MAF File Analysis

Mohamed Faisal

Package Installations

```
# install maftools
if (!require(package = BiocManager))
    install.packages(pkgs = "BiocManager")
```

```
## Loading required package: BiocManager
```

```
## Bioconductor version '3.19' is out-of-date; the current release version
'3.20'
##   is available with R version '4.4'; see https://bioconductor.org/install
```

```
BiocManager::install(pkgs = "maftools")
```

```
## Bioconductor version 3.19 (BiocManager 1.30.25), R 4.4.2 (2024-10-31)
```

```
## Warning: package(s) not installed when version(s) same as or greater than
current; use
##   `force = TRUE` to re-install: 'maftools'
```

```
library(package = maftools)
```

```
# install ggplot2
if (!require(package = "ggplot2"))
    install.packages(pkgs = "ggplot2")
```

```
## Loading required package: ggplot2
```

```
library(package = ggplot2)
```

1 - Uploading Files

```
work.dir <- "~/Desktop/MAF/"
setwd(dir = work.dir)

# read in sample data
sample.info <- read.table(file = "sample-information.tsv",
                          header = T, sep = "\t")

maf.dir <- paste0(work.dir, "mafs")

maf.files.path <- list.files(path = maf.dir, full.names = T)

# join all maf files into one large maf files
df <- read.table(file = maf.files.path[1], header = T, sep = "\t")
df$Patient_ID <- "Patient-0"

for(i in 2:length(x = maf.files.path)){
  df.h <- read.table(file = maf.files.path[i], header = T, sep = "\t")
  df.h$Patient_ID <- unlist(x = strsplit(x = df.h$Tumor_Sample_Barcode[1],
                                         split = "-Tumor"))
  df <- rbind(df, df.h)
}
```

2 - Filtering Silent Mutations & Marking Response Values

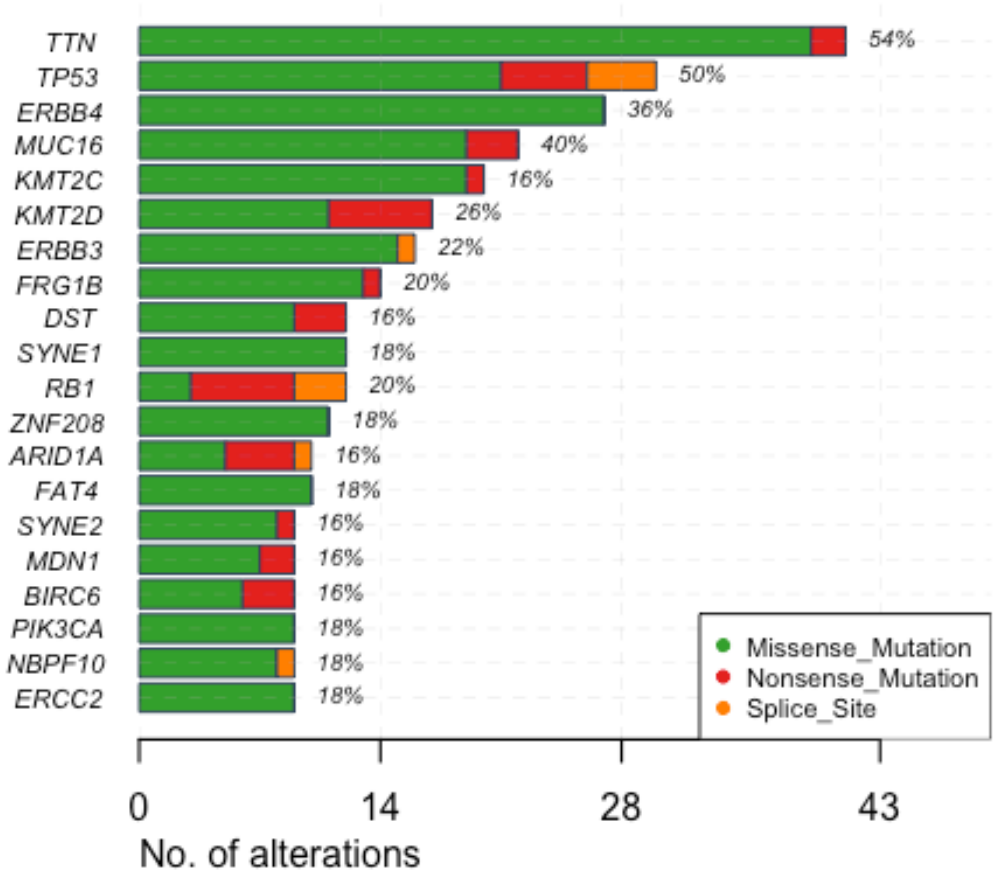
```r
# filter silent mutations
rm.rows <- which(x = df$Variant_Classification == "Silent")
df <- df[-rm.rows, ]

# correcting the order of the entries of the data frame
correct.order <- sample.info$Patient_ID
correct.index <- order(match(x = df$Patient_ID, table = correct.order))
df <- df[correct.index, ]

# labeling response values
responses <- sample.info$Response
Response <- c()
patients <- sample.info$Patient_ID

for(i in 1:length(x = patients)){
  response.i <- responses[i]
  patient.i <- patients[i]
  n <- sum(df$Patient_ID == patient.i)
  Response <- c(Response, rep(x = response.i, times = n))
}

 df$Response <- Response
```

3 - Finding The Most Common Mutated Genes & Mutations

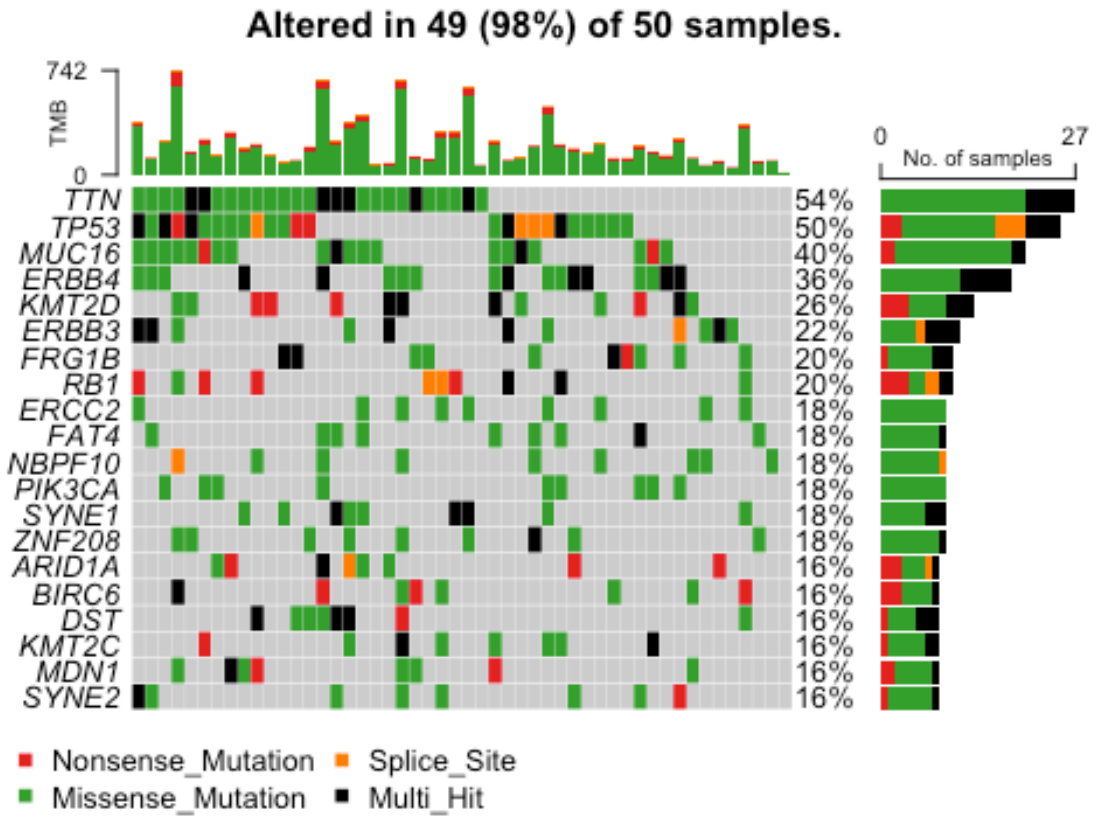
```
# convert df to MAF object with maftools
maf.df <- read.maf(maf = df, clinicalData = sample.info, verbose = F)

# get gene summary, i.e., genes with the most mutations
gene_summary.df <- getGeneSummary(x = maf.df)

# get the top 20 genes with the most mutations
top_20_mut_genes.df <- gene_summary.df[1:20, ]

# get the top 20 most mutated gene names
top_20_genes <- top_20_mut_genes.df$Hugo_Symbol
top_20_genes
```

```
##  [1] "TTN"    "TP53"   "MUC16"  "ERBB4"  "KMT2D"  "ERBB3"  "FRG1B"  "RB1"
##  [9] "SYNE1"  "ZNF208" "FAT4"   "ERCC2"  "NBPF10" "PIK3CA" "KMT2C"  "DST"
## [17] "ARID1A" "BIRC6"  "MDN1"   "SYNE2"
```

```
# Visualization of top 15 most common mutations
mafbarplot(maf = maf.df, n = 20)
```




```
oncoplot(maf = maf.df,  top = 20)
```




```
# Get top 20 most common mutations (Protein_Change)
top_20_muts <- head(x = df$Protein_Change, n = 20)
top_20_muts
```

```
##  [1] "p.Q612E"   "p.A676T"   "p.S1126C"  "p.V277I"   "p.S313L"
"p.E4442*"
##  [7] "p.D345N"   "p.R5H"     "p.R153T"   "p.E414Q"   "p.F1423L"  "p.D106N"
## [13] "p.D33700N" "p.E1616K"  "p.A3121T"  "p.S12C"    "p.E11K"    "p.R197Q"
## [19] "p.E1129K"  "p.E121K"
```

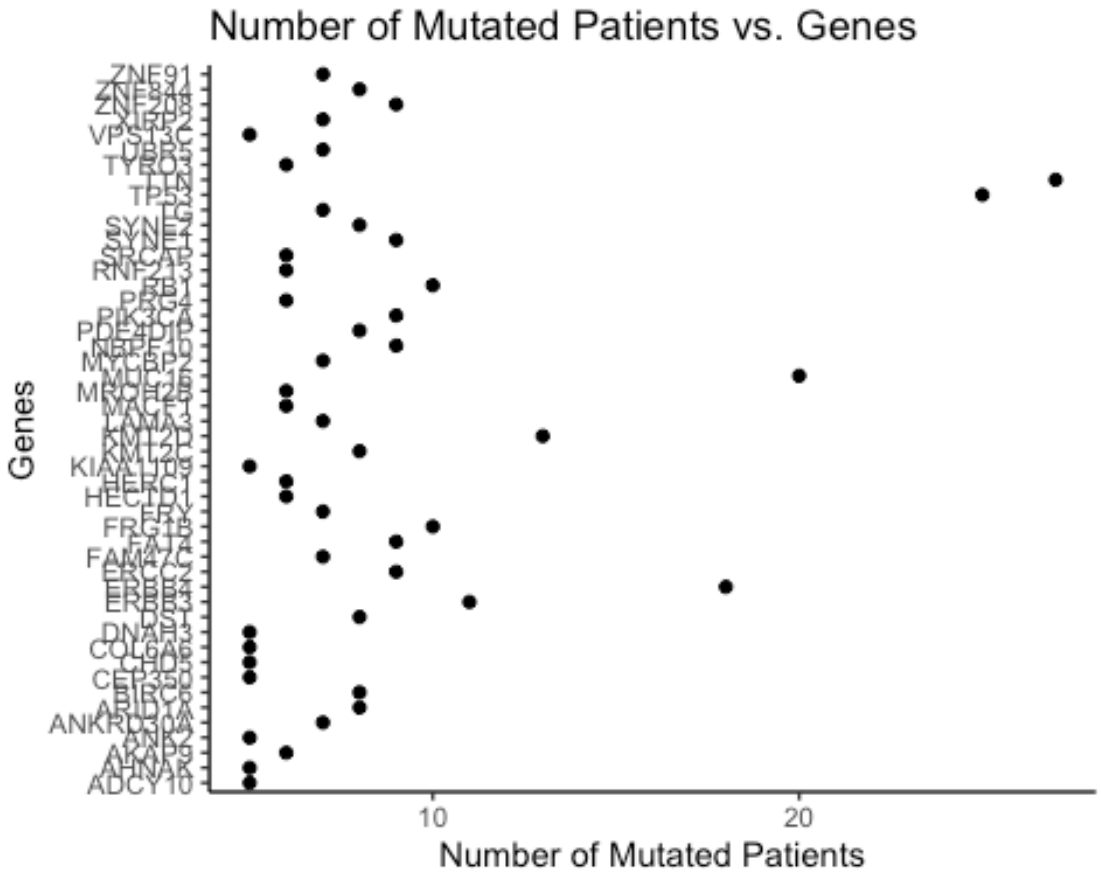
4 - Fisher Exact Test for Enriched Genes Amongst Cohorts

```
# split patients into responders vs non-responders cohorts
split.by <- which(x = df$Response == "Responder")

responders <- df[split.by, ]
nonresponders <- df[-split.by, ]

# convert cohorts to MAF objects via maftools
responders <- read.maf(maf = responders, verbose = F)
nonresponders <- read.maf(maf = nonresponders, verbose = F)

# compare the two cohorts using a Fisher Exact Test via maftools
# use a significance threshold of alpha = 0.05 for statistical significance
test.results <- mafCompare(m1 = responders, m2 = nonresponders,
                           m1Name = "Responders", m2Name = "Non-Responders",
                           minMut = 5)

head(test.results$results)
```

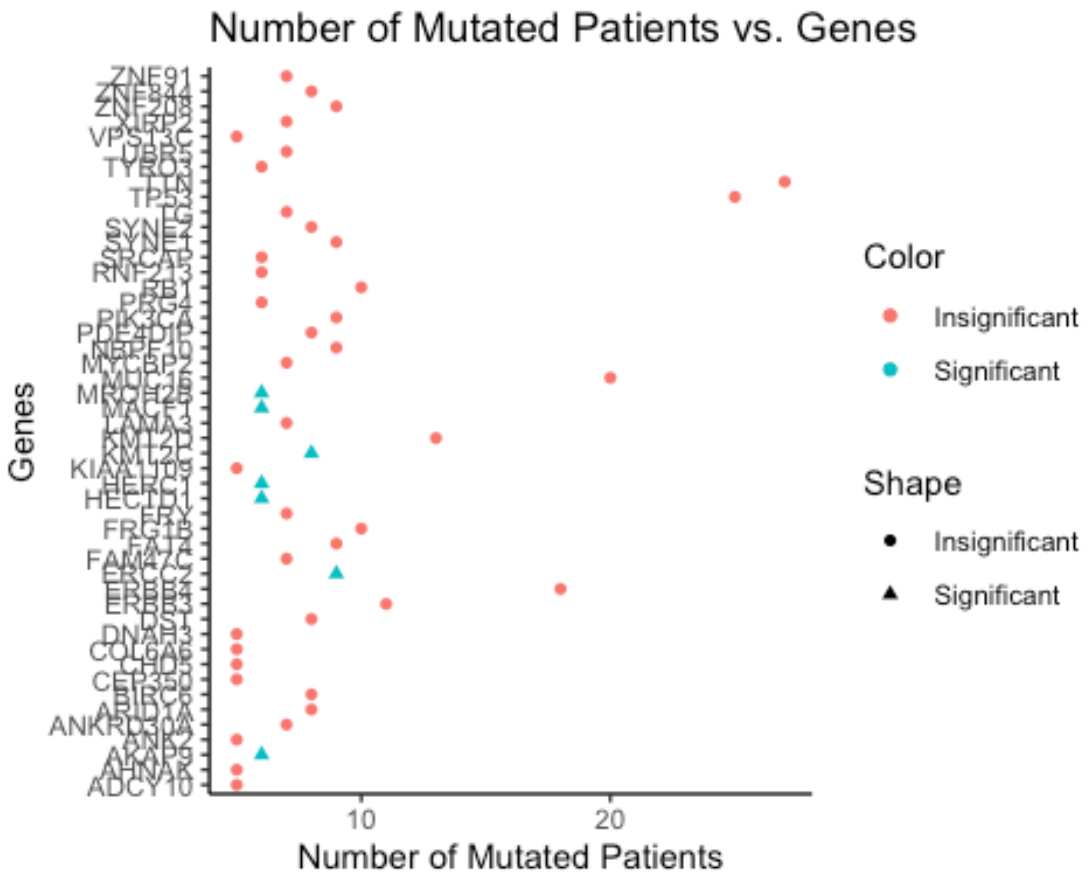
```
##    Hugo_Symbol Responders Non-Responders        pval    or ci.up   ci.low
##          <char>      <int>          <num>       <num> <num> <num>    <num>
## 1:       ERCC2          9              0 0.001630835   Inf   Inf 2.566473
## 2:       AKAP9          6              0 0.022289767   Inf   Inf 1.333588
## 3:      HECTD1          6              0 0.022289767   Inf   Inf 1.333588
## 4:       HERC1          6              0 0.022289767   Inf   Inf 1.333588
## 5:       MACF1          6              0 0.022289767   Inf   Inf 1.333588
## 6:      MROH2B          6              0 0.022289767   Inf   Inf 1.333588
##      adjPval
##        <num>
## 1: 0.0782801
## 2: 0.1504559
## 3: 0.1504559
## 4: 0.1504559
## 5: 0.1504559
## 6: 0.1504559
```

5 - Scatter Plot of Number of Mutated Patients vs. Top Enriched Genes

```
# get total number of mutated patients for the most enriched genes in both
# responder and non-responder cohorts
total_mutants <- test.results$results$Responders + test.results$results$`Non-
Responders`

test.results$results$total_mutants <- total_mutants

# plot number of mutated patients per gene vs. p-values of genes
s.df <- test.results$results[, c(1,9)]
colnames(s.df) <- c("Genes", "mutants")

s.plot <- ggplot(data = s.df, mapping = aes(x = mutants, y = Genes)) +
geom_point() + theme_bw() + theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank()) + xlab(label = "Number of Mutated
Patients") + ylab(label = "Genes") + ggtitle(label = "Number of Mutated
Patients vs. Genes")
# view scatter plot
s.plot
```



```r
## readability can be improved via utilizing p-values
s.df$pval <- test.results$results$pval

# significant points will be squares and red
# insignificant points will be circles and black
s.df$Shape <- as.factor(x = ifelse(test = s.df$pval <= 0.05,
                                   yes = "Significant",
                                   no = "Insignificant"))
s.df$Color <- as.factor(x = ifelse(test = s.df$pval <= 0.05,
                                   yes = "Significant",
                                   no = "Insignificant"))

s.plot <- ggplot(data = s.df, mapping = aes(x = mutants, y = Genes, shape =
Shape, color = Color)) + geom_point() + theme_bw() + theme(axis.line =
element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank()) + xlab(label = "Number of Mutated
Patients") + ylab(label = "Genes") + ggtitle(label = "Number of Mutated
Patients vs. Genes")

s.plot
```



6 - Plot of Nonsynonymous Sites/Mb for Mutant vs. Wild-Type Patients

```r
# the top enriched gene is the gene with the lowest p-value
gene.index <- which.min(x = test.results$results$pval)
most_enriched_gene <- test.results$results$Hugo_Symbol[gene.index]

# output the most enriched gene
most_enriched_gene
```

```
## [1] "ERCC2"
```

```r
# subset df for only mutants of the most enriched gene "ERCC2"
gene.df <- subsetMaf(maf = maf.df, genes = "ERCC2")
```

```
## -Processing clinical data
```

```r
# get mutant Patient-ID's for "ERCC2" -> 9 mutant samples
mutant.patients <- gene.df@data$Patient_ID
length(mutant.patients)
```

```
## [1] 9
```

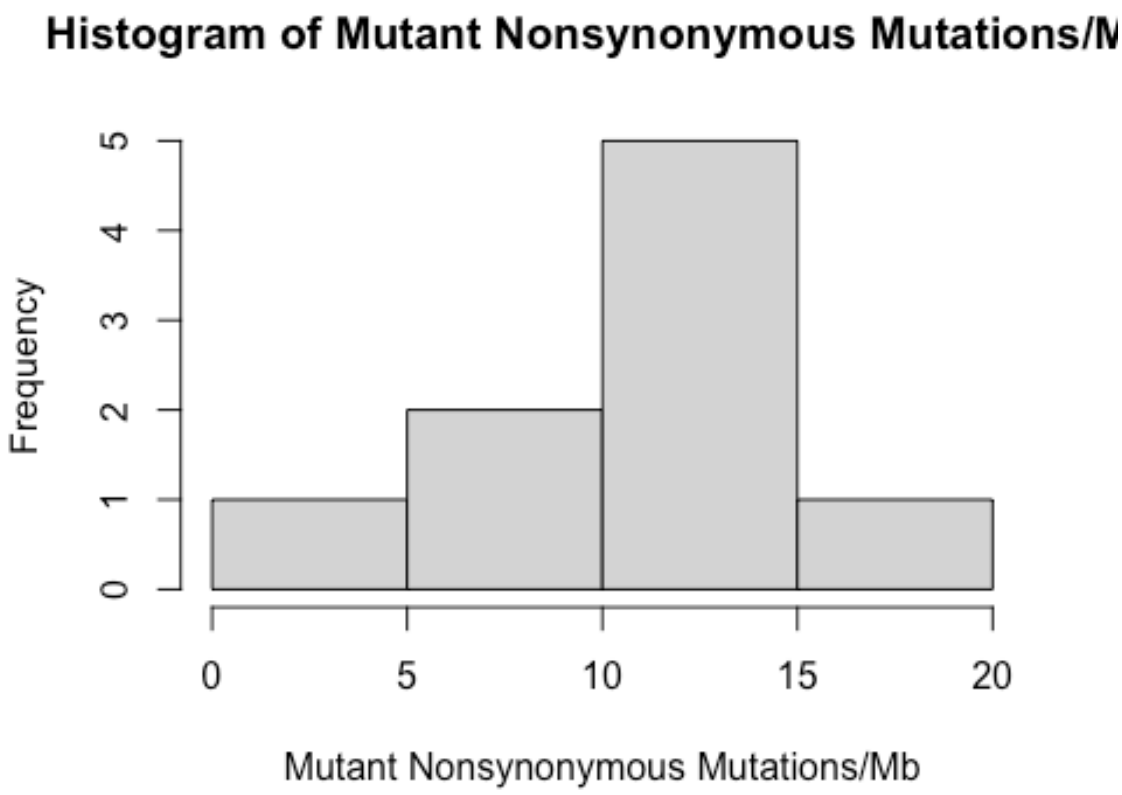
```r
# get index positions of mutant patients
mutant.index <- match(x = mutant.patients, table = sample.info$Patient_ID)

# get wild-type Patient-ID's for "ERCC2" -> 41 wild-type samples
wildtype.patients <- sample.info$Patient_ID[-mutant.index]
length(wildtype.patients)
```

```
## [1] 41
```

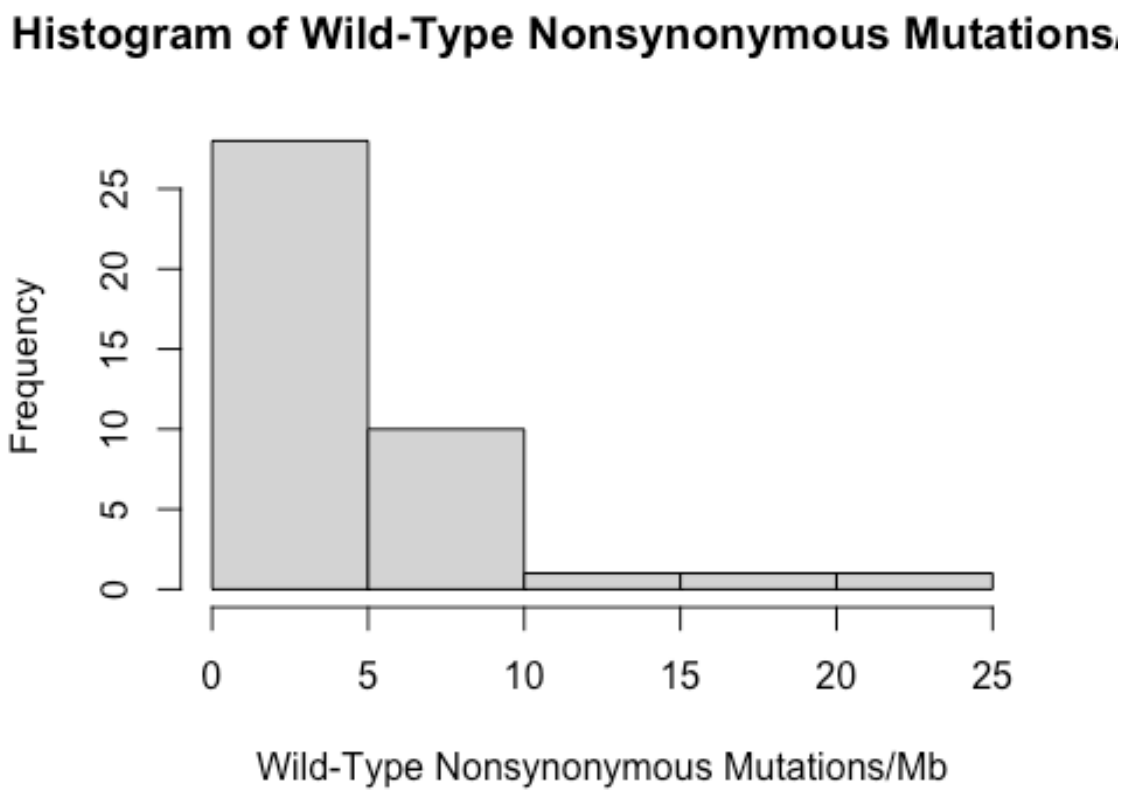
```r
## plotting nonsynonymous mutations/Mb for wild-type vs mutant patients

# make nonsynonymous df for plotting
nonsynon.df <- sample.info[c(1, 6)]
colnames(nonsynon.df)[2] <- "nonsynMb"
nonsynon.df$ID <- 0:49

# color code for mutants (1) vs wild-type (0)
nonsynon.df$ERCC2 <- "Wild-Type"
nonsynon.df$ERCC2[mutant.index] <- "Mutant"

# plot
non.p <- ggplot(data = nonsynon.df, mapping = aes(x = ID, y = nonsynMb, color
= ERCC2)) + geom_boxplot() + theme_bw() +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank()) + xlab(label = "Patient-ID") +
  ylab(label = "Nonsynonymous Mutations per Mb") +
```

```
  ggtitle(label = "Nonsynonymous Mutations per Mb in Mutant vs Wild-Type
Samples")

non.p
```


```
# testing for statistical significance between the two groups
mutant.nonysn <- nonsynon.df$nonsynMb[mutant.index]
wildtype.nonsyn <- nonsynon.df$nonsynMb[-mutant.index]

# checking normality of data to inform testing method
hist(x = mutant.nonysn, xlab = "Mutant Nonsynonymous Mutations/Mb",
     main = "Histogram of Mutant Nonsynonymous Mutations/Mb")
```




```
hist(x = wildtype.nonsyn, xlab = "Wild-Type Nonsynonymous Mutations/Mb",
     main = "Histogram of Wild-Type Nonsynonymous Mutations/Mb")
```

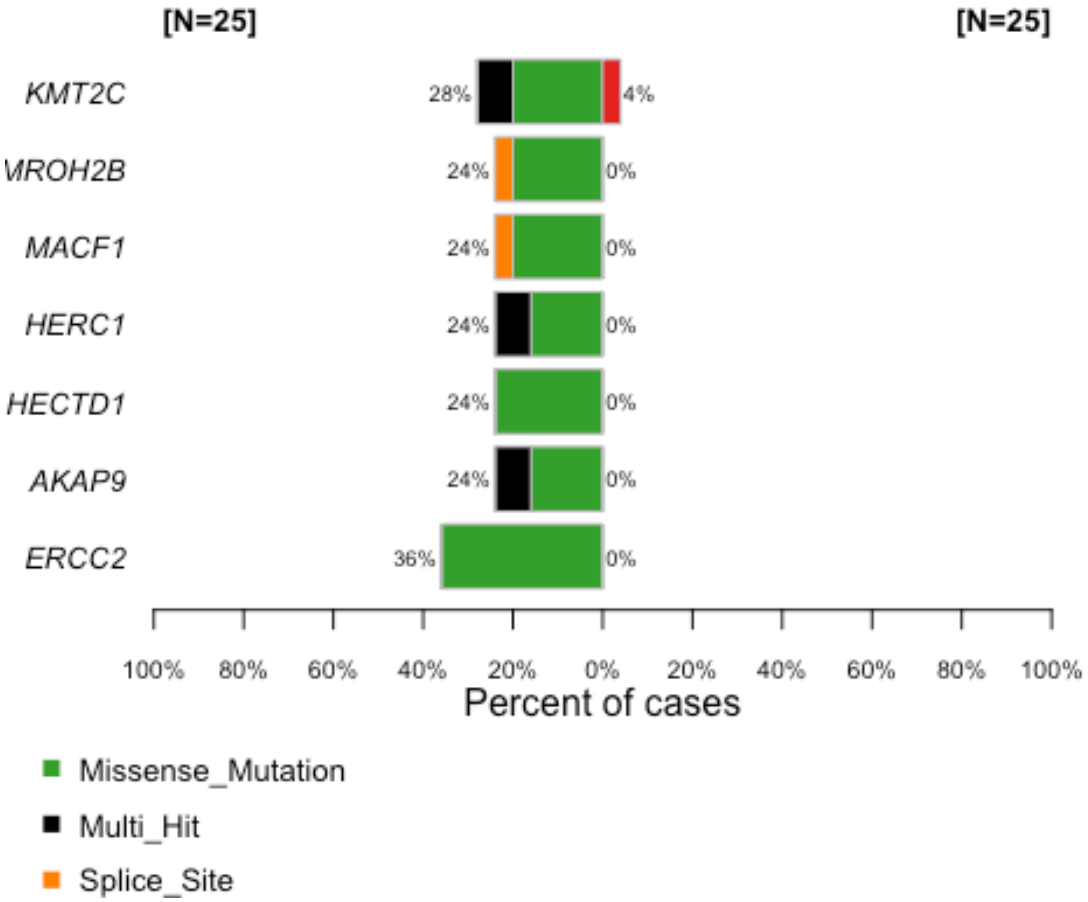



```
# normality assumption not met -> use a nonparametric test

## Wilcoxon Rank Sum Test

# Null Hypothesis - No difference in number of nonsynonymous mutations
between mutant and wild-type ERCC2 patients
# significance level = 0.05
w.test <- wilcox.test(x = mutant.nonysn, y = wildtype.nonsyn)

## Warning in wilcox.test.default(x = mutant.nonysn, y = wildtype.nonsyn):
cannot
## compute exact p-value with ties

w.test

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  mutant.nonysn and wildtype.nonsyn
## W = 309.5, p-value = 0.001666
## alternative hypothesis: true location shift is not equal to 0

w.test$p.value
```

```
## [1] 0.001666124

# The p-value of 0.001666124 <= 0.05, which allows us to reject the null
hypothesis of no difference in number of nonsynonymous mutations between
mutant and wild-type ERCC2 patients.

## There seems to be a statistically significant difference in the number of
nonsynonymous mutations per Mb between mutant and wild-type ERCC2 patients
```

7 - Concluding Remarks

```
# get the statistically significant enriched genes between cohorts and plot them
significant_genes <- s.df$Genes[1:7]
coBarplot(m1 = responders, m2 = nonresponders, genes = significant_genes)
```



As seen by the barplot comparison above of the statistically enriched genes for the responder vs. non-responder cohorts, all of the genes seem to be enriched within the responder cohort (left) as opposed to the non-responder cohort (right). The seven genes all encode and are crucial for many important cellular processes such as DNA damage repair and cell signaling. When mutated some of the genes (KMT2C, ERCC2, HECTD1) can help in the formation of cancer since they play a crucial role in repairing damaged DNA, and hence when mutated fail to do so and tend to allow for the proliferation of the cancer. Other genes when mutated (MACF1, MROH2B, HECTD1, HERC1) tend to cause cancer since they play a crucial role in various cell signaling pathways, hence, their mutant forms tends to encode products that aid in the proliferation and metastasis of cancer. Furthermore, the most statistically enriched gene ERCC2 is noted to be heavily involved in DNA repair as aforementioned. ERCC2 encodes for an essential subunit of the TFIIH complex (important for DNA repair and transcription regulation), and hence it is plausible the mechanism of action for the treatment is aiding in DNA repair by targeting the faulty defects of the TFIIH

complex. This analysis can be expanded upon by utilizing other forms of testing such as logistic regression. Since the response variable is binary (case-control) if more covariates (any suspected confounding factors) of the patients are obtained one could attempt to model the response variable and attempt to measure the effect sizes (weights/coefficients) of the genes that seem to be associated the most with response to treatment. This analysis can also be expanded upon by taking a larger sample size to increase statistical power and find more mutation associations with treatment response.