Broad_H3K27me3_percentage"
[15] "Broad_H3K9me3_percentage"
[16] "Broad_H3K9ac_percentage"
[17] "Broad_H3K9me2_percentage"
[18] "CNA_deletion"
[19] "Exon_Cons"
[20] "Polyphen2"
[21] "MGAentropy"
[22] "LOF_TO_Silent_Ratio"
[23] "Gene_body_hypermethylation_in_cancer"
[24] "S50_score_replication_timing"
[25] "BioGRID_log_degree"
[26] "N_Missense"
[27] "Missense_Damaging_TO_Missense_Benign_Ratio"
[28] "intolerant_pRec"
[29] "ncGERP"
```

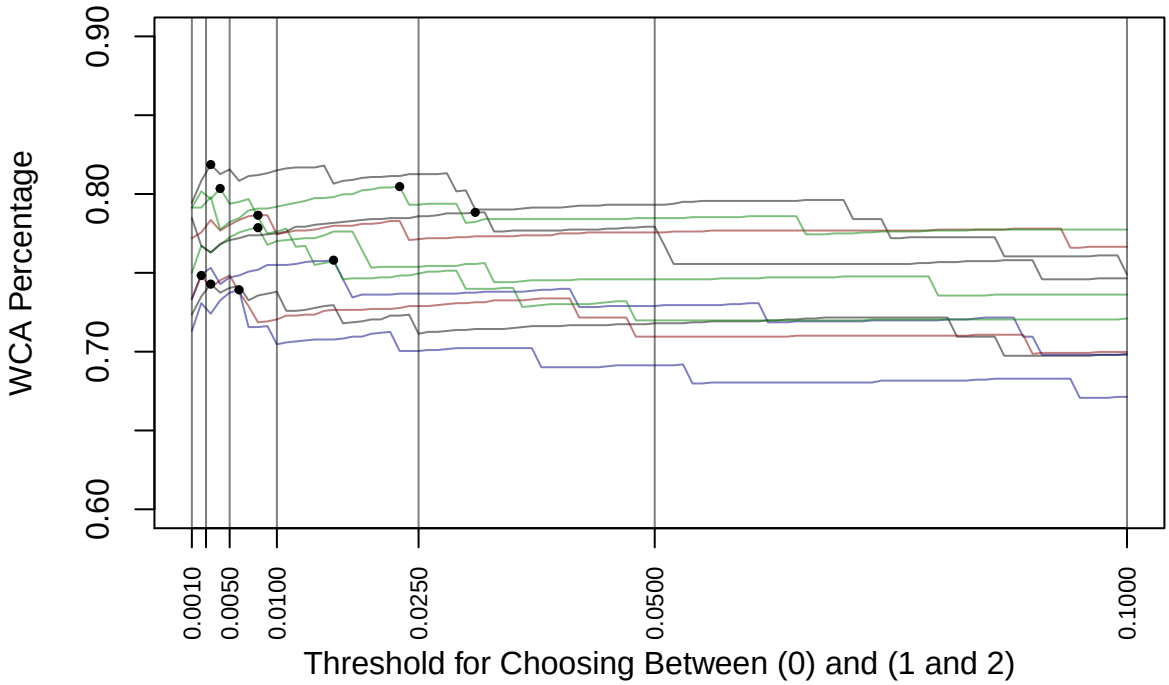
```r
cor_mtx <- round(cor(vars[, terms]), 2)
library(reshape2)
```

```
melted_cor_mtx <- melt(cor_mtx)
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 = "Lab", name="Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1))
cor_heatmap
```




```
vars <- training %>% select(all_of(terms), class)
set.seed(217)
thresh_list <- list()
length(thresh_list) <- 10
j <- 1
# for() loop to validate post cross-validation with WCA and varying thresholds
for (rand in ceiling(runif(10, min = 1, max = 10000000))) {
  set.seed(rand)
  vars_split <- createDataPartition(vars$class, p = 0.8, list = FALSE)
  vars_train <- vars[vars_split, ]
  vars_test <- vars[-vars_split, ]

  train_cont <- trainControl(method = "cv", number = 5, classProbs = TRUE, savePredictions = TRUE)
  lda_ft <- train(class ~ ., data = vars_train, method = "lda", preProc = c("center", "scale"),
                  trControl = train_cont)
  preds <- predict(lda_ft, newdata = vars_test, type = "prob")
  scores <- numeric(100)
  for (i in seq(from = 0.001, to = 0.1, by = 0.001)) {
```

```
    lda_mod <- table("pred"=apply(preds, 1, classify, k = i), "obs" = vars_test$class)
    lda_mod
    scores[i*1000] <- score(lda_mod)
  }
  thresh_list[[j]] <- scores
  j <- j + 1
}

# Plot WCA as a function of threshold for various seeds to see best threshold
thresh_vals <- c(0.001, 0.0025, 0.005, 0.01, 0.025, 0.05, 0.1)
plot(thresh_list[[1]] ~ seq(from = 0.001, to = 0.1, by = 0.001),
     type = "l", ylim = c(0.6, 0.9), col = rgb(0, 0, 0, alpha = 0.5),
     xaxt = "n", ylab = "WCA Percentage",
     xlab = "Threshold for Choosing Between (0) and (1 and 2)")
for (i in 2:10) {
  red <- 0
  green <- 0
  blue <- 0
  if (i %% 4 == 1) {red <- 0.5}
  if (i %% 4 == 2) {green <- 0.5}
  if (i %% 4 == 3) {blue <- 0.5}
  lines(seq(from = 0.001, to = 0.1, by = 0.001), thresh_list[[i]],
        col = rgb(red, green, blue, alpha = 0.5))

}
abline(v = thresh_vals, col = rgb(0, 0, 0, alpha = 0.5))
points(unlist(lapply(thresh_list, which.max)) * 0.001, unlist(lapply(thresh_list, max)),
       pch = 19, cex = 0.5)
axis(side = 1, at = thresh_vals, cex.axis = 0.75, las = 2)
```




```
best_thresh <- unlist(lapply(thresh_list, which.max))
median(best_thresh * 0.001)
```

```
[1] 0.007
```

```
thresh_val_scores <- matrix(nrow = 10, ncol = length(thresh_vals))
colnames(thresh_val_scores) <- thresh_vals
for (i in 1:10) {
  thresh_val_scores[i, ] <- thresh_list[[i]][thresh_vals * 1000]
}
apply(thresh_val_scores, 2, mean)
```

```
    0.001    0.0025     0.005      0.01     0.025      0.05       0.1
0.7587629 0.7674348 0.7688902 0.7620982 0.7543966 0.7446331 0.7264403
```

```
# Use threshold of 0.005 for determining when to classify as OG or TSG (highest peak)
set.seed(1248)
vars_split <- createDataPartition(vars$class, p = 0.8, list = FALSE)
vars_train <- vars[vars_split, ]
vars_test <- vars[-vars_split, ]
train_cont <- trainControl(method = "cv", number = 5,
                           classProbs = TRUE, savePredictions = TRUE)
lda_ft <- train(class ~ ., data = vars_train, method = "lda",
                preProc = c("center", "scale"), trControl = train_cont)
preds <- predict(lda_ft, newdata = vars_test, type = "prob")
lda_mod <- table("pred" = apply(preds, 1, classify, k = 0.005),
                 "obs" = vars_test$class)
lda_mod
```

```
    obs
pred  NG  OG TSG
   0 479   2   1
   1  27  22   3
   2  43   5  22
```

```
score(lda_mod)
```

```
[1] 0.8241358
```

```
# Write prediction model on test data
tests <- read.csv("test.csv")
preds <- predict(lda_ft, newdata = tests, type = "prob")
preds <- apply(preds, 1, classify, k = 0.005)
names(preds) <- tests$id
csv_file <- data.frame(id = tests$id,
                       class = preds)
# write.csv(csv_file, "modelpredictions13.csv", row.names = FALSE)
best_mod <- read.csv("modelpredictions12.csv")
table("Best Model" = best_mod$class, csv_file$class)
```

```
Best Model    0    1    2
         0 1070   15   28
         1    1  105    3
         2    0    2  139
```

```
mean(best_mod$class == csv_file$class)
```

```
[1] 0.9640499
```