BCB546X R-Assignment

Code

Colton McNinch

10/16/2017

Background

In this `R-Assignment-Notebook.Rmd` file I will explain the steps/processes I undertook to complete the R Assignment in BCB546X. The ultimate goal of this assignment was to combine two text files, `fang_et_al_genotypes.txt` & `snp_position.txt` and then parse their consolidated information into a total of 40 separate files based on various criteria (these output files can be found in the `maize_genotype_files/` and `teosinte_genotype_files/` directories.) Additionally, various visualizations were made with the merged data from the two files.

To accomplish this goal I have highlighted each key step in the process. These steps are as follows:

- 1. Loading the data (see the “Load data” section)
- 2. Inspecting the data (see the “Inspect data” section)
- 3. Processing the data (see the “Process data” section)
- 4. Visualizing the data (see the “Visualize data” section)

Load data

- ### Step 1: Load required packages

Hide

```
library(dplyr)
library(tidyr)
library(pryr)
library(tibble)
library(ggplot2)
```

- ### Step 2: Read data files

Hide

```
genotypes <- read.delim("~/Desktop/BCB546X/R-Assignment/fang_et_al_genotypes.txt")
SNP_Positions <- read.delim("~/Desktop/BCB546X/R-Assignment/snp_position.txt") %>%
                    select(SNP_ID, Chromosome, Position) %>%
                    mutate(SNP_ID = as.character(SNP_ID))
```

Inspect data

- ### Step 1: Inspect file sizes

Hide

```
object_size(genotypes)
```

```
11.7 MB
```

Hide

```
object_size(SNP_Positions)
```

```
132 kB
```

- **Step 2: Inspect column totals**

Hide

```
ncol(genotypes)
```

```
[1] 986
```

Hide

```
ncol(SNP_Positions)
```

```
[1] 3
```

- **Step 3: Inspect row totals**

Hide

```
nrow(genotypes)
```

```
[1] 2782
```

Hide

```
nrow(SNP_Positions)
```

```
[1] 983
```

- **Step 4: Inspect a subset of each file**

Hide

```
select(genotypes, 1:10) %>%
  slice(1:10)
select(SNP_Positions, 1:3) %>%
  slice(1:10)
```

Process data

- **Step 1: Create a transposed maize genotype file with SNP info**

Hide

```
maize_target <- c("ZMMIL", "ZMMLR", "ZMMMR")
maize_genotypes <- select(genotypes, -JG_OTU) %>%
                      filter(Group %in% maize_target) %>%
                      t()
colnames(maize_genotypes) <- as.character(unlist(maize_genotypes[1,]))
maize_genotypes = as.data.frame(maize_genotypes[-(1:2), ]) %>%
                    rownames_to_column(var = "SNP_ID")
maize_genotypes <- left_join(SNP_Positions, maize_genotypes, by = "SNP_ID") %>%
                      filter(Chromosome != "unknown") %>%
                      filter(Chromosome != "multiple")
maize_genotypes[, 4:ncol(maize_genotypes)] <- as.character(unlist(maize_genotypes[, 4:nc
ol(maize_genotypes)]))
maize_genotypes[maize_genotypes == "multiple"] <- NA
maize_genotypes$Position <- as.numeric(as.character(maize_genotypes$Position))
```

- **Step 2: Create a transposed teosinte genotype file with SNP info**

Hide

```
teosinte_target <- c("ZMPBA", "ZMPIL", "ZMPJA")
teosinte_genotypes <- select(genotypes, -JG_OTU) %>%
                        filter(Group %in% teosinte_target) %>%
                        t()
colnames(teosinte_genotypes) <- as.character(unlist(teosinte_genotypes[1,]))
teosinte_genotypes = as.data.frame(teosinte_genotypes[-(1:2), ]) %>%
                        rownames_to_column(var = "SNP_ID")
teosinte_genotypes <- left_join(SNP_Positions, teosinte_genotypes, by = "SNP_ID") %>%
                        filter(Chromosome != "unknown") %>%
                        filter(Chromosome != "multiple")
teosinte_genotypes[, 4:ncol(teosinte_genotypes)] <- as.character(unlist(teosinte_genotyp
es[, 4:ncol(teosinte_genotypes)]))
teosinte_genotypes[teosinte_genotypes == "multiple"] <- NA
teosinte_genotypes$Position <- as.numeric(as.character(teosinte_genotypes$Position))
```

- **Step 3: Create 10 maize genotype files (1 for each Chr.) with SNPs ordered in increasing position**

Hide

```
chrs <- unique(maize_genotypes$Chromosome)
for (i in seq_along(chrs)){
  file <- filter(maize_genotypes, Chromosome == chrs[i]) %>%
            mutate(Position = as.numeric(Position)) %>%
            arrange(Position)
  write.table(file, paste("maize_genotype_files/", "Chr", chrs[i], "-", "MaizeGenotypes-
", "Ascending", ".txt", sep = ""))
}
```

- **Step 4: Create 10 maize genotype files (1 for each Chr.) with SNPs ordered in decreasing position and missing values replaced with -**

Hide

```
chrs <- unique(maize_genotypes$Chromosome)
for (i in seq_along(chrs)){
  file <- filter(maize_genotypes, Chromosome == chrs[i]) %>%
            mutate(Position = as.numeric(Position)) %>%
            arrange(-Position)
  file[ , 4:ncol(file)] <- lapply(file[ , 4:ncol(file)], function(x) gsub(pattern = "?",

                                                                          replacement =
"-",
                                                                          fixed = TRUE,
                                                                          x = x))
  write.table(file, paste("maize_genotype_files/", "Chr", chrs[i], "-", "MaizeGenotypes-
", "Descending", ".txt", sep = ""))
}
```

- **Step 5: Create 10 teosinte genotype files (1 for each Chr.) with SNPs ordered in increasing position**

Hide

```
chrs <- unique(teosinte_genotypes$Chromosome)
for (i in seq_along(chrs)){
  file <- filter(teosinte_genotypes, Chromosome == chrs[i]) %>%
            mutate(Position = as.numeric(Position)) %>%
            arrange(Position)
  write.table(file, paste("teosinte_genotype_files/", "Chr", chrs[i], "-", "TeosinteGeno
types-", "Ascending", ".txt", sep = ""))
}
```

- **Step 6: Create 10 teosinte genotype files (1 for each Chr.) with SNPs ordered in decreasing position and missing values replaced with -**

Hide

```
chrs <- unique(teosinte_genotypes$Chromosome)
for (i in seq_along(chrs)){
  file <- filter(teosinte_genotypes, Chromosome == chrs[i]) %>%
            mutate(Position = as.numeric(Position)) %>%
            arrange(-Position)
  file[ , 4:ncol(file)] <- lapply(file[ , 4:ncol(file)], function(x) gsub(pattern = "?",

                                                                          replacement =
"-",
                                                                          fixed = TRUE,
                                                                          x = x))
  write.table(file, paste("teosinte_genotype_files/", "Chr", chrs[i], "-", "TeosinteGeno
types-", "Descending", ".txt", sep = ""))
}
```

Visualize data

- **Step 1: Create a data frame for SNPs per chromosome visualization**

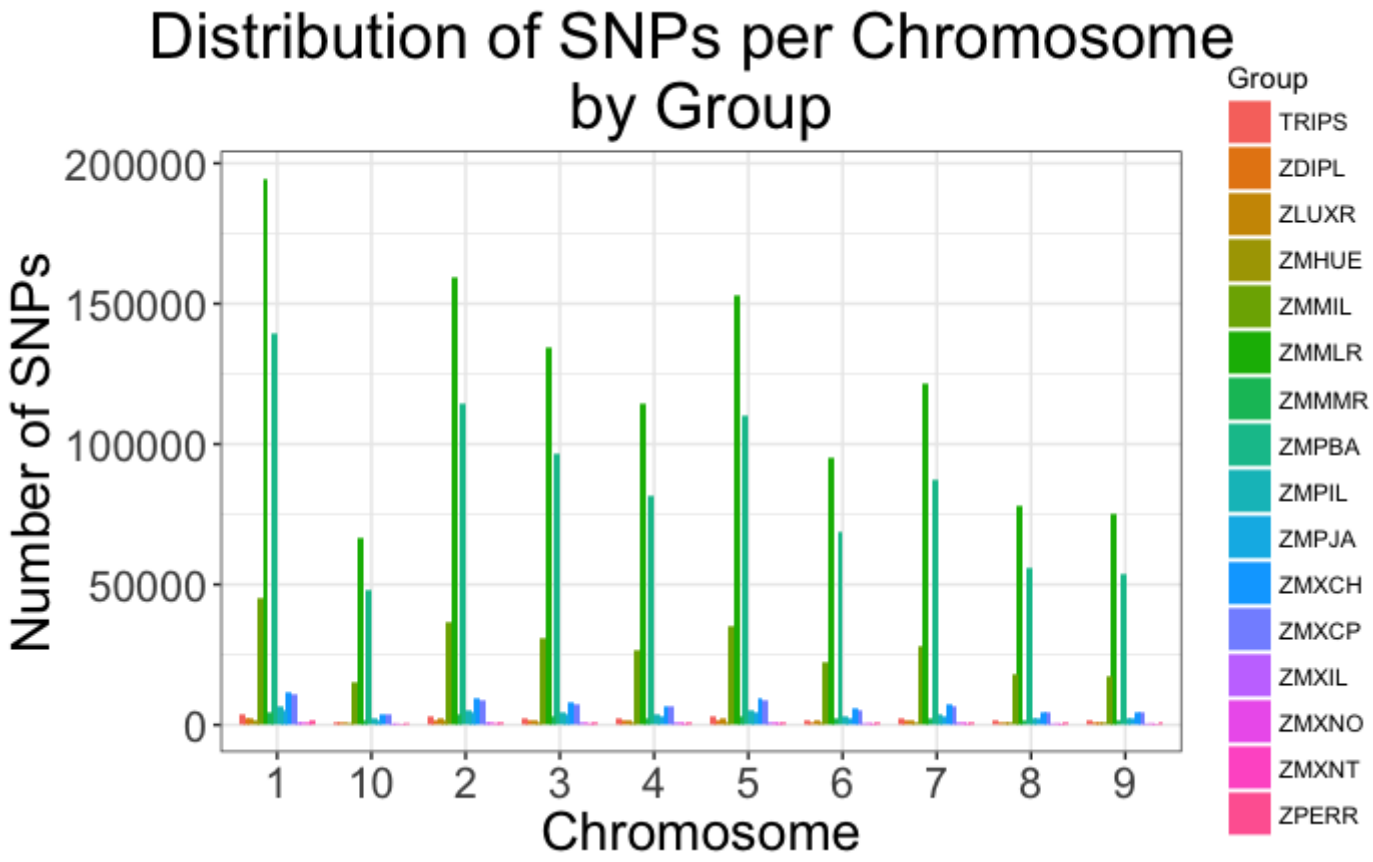
Hide

```
SNPdf <- select(genotypes, -JG_OTU) %>%
                      t()
colnames(SNPdf) <- as.character(unlist(SNPdf[1,]))
SNPdf = as.data.frame(SNPdf[-1, ]) %>%
          rownames_to_column(var = "SNP_ID")
SNPdf <- left_join(SNP_Positions, SNPdf, by = "SNP_ID") %>%
            filter(Chromosome != "unknown") %>%
            filter(Chromosome != "multiple")
SNPdf[, 4:ncol(SNPdf)] <- as.character(unlist(SNPdf[, 4:ncol(SNPdf)]))
SNPdf[SNPdf == "multiple"] <- NA
SNPdf$Position <- as.numeric(as.character(SNPdf$Position))
SNPdf <- gather(SNPdf, "Sample_ID", "Genotype_Call", 4:ncol(SNPdf)) %>%
            mutate(Sample_ID = as.factor(Sample_ID))
SNPdf$Position <- as.numeric(as.character(SNPdf$Position))
SNPdf <- left_join(SNPdf, select(genotypes, Sample_ID, Group), by = "Sample_ID")
SNPCounts <- group_by(SNPdf, Chromosome, Group) %>%
             count()
```

- **Step 2: Create a bar graph with the frequency of SNPs per chromosome and group plotted**

Hide

```
ggplot(SNPCounts, aes(x = Chromosome, y = n, fill = Group)) +
  geom_bar(stat = "identity", position = position_dodge(0.8), width = 1) +
  theme_bw() +
  theme(axis.text = element_text(size = 16),
        axis.title = element_text(size = 20),
        plot.title = element_text(size = 24, hjust = 0.5)) +
  labs(y = "Number of SNPs", title = "Distribution of SNPs per Chromosome \nby Group")
```



- **Step 3: Create a data frame for SNP call type (i.e. homozygous/heterozygous/missing) visualization**

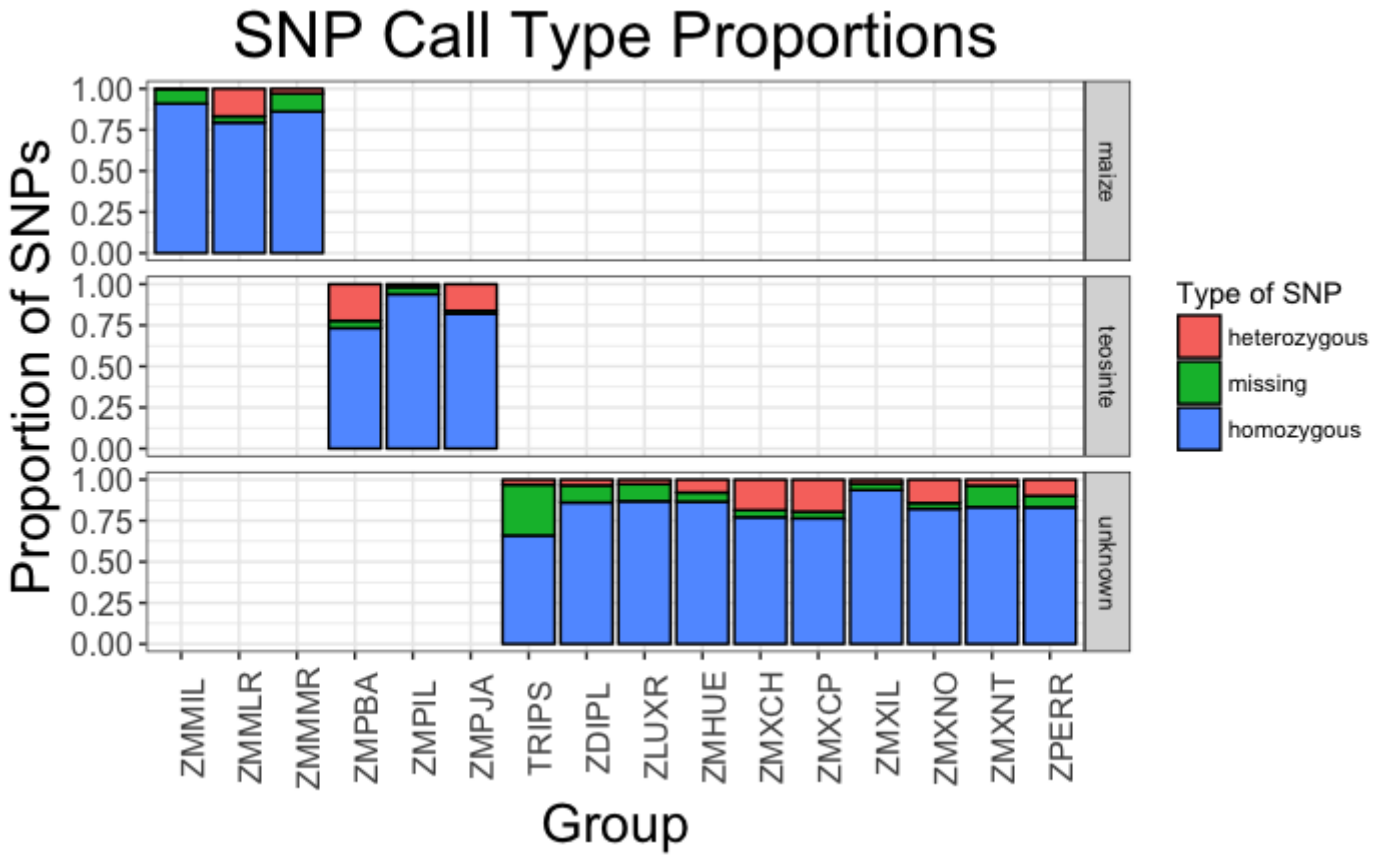
Hide

```
SNPdf <- mutate(SNPdf, Genotype_Call = replace(Genotype_Call, Genotype_Call == "?/?", "N
A"))
SNPdf <- SNPdf %>% mutate(SNP_Call_Type = ifelse(Genotype_Call == "A/A", "homozygous",
                                                 ifelse(Genotype_Call == "T/T", "homozy
gous",
                                                 ifelse(Genotype_Call == "C/C", "homozy
gous",
                                                 ifelse(Genotype_Call == "G/G", "homozy
gous",
                                                 ifelse(Genotype_Call == "NA", "missin
g", "heterozygous"))))))
SNPdf <- SNPdf %>% mutate(Species_ID = ifelse(Group == "ZMMIL", "maize",
                                              ifelse(Group == "ZMMLR", "maize",
                                              ifelse(Group == "ZMMMR", "maize",
                                              ifelse(Group == "ZMPBA", "teosinte",
                                              ifelse(Group == "ZMPIL", "teosinte",
                                              ifelse(Group == "ZMPJA", "teosinte", "unkn
own")))))))
SNPdf <- SNPdf %>% arrange(Group, Species_ID)
SNP_Type_Summary <- group_by(SNPdf, Group, Species_ID, SNP_Call_Type) %>%
                      count() %>%
                      ungroup(Group, Species_ID, SNP_Call_Type) %>%
                      spread(SNP_Call_Type, n) %>%
                      mutate(total = heterozygous + homozygous + missing) %>%
                      mutate(het_proportion = heterozygous/total,
                             homo_proportion = homozygous/total,
                             missing_proportion = missing/total) %>%
                      gather("Proportion_Type", "Proportion", 7:9) %>%
                      mutate(Proportion_Type = ifelse(Proportion_Type == "het_proportio
n", "heterozygous",
                                                      ifelse(Proportion_Type == "homo_pr
oportion", "homozygous",
                                                      ifelse(Proportion_Type == "missing
_proportion", "missing","NA"))))
#reorder factors to make later graph more appealing
SNP_Type_Summary$Proportion_Type <- factor(SNP_Type_Summary$Proportion_Type,
                                           levels = c( "heterozygous", "missing", "homoz
ygous"))
SNP_Type_Summary$Group <- factor(SNP_Type_Summary$Group,
                                 levels = c("ZMMIL", "ZMMLR", "ZMMMR", "ZMPBA", "ZMPIL",
 "ZMPJA",
                                            "TRIPS", "ZDIPL", "ZLUXR", "ZMHUE", "ZMXCH",
 "ZMXCP", "ZMXIL",
                                            "ZMXNO", "ZMXNT", "ZPERR"))
```

- **Step 4: Create a bar graph which reveals the proportion of total SNPs comprised of “heterozygous”, “homozygous”, or “missing” SNPs**

Hide

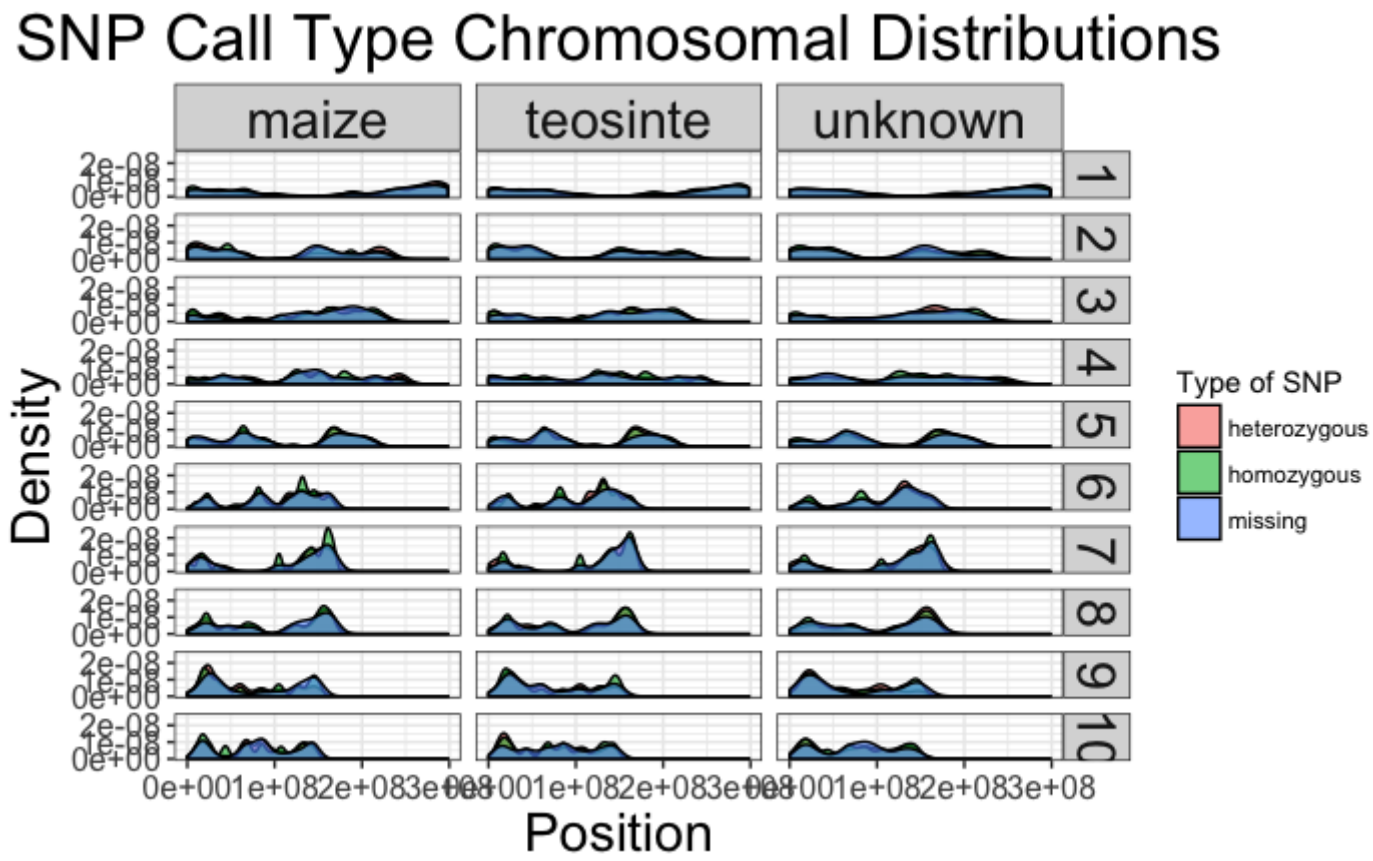
```
ggplot(SNP_Type_Summary, aes(x = Group, y = Proportion, fill = Proportion_Type)) +
  geom_bar(stat = "identity", color = "black") +
  facet_grid(Species_ID ~ .) +
  theme_bw() +
  theme(axis.text.x = element_text(size = 12, angle = 90),
        axis.text.y = element_text(size = 12),
        axis.title = element_text(size = 20),
        plot.title = element_text(size = 24, hjust = 0.5)) +
  labs(y = "Proportion of SNPs",
       title = "SNP Call Type Proportions",
       fill = "Type of SNP")
```



- **Step 5: Visualize where along each chromosome the three different SNP call types are occurring by creating a facetted density curve plot**

Hide

```
#order Chromsomes for better visualization
SNPdf$Chromosome <- factor(SNPdf$Chromosome,
                           levels = c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10"))
ggplot(SNPdf) +
  geom_density(aes(x = Position, fill = SNP_Call_Type), alpha = 0.6) +
  facet_grid(Chromosome ~ Species_ID) +
  theme_bw() +
  theme(axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12),
        axis.title = element_text(size = 20),
        plot.title = element_text(size = 24, hjust = 0.5),
        strip.text = element_text(size = 20)) +
  labs(y = "Density",
       title = "SNP Call Type Chromosomal Distributions",
       fill = "Type of SNP")
```



Hide

```
NA
```