Stats744: HW6

Illustrating Statistical Inference

Data: Thesis Again...

As part of my thesis I have information about substitutions and their variation across 6 *E. coli* genomes. The data is binary in nature: determining at each site in the genome if there is a substitution (1), or there is not (0). This was also broken down into protein coding and non-protein coding sections of the genome. We expect that non-protein coding segments of the genome to be undergoing more substitutions because they are typically "less important" than protein-coding segments of the genome.

In bacteria genomic reorganization such as rearrangements, inversions, and duplications happen often. This means that genes change position in bacterial genomes often. Therefore, it is possible for one gene to be present at the beginning of the genome in one bacterial strain, and the same gene could be found towards the end of the genome in another bacteria.

My data takes this genome reorganization into account by adding a phylogenetic component to this analysis. I reconstructed both the sequence and the genomic position of that sequence in the ancestors of the 6 *E. coli* genomes. This allows for a gene, or more generally a segment of sequence to be found in various genomic positions between the different strains of *E. coli*. This also means that for a particular site in the genome, I have 11 data points, one for each of the nodes in the phylogenetic tree of these bacteria. a.k.a millions of data points. #**INSERT LINK TO DATA HERE** #This data can be found here.

Again, this takes too read in and to run on my laptop so I have condensed the data into total number of substitutions over 10,000bp segments of the genome.

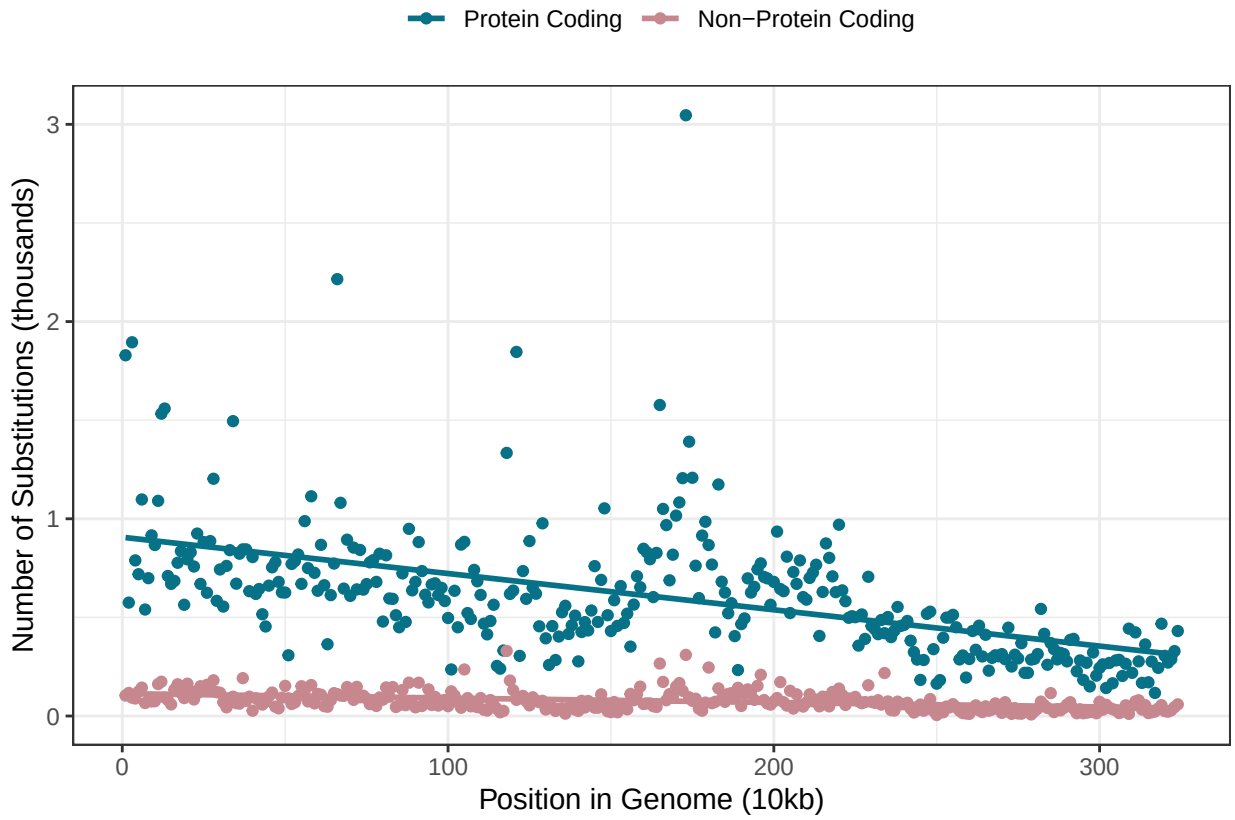
#**INSERT LINK TO DATA HERE** #This binned data can be found here.

```
chrom_datafile <- "binned_dat.csv"
chrom_data <- read.csv(chrom_datafile)
chrom_data <- chrom_data[, -1]
colnames(chrom_data) <- c("position", "cod_sub", "ncod_sub", "cod_tot",
    "ncod_tot", "cod_weight", "ncod_weight")
# subset data
chrom_data_raw <- chrom_data[, c("position", "cod_sub", "ncod_sub")]

# tidy df
chrom_data_raw <- chrom_data_raw %>% gather(genom_cat, val, cod_sub:ncod_sub) %>%
    mutate(genom_cat = gsub("genom_cat", "", genom_cat)) %>% arrange(position,
    genom_cat)

# scale data to look better on graph
chrom_data_raw$position <- chrom_data_raw$position/10000
```

```
chrom_data_raw$val <- chrom_data_raw$val/1000
```

Graph 1

```
# base graph
base_g <- ggplot(data = chrom_data_raw, aes(x = position, y = val, color = genom_cat))
# defining colours
cols <- c("#077187", "#C6878F")

rg <- (base_g + geom_point() + geom_smooth(method = "lm", se = FALSE) +
    theme_bw() + theme(legend.position = "top") + labs(x = "Position in Genome (10kb)",
    y = "Number of Substitutions (thousands)") + scale_color_manual(values = cols,
    name = "", labels = c("Protein Coding", "Non-Protein Coding")))
rg
```



Discription of Graph 1:

In this graphic my main goal was to show the difference in distribution of protein coding substitutions and non-protein coding substitutions. The above graph is looking at the "raw" total counts of substitutions in each 10,000 bp (or 10kb) region of the genome. According to the Cleveland hierarchy, we are best at identifying positions along a common scale. Here

I am showing how protein coding and non-protein coding substitutions vary with genomic position (a common scale). I believe that the font size and aspect ratio are all fine. I chose to re-scale the data so that the numbers along the x and y axis look "prettier". Since the genome and number of substitutions are integers, they can be scaled while still having the graphs make sense. This also necessitates the use of units so the viwer can determine the magnitude of the values on the graph. Since it is possible to have zero substitutions at a particular site as well as having the genome begin at zero, anchoring the graph at zero in both the x and y direction is appropriate and accurate. I adjusted the legend to be at the top of the graph (instead of the default to the right of the graph) to avoide unecessary white space. I also ensured that the protein coding and non-protein coding labels matched the way the data is organized in the grap, with protein coding on top and non-protein coding on the bottom, and therefore the legend being protein coding and then non-protein coding.

I also added a trend line to help guide the viewer in seeing how each of these genomic classifications varies with genomic position. It is clear that the protein coding substitutions have a negative trend, indicating that there are higher numbers of substitutions near the beginning of the genome than at the end. Biologically this is curious, we expect there to be more essential genes located near the beginning of the genome, and these genes should therefore have fewer substitutions because any mutation could be detrimental to the organism. For the non-protein coding genes it is less clear if this trendline is positive, negative or relatively flat.

direct labels? bars?

Proportional Data:

I thought that the above graph was deceiving because it makes it look like there are more protein coding substitutions than non-protein coding substitutions. However, there is a disproportional amount of protein coding sites in bacterial genomes compared to non-protein coding sites, bacterial genomes are almost exclusivly protein coding! So, I decided to divide the total number of protein coding or non-protein coding substitutions in a 10,000bp segment by the total number of protein coding or non-protein coding sites respectively.

```
# subset data
chrom_data_weight <- chrom_data[, c("position", "cod_weight", "ncod_weight")]

# tidy df
chrom_data_weight <- chrom_data_weight %>% gather(genom_cat, val, cod_weight:ncod_weight
    mutate(genom_cat = gsub("genom_cat", "", genom_cat)) %>% arrange(position,
    genom_cat)

# scale data to look better on graph
chrom_data_weight$position <- chrom_data_weight$position/10000
# chrom_data_weight$val <- chrom_data_weight$val / 1000
```

```
#base graph
base_g <- ggplot(data=chrom_data_weight, aes(x=position, y=val, color= genom_cat))
```

```r
#defining colours
cols <- c("#077187","#C6878F")

wg <- (base_g
       + geom_point()
       + geom_smooth(method = 'lm', se = FALSE)
       + theme_bw()
       + theme(legend.position="top")
       + labs(x = "Position in Genome (10kb)", y = "Substitutions per site")
#       + scale_color_discrete(values = cols, name="Genomic Category",labels = c("Prot
       + scale_color_manual(values = cols, name="",labels = c("Protein Coding","Non-Prot
)
wg
```