S&DS361 Final Project

Contents

Background

Reverse transcription is the process by which an enzyme (a kind of protein) called a reverse transcriptase (RT) reads and copies a strand of RNA and turns it into cDNA. RNA molecules are made up of a sequence of nucleotides, chemical bases Adenine, Uracil, Cytosine, and Guanine. Many of these nucleotide sites have chemical modifications on them that assist the cell in regulating many important cellular processes like gene expression. It is thus very useful to identify where modifications exist on the length of the RNA. Currently, this process involves painstakingly analyzing each site in a complex biochemical workflow. This methodology is extremely coarse-grained, nonspecific, and often necessitates multiple iterations of validation before modified sites on an RNA are confirmed. We characterise and study a system that may allow us to quickly and accurately detect chemical modifications on the length of an RNA sequence based on information about mutations that the RT incorporates into the cDNA molecule that it is synthesizing.

Eubacterium rectale maturase reverse transcriptase (Marathon RT), an enzyme discovered and studied by the Pyle Lab in the Yale MCDB Department, has been shown to be extremely processive and accurate in copying long RNA molecules in the presence of Mg^{2+} solution [Wang et. al.]. We call the action of the RT in the presence of Magnesium "high fidelity," because it incorporates very few mutations when it copies RNA to cDNA. However, in the presence of Mn^{2+} solution, MRT is susceptible to incorporating mutations during its reverse transcription mechanism. It has been hypothesized that nucleotide sites with endogenous covalent modifications on the RNA are especially susceptible to mutation incorporation by MRT in the presence of Mn^{2+}, i.e. the reverse transcriptase preferentially incorporates mutations at chemically modified nucleotide sites. In the presence of Manganese, therefore, the RT is "low fidelity": mutations are incorporated at a high rate.

The following code shows how we cleaned the original dataset (xlsx file) and created working dataset `new.filtered`. To the original nucleotide sequence data, a vector of modification types was appended to the dataframe (for example, at site 27, the nucleic acid shows an "Am"- type modification, which is a methyladenosine modification). Additionally, another column notes whether or not the site is modified at all (a factor). Lastly, one point was removed from the dataset because its mutation rate in both the control and treatment trials was 0.998 (very close to 1); we believe this was an experimental error.

```
dataset <- read_excel("1Mg2Mn_18S_human_18S_profile.xlsx")


#modifications and positions extracted from 3D Modifications U of Albany database
```

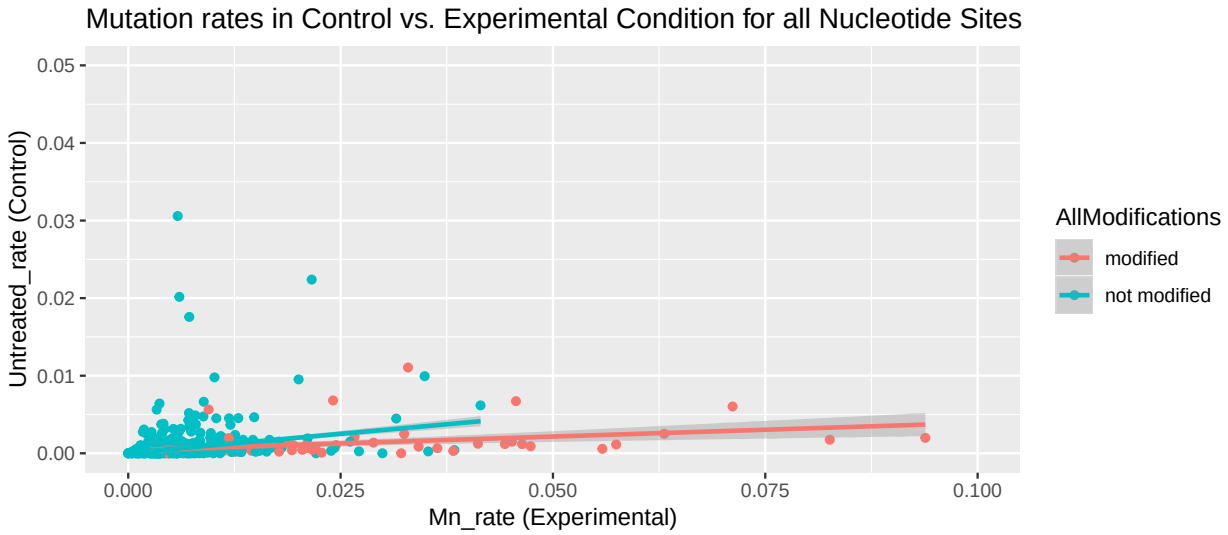
```
mods <- c("Am", "pseu", "pseu", "pseu", "Am", "pseu", "pseu", "Um", "pseu",
          "Um", "Am", "Am", "Um", "Cm", "pseu", "pseu", "pseu", "Um", "Gm",
          "Cm", "Am", "Am", "Gm", "Am", "Cm", "pseu", "Am", "Am", "Gm", "pseu",
          "Um", "Gm", "pseu", "pseu", "Am", "pseu", "Gm", "pseu", "Cm", "Um",
          "pseu", "pseu", "pseu", "pseu", "pseu", "pseu", "Gm", "pseu", "pseu",
          "pseu", "Am", "pseu", "pseu", "pseu", "pseu", "pseu", "m1acp3Y", "Cm",
          "Um", "pseum", "Gm", "pseu", "pseu", "Am", "Cm", "Um", "pseu", "Gm",
          "pseu", "m7G", "pseu", "Am", "pseu", "Cm", "Um", "m6A", "m62A", "m62A")

nums <- c(27, 44, 46, 93, 99, 105, 109, 116, 119, 121, 159, 166, 172, 174, 210,
          218, 406, 428, 436, 462, 468, 484, 509, 512, 517, 572, 576, 590, 601,
          609, 627, 644, 659, 651, 668, 681, 683, 686, 797, 799, 801, 814, 815,
          822, 863, 866, 867, 918, 966, 1004, 1031, 1056, 1081, 1174, 1238, 1244,
          1248, 1272, 1288, 1326, 1328, 1347, 1367, 1383, 1391, 1442, 1445, 1490,
          1625, 1639, 1643, 1678, 1692, 1703, 1804, 1832, 1850, 1851)

#intersperse modifications with labels of "none" to create a vector with same size as dataset
modvector <- rep("none", dim(dataset)[1])
allmods <- rep("not modified", dim(dataset)[1])
k <- 1
for (h in nums) {
  modvector[h] <- mods[k]
  allmods[h] <- "modified"
  k <- k + 1
}

#bind vector of modifications onto the beginning of our dataset
new <- cbind(modvector, dataset)
new <- cbind(allmods, new)

#now, we remove the unnecessary columns (cataloguing information we are not going to use)
#also, renaming columns for clarity (...modified... -> ...Mn...)
new.filtered <- new %>% select(-ends_with("mapped_depth")) %>%
  select(-ends_with("read_depth")) %>%
  select(-starts_with("Denatured")) %>%
  select(-ends_with("profile")) %>% select(-ends_with("err")) %>%
  rename(
    Mn_mutations = Modified_mutations,
    Mn_effective_depth = Modified_effective_depth,
    Mn_rate = Modified_rate,
    Modifications = modvector,
    AllModifications = allmods
    )


#We filter out the one point with > 0.99 mutation rate
new.filtered <- new.filtered %>% filter(Mn_rate < 0.99, Untreated_rate < 0.99)
```

Data Visualization and Analysis

1.**Visualization of all points.** The following plot compares the mutation rates in the control vs. experimental stages, separated by whether or not the plot is modified. For better viewing we have restricted the

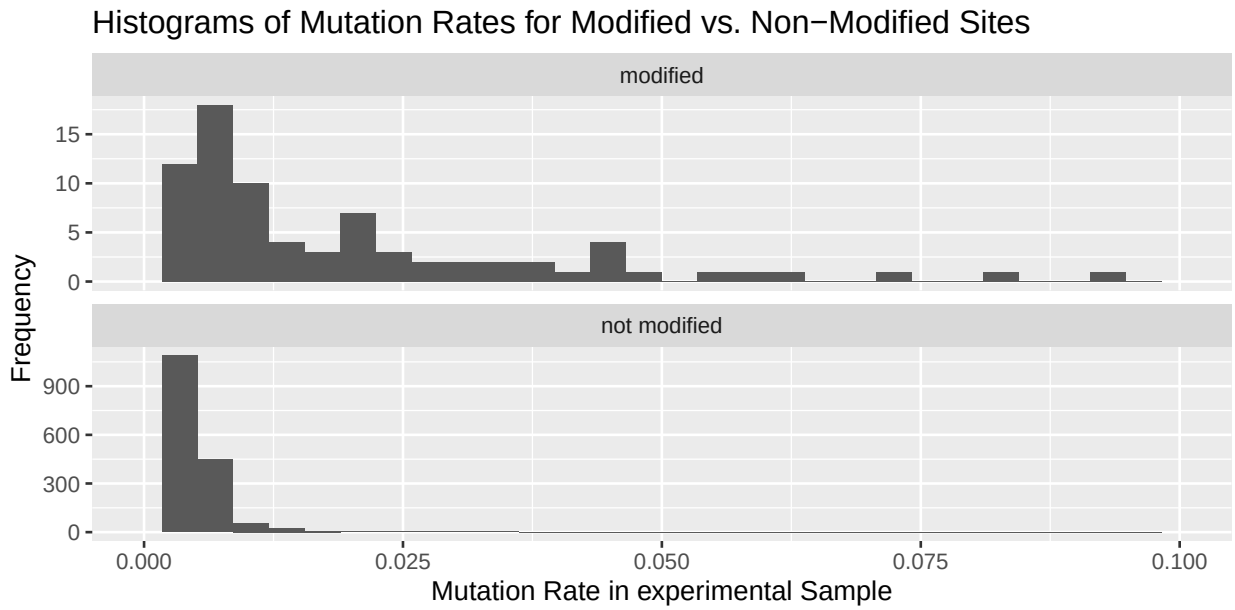
axis window (this excludes two outlier points). We can see that there is some grouping here; points with highee mutation rates in the experimental test (`Mn_rate`) seem more likely to be modified originally, while points with lower `Mn_rate` and higher `Untreated_rate` seem more likely to be unmodified sites. We'll use these two variables to try and predict the modifications.

```
plot <- ggplot(data = new.filtered, mapping = aes(x = Mn_rate, y = Untreated_rate,
                                                  col = AllModifications))

plot + geom_point() + xlim(0, 0.10) + ylim(0, 0.05) +
  geom_smooth(method = "lm") +
  labs(title = "Mutation rates in Control vs. Experimental Condition for all Nucleotide Sites",
       x = "Mn_rate (Experimental)", y = "Untreated_rate (Control)")
```



2. **Modified vs. Non-modified sites.** These histograms only consider the variable `Mn_rate` and compare the distribution of mutation rates of different sites between modified and un-modified sites. As we can see the modified sites show are much larger skew to the right.

```
new.filtered %>% ggplot(aes(x= Mn_rate)) + geom_histogram() +
  facet_wrap("AllModifications", ncol = 1, scales = "free_y") +
  xlim(0, 0.1) +
  labs(title = "Histograms of Mutation Rates for Modified vs. Non-Modified Sites",
       x = "Mutation Rate in experimental Sample", y = "Frequency")
```



We can use this to give insight to our hypothesis that there is preferential mutational incorporation at modified sites.

Notice that the distribution seems clearly non-normal (we can check this using qqplot, but the histogram does show a heavy right skew). We can use a nonparametric test to compare two means. The small p-value only tells us that there is in fact a difference in the mutation rates between the modified and unmodified sites. Notice there is also a (small) difference in mutation rates for the control condition as well.

```
#Filters the dataset into two datasets, one with modified sites and one without
modified <- new.filtered %>% filter(AllModifications == "modified")
unmodified <- new.filtered %>% filter(AllModifications == "not modified")

#Comparison of unmodified vs. modified sites in the experimental condition yields very small p-value
wilcox.test(modified$Mn_rate, unmodified$Mn_rate, alternative = "greater")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  modified$Mn_rate and unmodified$Mn_rate
## W = 123321, p-value < 2.2e-16
## alternative hypothesis: true location shift is greater than 0
```

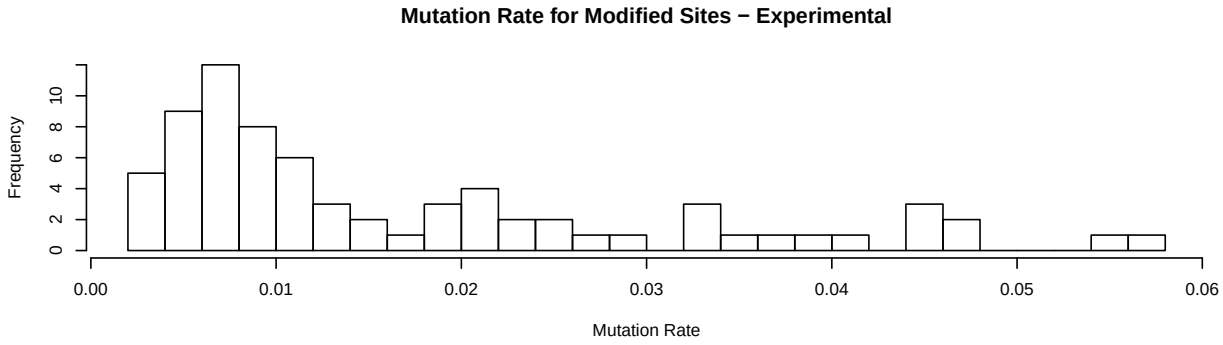
```
#Comparison in the control condition also yields small p-value
wilcox.test(unmodified$Mn_rate, unmodified$Untreated_rate, alternative = "greater")
```

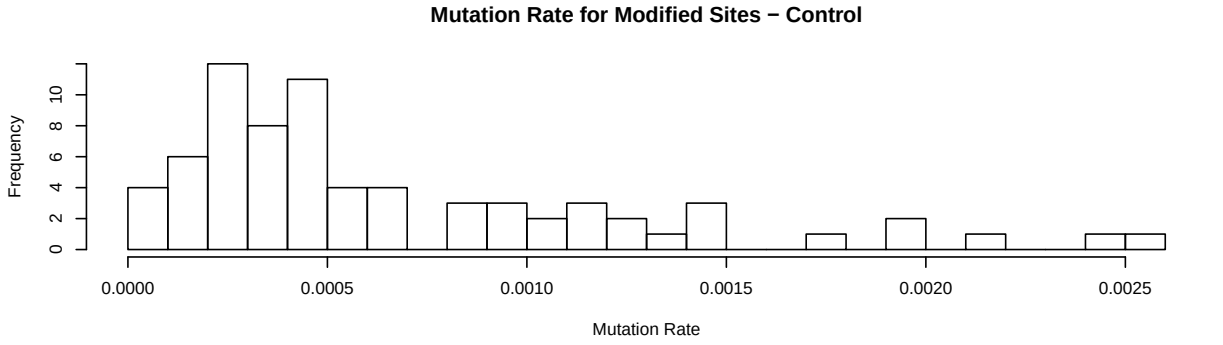
```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  unmodified$Mn_rate and unmodified$Untreated_rate
## W = 3146862, p-value < 2.2e-16
## alternative hypothesis: true location shift is greater than 0
```

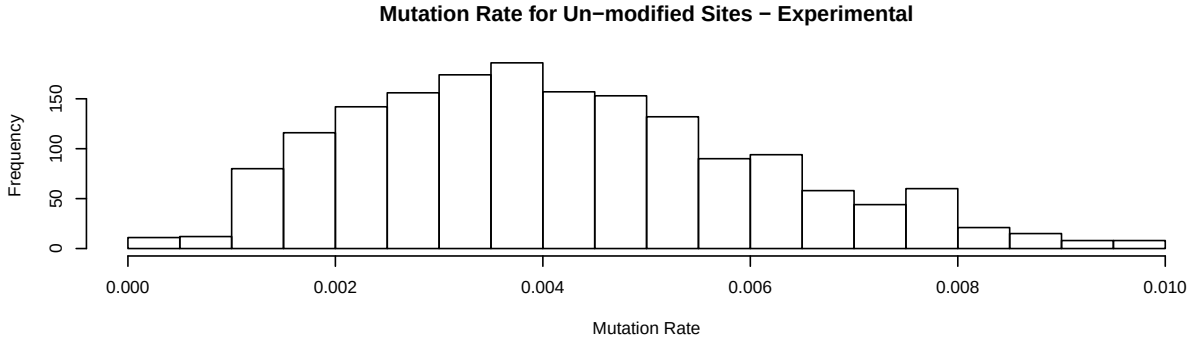
3. **Comparison of Control and Experimental conditions for both modification types.** The following histograms allow us to examine each category more deeply. Here the scales are different and

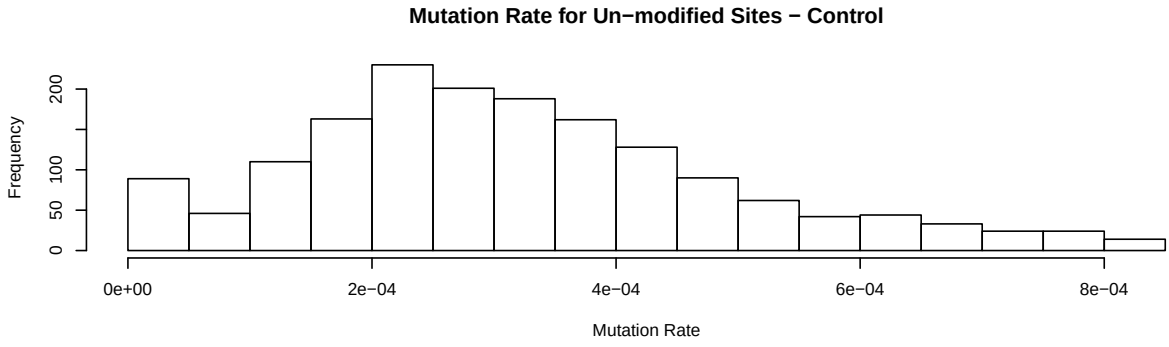
outliers are removed for better viewing. At a closer scale, it seems that the distribution of mutation rates for unmodified sites are still right-skewed, but show more normality than with modified sites.

```
#Looking at distribution of mutation rate for modified regions
modified_experimental_outliers <- boxplot(modified$Mn_rate, plot = FALSE)$out
modified_experimental_outliersrm <- modified[-which(modified$Mn_rate
                                                    %in% modified_experimental_outliers),]


modified_control_outliers <- boxplot(modified$Untreated_rate, plot = FALSE)$out
modified_control_outliersrm <- modified[-which(modified$Untreated_rate
                                               %in% modified_control_outliers),]


#Looking at distribution of mutation rate for unmodified regions
unmodified_experimental_outliers <- boxplot(unmodified$Mn_rate, plot = FALSE)$out

unmodified_experimental_outliersrm <- unmodified[-which(unmodified$Mn_rate
                                                        %in% unmodified_experimental_outliers),]


unmodified_control_outliers <- boxplot(unmodified$Untreated_rate, plot = FALSE)$out
unmodified_control_outliersrm <- unmodified[-which(unmodified$Untreated_rate
                                                   %in% unmodified_control_outliers),]




par(mfrow = c(4,1))
hist(modified_experimental_outliersrm$Mn_rate, nclass = 20,
     main = "Mutation Rate for Modified Sites - Experimental",, xlab = "Mutation Rate")
hist(modified_control_outliersrm$Untreated_rate, nclass = 20,
     main = "Mutation Rate for Modified Sites - Control",, xlab = "Mutation Rate")
hist(unmodified_experimental_outliersrm$Mn_rate, nclass = 20,
     main = "Mutation Rate for Un-modified Sites - Experimental",, xlab = "Mutation Rate")
hist(unmodified_control_outliersrm$Untreated_rate, nclass = 20,
     main = "Mutation Rate for Un-modified Sites - Control",, xlab = "Mutation Rate")
```





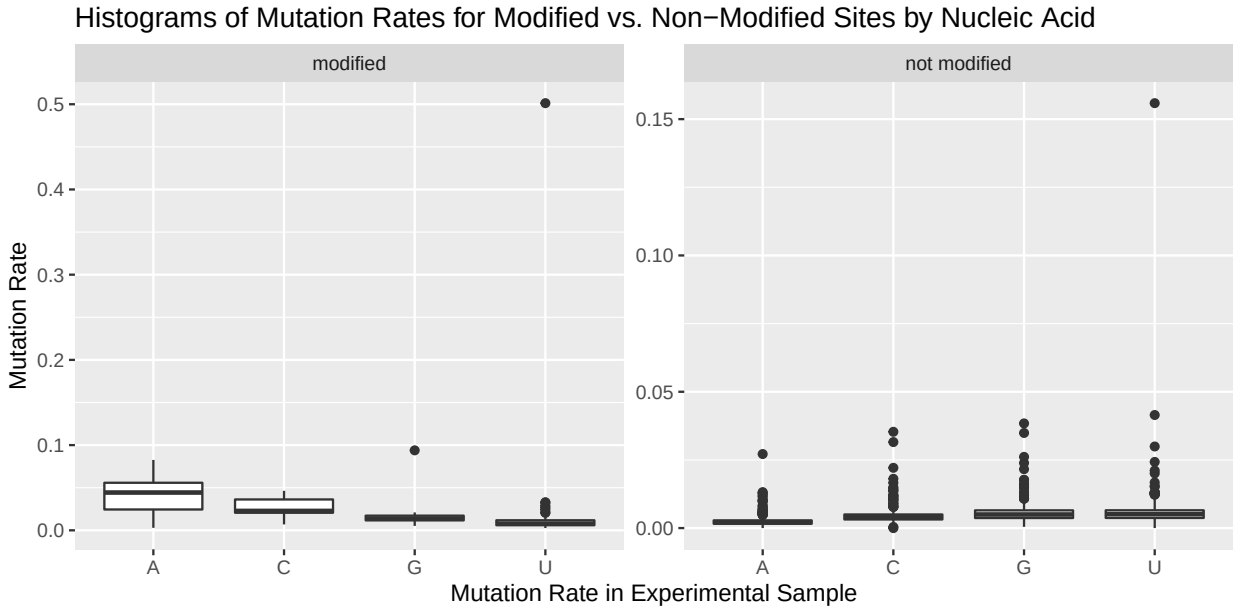




4. **Mutation Rate by Amino Acid** Another one of our hypotheses is that the distribution of mutations is different for each individual nucleic acid. The following histograms give us a bit more insight into this prediction. It seems that especially at non-modified sites, mutation rates differ among amino acids. This might indicate that classification might be better acheived if we separate by amino acid; over, we

also need to consider that there are a very small number of modified site data points in total.

```
new.filtered  %>% ggplot(aes(x= Sequence, y = Mn_rate)) + geom_boxplot() +
  facet_wrap("AllModifications", scale = "free_y") +
  #xlim(0, 0.1) +
  labs(title = "Histograms of Mutation Rates for Modified vs. Non-Modified Sites by Nucleic Acid",
       x = "Mutation Rate in Experimental Sample", y = "Mutation Rate")
```



A non-parametric test (Kruskal-Wallis) to compare the four samples confirms that the four samples (four amino acids) do not come from the same distribution.

```
#Small p-value indicates that the groups do not come from the same distribution
kruskal.test(Mn_rate ~ Sequence, data = new.filtered)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Mn_rate by Sequence
## Kruskal-Wallis chi-squared = 527.71, df = 3, p-value < 2.2e-16
```

Logistic Regression

Dataset modification. Here we attempt to predict whether each site is modified or unmodified using our existing variables. We've created the dataset `logdata` which assigns value 1 as "modified" and 0 as "un-modified).

```
logdata <- new.filtered %>% select(AllModifications, Mn_rate, Untreated_rate)

allmods <- gsub("not modified", 0, logdata$AllModifications)
allmods <- gsub("modified", 1, allmods)

logdata$AllModifications <- as.numeric(allmods)
```

Binomial logistic Regression. We regress on `Mn_rate` and `Untreated_rate`

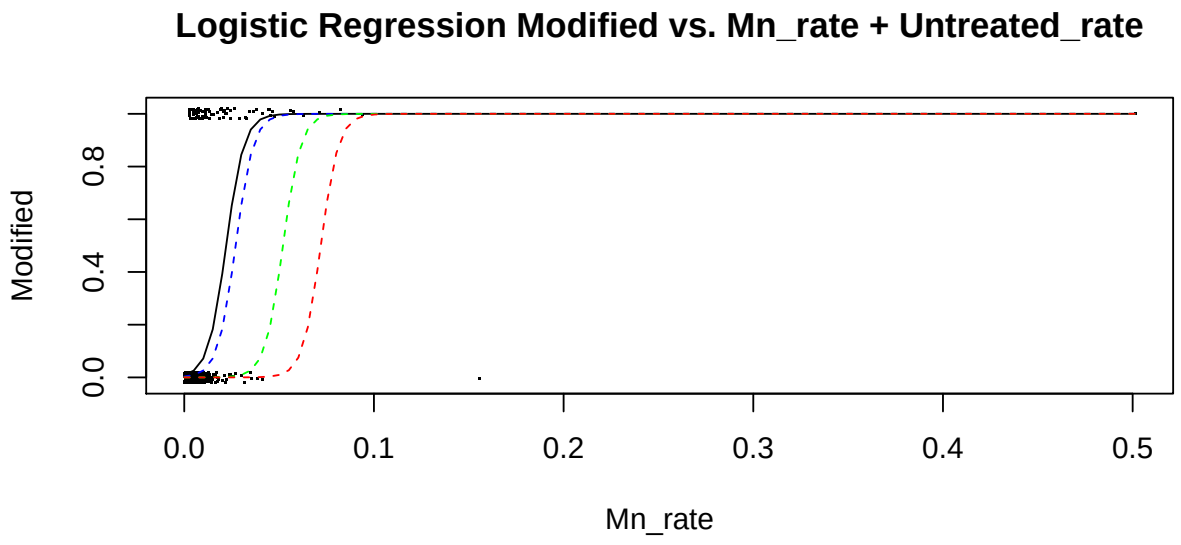
```
hm1 <- glm(AllModifications ~ Mn_rate + Untreated_rate, family = binomial, logdata)

faraway::sumary(hm1)
```

```
##                   Estimate Std. Error z value  Pr(>|z|)
## (Intercept)       -4.69078    0.21684 -21.632 < 2.2e-16
## Mn_rate          212.42196   21.09683  10.069 < 2.2e-16
## Untreated_rate -211.92741   64.88906  -3.266  0.001091
##
## n = 1868 p = 3
## Deviance = 441.51275 Null Deviance = 648.13945 (Difference = 206.62670)
```

Notice that the difference between deviance and null deviance is very large, and likely would not come from a chi-square distribution with 2 degrees of freedom (if we calculate the probability, it is almost 0). Also, the coefficients for both variables seem to be statistically significant.

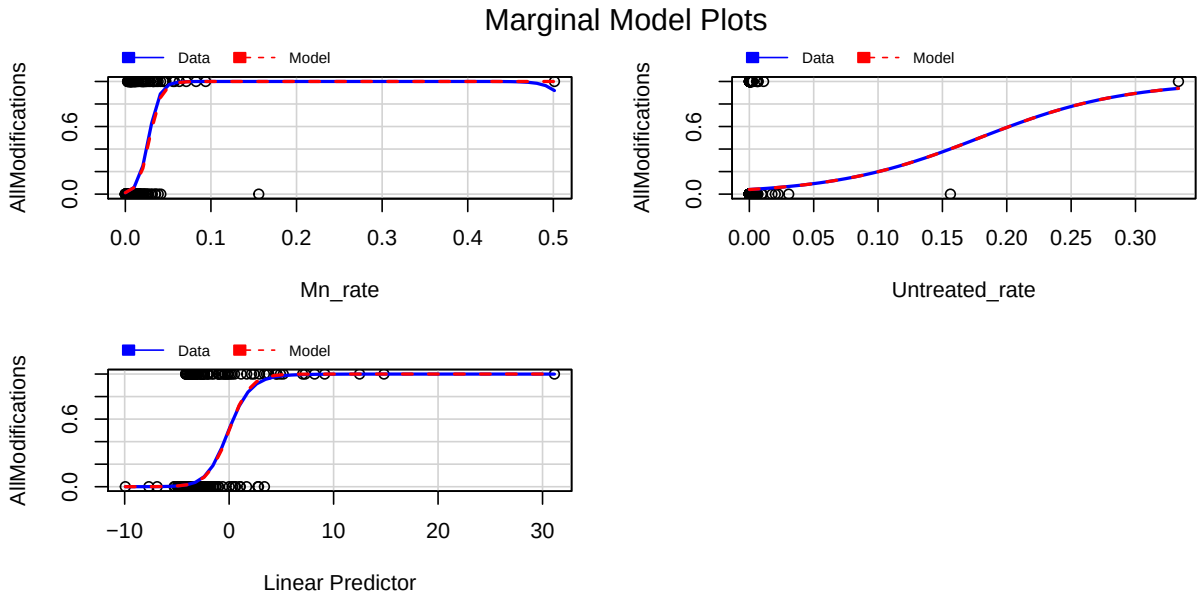
Visualizing the regression. The following curves show predictions for different values of the control mutation rate; as Untreated_rate increases, the probability that the point is *not* modified increases. However, we can see that this predictive model almost always predicts that the point is un-modified.

```
beta <- coef(hm1)

plot(jitter(AllModifications,0.1) ~ jitter(Mn_rate),
     data=logdata, xlab="Mn_rate", ylab="Modified",pch=".",
     main = "Logistic Regression Modified vs. Mn_rate + Untreated_rate")

curve(ilogit(beta[1] + beta[2]*x + beta[3]*0), add = TRUE)
curve(ilogit(beta[1] + beta[2]*x + beta[3]*0.005), add = TRUE, lty = 2, col = "blue")
curve(ilogit(beta[1] + beta[2]*x + beta[3]*0.03), add = TRUE, lty = 2, col = "green")
curve(ilogit(beta[1] + beta[2]*x + beta[3]*0.05), add = TRUE, lty = 2, col = "red")
```



Marginal Model Plots. Instead of looking at residuals (since all residuals will be either 0 or 1), we'll look at the marginal model plots for this model. It seems that the data and the model agree for this model – in fact, they seem to be exactly the same. This is probably because there are so few points in the modified category.

```
#Code adopted from 4/13 notes
car::marginalModelPlots(hm1)
```



Prediction. It looks like the probability of correct prediction for unmodified sites is very high, while the probability of correct prediction for modified sites is not (about 33 percent when our threshold is 0.5). This does make sense because there are so many more data points with unmodified sites.

```
#Code adopted from 4/13 notes
logistic <- function(x){1/(1+exp(-x))}

diagnostics <- mutate(logdata, residuals=residuals(hm1), linpred=predict(hm1),
                      predprob = logistic(linpred), check=predict(hm1, type="response"))
diagnostics <- mutate(diagnostics, predout = ifelse(predprob < 0.5, "no", "yes"))

xtabs( ~ AllModifications + predout, diagnostics)
```

```
##                 predout
## AllModifications   no  yes
##                0 1779   11
##                1   56   22
```