~~~
library(tidyverse)
library(tidyr)
```
~~~

# Data Inspection

```{r}
fang <- read.delim("C:/Users/risha/Downloads/fang_et_al_genotypes.txt")
fangdim=dim(fang) #return number of rows and columns
fang_info=(file.info('fang_et_al_genotypes.txt'))
```

```{r}
snp <- read.delim("C:/Users/risha/Downloads/snp_position (1).txt")
snpdim=dim(snp) #return number of rows and columns
snp_info=(file.info('snp_position.txt'))
```

# Data Processing

Use the transposed data before joining

```{r}
fang_t <- read.delim("C:/Users/risha/Downloads/transposed_genotypes.txt")

```

From the genotype data, we remove the rows containing Sample_ID and JG_OTU, and arrange the table based on the GROUP row as header to facilitate merging and sorting

```{r}
fang_t <- as.data.frame(fang_t)

new_fang<-fang_t[-c(0,1),]

colnames(new_fang)<-as.character(new_fang[1,])

new_fang<-new_fang[-c(1),]
```

Joining the genotype data with the SNP data

```{r}
merged<-merge(snp,new_fang, by.x="SNP_ID",by.y="Group", all=TRUE )
```

Removing columns other than SNP_ID, Chromosome and Position

```{r}
final <-merged[-c(2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,15)]
```

Maize Dataset

Find columns containing "ZMMIL", "ZMMLR", and "ZMMMR" and remove the rest

```{r, include=FALSE}
allcols<-colnames(final)
grep("ZMMIL",allcols)
grep("ZMMLR",allcols)
grep("ZMMMR",allcols)

```

```{r}
maize<-final[c(1,2,3,1213:2468, 2469:2495, 2496:2785)]
maize<-as.data.frame(maize)
```

We have all maize data now.

```{r}
maize_inc=maize
maize_inc[maize_inc=="?/?"]<-"?"
```

```{r}
inc_chr <- split(maize_inc, maize_inc$Chromosome)
```

sorting each list based on increasing position values

~~~
```{r,include=FALSE}
sorted_data <- lapply(inc_chr, function(df) {
  df[order(as.numeric(df$Position)),]
})

lapply(names(sorted_data), function(chr) {
  write.csv(sorted_data[[chr]], file=paste0("inc_chromosome_", chr, ".txt"), row.names=FALSE)
})
```
~~~

~~~
```{r}
maize_dec=maize
maize_dec[maize_dec=="?/?"]<-"-"
```
~~~

~~~
```{r}
dec_chr<- split(maize_dec, maize_dec$Chromosome)
```
~~~

~~~
```{r,include=FALSE}
sorted_data <- lapply(dec_chr, function(df) df[order(as.numeric(df$Position), decreasing = TRUE),])
~~~

lapply(names(sorted_data), function(chr) {write.csv(sorted_data[[chr]], file=paste0("dec_chromosome_", chr, ".txt"), row.names=FALSE, quote=FALSE)
})

Thus we have the required 20 files.

Teosinte Dataset

```{r,include=FALSE}
grep("ZMPBA",allcols)
grep("ZMPIL",allcols)
grep("ZMPJA",allcols)
```

```{r}
teosinte=final[c(1, 2, 3, 77:976, 977:1010, 1166:1206)]
teosinte<-as.data.frame(teosinte)
```

Let's generate 10 files (1 for each chromosome) with SNPs ordered based on increasing position values and with missing data encoded by this symbol: ?

```{r,include=FALSE}
teosinte_inc=teosinte
teosinte_inc[teosinte_inc=="?/?"]<-"?"
inc_tchr <- split(teosinte_inc, teosinte_inc$Chromosome)
sorted_data <- lapply(inc_tchr, function(df) {
  df[order(as.numeric(df$Position)),]
})
lapply(names(sorted_data), function(chr) {
  write.csv(sorted_data[[chr]], file=paste0("teo_inc_chromosome_", chr, ".txt"), row.names=FALSE)
})
```

Next we generate 10 files (1 for each chromosome) with SNPs ordered based on decreasing position values and with missing data encoded by this symbol: -

```{r,include=FALSE}
teosinte_dec=teosinte
teosinte_dec[teosinte_dec=="?/?"]<-"-"
dec_tchr <- split(teosinte_dec, teosinte_inc$Chromosome)
sorted_data <- lapply(dec_tchr, function(df) {
  df[order(as.numeric(df$Position), decreasing = TRUE),]
})
lapply(names(sorted_data), function(chr) {
  write.csv(sorted_data[[chr]], file=paste0("teo_dec_chromosome_", chr, ".txt"), row.names=FALSE)
})
```

Thus we have all required files.

# Part II Visualization

Step-1: Plotting total number of SNPs per chromosome

~~~
```{r}
library(dplyr)
```
~~~

~~~
```{r}
maize_snp_count <- aggregate(SNP_ID ~ Chromosome, data = maize, FUN = length)
colnames(maize_snp_count)[2] <- "SNP_Count"
maize_snp_count$Group <- "Maize"
```
~~~

~~~
```{r}
teosinte_snp_count <- aggregate(SNP_ID ~ Chromosome, data = teosinte, FUN = length)
colnames(teosinte_snp_count)[2] <- "SNP_Count"
teosinte_snp_count$Group <- "Teosinte"
```
~~~

~~~
```{r}
snp_counts <- rbind(maize_snp_count, teosinte_snp_count)
```
~~~

The chromosomes need to be sorted to be plotted.

~~~
```{r}
snp_counts$Chromosome_Numeric <- as.numeric(snp_counts$Chromosome)
snp_counts$Chromosome_Numeric[snp_counts$Chromosome == "multiple"] <- 11
snp_counts$Chromosome_Numeric[snp_counts$Chromosome == "unknown"] <- 12
~~~

~~~
unique_chromosomes <- unique(snp_counts[, c("Chromosome", "Chromosome_Numeric")])
unique_chromosomes <- unique_chromosomes[order(unique_chromosomes$Chromosome_Numeric), ]
sorted_chromosome_levels <- unique_chromosomes$Chromosome


snp_counts$Chromosome <- factor(snp_counts$Chromosome, levels = sorted_chromosome_levels)


ggplot(snp_counts, aes(x = Chromosome, y = SNP_Count, fill = Group)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Distribution of SNPs Across Chromosomes",
       x = "Chromosome",
       y = "Number of SNPs") +
  scale_fill_manual(values = c("Maize" = "#E69F00", "Teosinte" = "#56B4E9")) +
  theme_minimal() +
  theme(legend.position = "top")
```
~~~

Step-2: Identifying homozygous and heterozygous sites

~~~
```{r,include=FALSE}
library(reshape)
library(data.table)
both_long <- filter(fang, Group == "ZMMIL" | Group == "ZMMLR" | Group == "ZMMMR" | Group == "ZMPBA" | Group == "ZMPIL" | Group == "ZMPJA")
both <- melt(as.data.table(both_long), measure.vars = colnames(fang)[4:986])
~~~

~~~~
colnames(both)[4:5] <- c("SNP_ID", "Homozygous")

colnames(both)

```

```{r}

both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("A/A", "C/C", "G/G", "T/T"), TRUE, Homozygous))

both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("A/C", "A/G", "A/T", "C/G", "C/T", "G/T"), FALSE, Homozygous))

both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("?/?"), NA, Homozygous))

both <- arrange(both, Sample_ID, Group)

```



```{r}

ggplot(data = both) +

  geom_bar(mapping = aes(x = Group, fill = Homozygous), stat = "count") +

  ggtitle(label = "SNPs by  groups") +

  ylab(label = "Number of SNPs") +

  ggtitle(label = "SNPs across groups") +

  xlab(label = "Group") +

  ylab(label = "Number of SNPs") +

  theme(

    plot.title = element_text(hjust = 0.5, size = 16),  # Center the plot title

    axis.text = element_text(size = 11),

    axis.title = element_text(size = 11)

  )
~~~~

~~~
```
~~~

~~~
```{r}
ggplot(data = both) +
 geom_bar(mapping = aes(x = Sample_ID, fill = Homozygous), stat = "count") +
 ggtitle(label = "SNPs by Ordered Sample_ID") +
 ylab(label = "Number of SNPs") +
 ggtitle(label = "SNPs across sample") +
 xlab(label = "Sample") +
 ylab(label = "Number of SNPs") +
 theme(
  plot.title = element_text(hjust = 0.5, size = 16),  # Center the plot title
  axis.title = element_text(size = 12)
 )
```
~~~

we can see that the proportion of homozygous sites are higher compared to heterozygous sites.

Step-3: Own Analysis

Reshaping the original data:

~~~
```{r}
fang_long <- pivot_longer(fang,
                          cols = -c(Sample_ID, JG_OTU, Group),
                          names_to = "SNP",
                          values_to = "Genotype")
~~~

~~~
```
~~~

Let us analyse the proportion of homozygous and heterozygous sites in all of the groups

~~~
```{r}
fang_long <- mutate(fang_long,Genotype_Type = case_when( Genotype == "?" ~ "Missing", str_detect(Genotype, "/") & str_sub(Genotype, 1, 1) ==str_sub(Genotype, 3, 3) ~ "Homozygous", str_detect(Genotype, "/") & str_sub(Genotype, 1, 1) != str_sub(Genotype, 3, 3) ~ "Heterozygous",TRUE ~ "Other" ))
```
~~~

~~~
```{r,include=FALSE}
summary_data <- fang_long %>%
 filter(Genotype_Type != "Missing") %>%
 group_by(Group, Genotype_Type) %>%
 summarise(Count = n()) %>%
 mutate(Proportion = Count / sum(Count))
```
~~~

~~~
```{r}
ggplot(summary_data, aes(x = Group, y = Proportion, fill = Genotype_Type)) +
 geom_bar(stat = "identity", position = "dodge") +
 labs(title = "Proportion of Homozygous vs. Heterozygous Genotypes by Group",
    x = "Group",
    y = "Proportion",
    fill = "Genotype Type") +
 theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, hjust = 1))
```
~~~

~~~~
library(tidyverse)
library(tidyr)
```


# Data Inspection


```{r}
fang <- read.delim("C:/Users/risha/Downloads/fang_et_al_genotypes.txt")
fangdim=dim(fang) #return number of rows and columns
fang_info=(file.info('fang_et_al_genotypes.txt'))
```


```{r}
snp <- read.delim("C:/Users/risha/Downloads/snp_position (1).txt")
snpdim=dim(snp) #return number of rows and columns
snp_info=(file.info('snp_position.txt'))
```


# Data Processing


Use the transposed data before joining


```{r}
fang_t <- read.delim("C:/Users/risha/Downloads/transposed_genotypes.txt")


```
~~~~

From the genotype data, we remove the rows containing Sample_ID and JG_OTU, and arrange the table based on the GROUP row as header to facilitate merging and sorting

```{r}
fang_t <- as.data.frame(fang_t)

new_fang<-fang_t[-c(0,1),]

colnames(new_fang)<-as.character(new_fang[1,])

new_fang<-new_fang[-c(1),]
```

Joining the genotype data with the SNP data

```{r}
merged<-merge(snp,new_fang, by.x="SNP_ID",by.y="Group", all=TRUE  )
```

Removing columns other than SNP_ID, Chromosome and Position

```{r}
final <-merged[-c(2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,15)]
```

Maize Dataset

Find columns containing "ZMMIL", "ZMMLR", and "ZMMMR" and remove the rest

~~~
```{r, include=FALSE}
allcols<-colnames(final)
grep("ZMMIL",allcols)
grep("ZMMLR",allcols)
grep("ZMMMR",allcols)

```
~~~

~~~
```{r}
maize<-final[c(1,2,3,1213:2468, 2469:2495, 2496:2785)]
maize<-as.data.frame(maize)
```
~~~

We have all maize data now.

~~~
```{r}
maize_inc=maize
maize_inc[maize_inc=="?/?"]<-"?"
```
~~~

~~~
```{r}
inc_chr <- split(maize_inc, maize_inc$Chromosome)
```
~~~

sorting each list based on increasing position values

~~~
```{r,include=FALSE}
sorted_data <- lapply(inc_chr, function(df) {
  df[order(as.numeric(df$Position)),]
})

lapply(names(sorted_data), function(chr) {
  write.csv(sorted_data[[chr]], file=paste0("inc_chromosome_", chr, ".txt"),
row.names=FALSE)
})
```
~~~

~~~
```{r}
maize_dec=maize
maize_dec[maize_dec=="?/?"]<-"-"
```
~~~

~~~
```{r}
dec_chr<- split(maize_dec, maize_dec$Chromosome)
```
~~~

~~~
```{r,include=FALSE}
sorted_data <- lapply(dec_chr, function(df) df[order(as.numeric(df$Position), decreasing =
TRUE),])
~~~

~~~
lapply(names(sorted_data), function(chr) {write.csv(sorted_data[[chr]], file=paste0("dec_chromosome_", chr, ".txt"), row.names=FALSE, quote=FALSE)
})
```
~~~

Thus we have the required 20 files.

Teosinte Dataset

~~~
```{r,include=FALSE}
grep("ZMPBA",allcols)
grep("ZMPIL",allcols)
grep("ZMPJA",allcols)
```
~~~

~~~
```{r}
teosinte=final[c(1, 2, 3, 77:976, 977:1010, 1166:1206)]
teosinte<-as.data.frame(teosinte)
```
~~~

Let's generate 10 files (1 for each chromosome) with SNPs ordered based on increasing position values and with missing data encoded by this symbol: ?

~~~
```{r,include=FALSE}
teosinte_inc=teosinte
teosinte_inc[teosinte_inc=="?/?"]<-"?"
inc_tchr <- split(teosinte_inc, teosinte_inc$Chromosome)
sorted_data <- lapply(inc_tchr, function(df) {
~~~

~~~
  df[order(as.numeric(df$Position)),]
})
lapply(names(sorted_data), function(chr) {
  write.csv(sorted_data[[chr]], file=paste0("teo_inc_chromosome_", chr, ".txt"), row.names=FALSE)
})
```
~~~

Next we generate 10 files (1 for each chromosome) with SNPs ordered based on decreasing position values and with missing data encoded by this symbol: -

~~~
```{r,include=FALSE}
teosinte_dec=teosinte
teosinte_dec[teosinte_dec=="?/?"]<-"-"
dec_tchr <- split(teosinte_dec, teosinte_inc$Chromosome)
sorted_data <- lapply(dec_tchr, function(df) {
  df[order(as.numeric(df$Position), decreasing = TRUE),]
})
lapply(names(sorted_data), function(chr) {
  write.csv(sorted_data[[chr]], file=paste0("teo_dec_chromosome_", chr, ".txt"), row.names=FALSE)
})
```
~~~

Thus we have all required files.

# Part II Visualization

Step-1: Plotting total number of SNPs per chromosome

~~~
```{r}
library(dplyr)
```

```{r}
maize_snp_count <- aggregate(SNP_ID ~ Chromosome, data = maize, FUN = length)
colnames(maize_snp_count)[2] <- "SNP_Count"
maize_snp_count$Group <- "Maize"
```


```{r}
teosinte_snp_count <- aggregate(SNP_ID ~ Chromosome, data = teosinte, FUN = length)
colnames(teosinte_snp_count)[2] <- "SNP_Count"
teosinte_snp_count$Group <- "Teosinte"
```

```{r}
snp_counts <- rbind(maize_snp_count, teosinte_snp_count)
```
~~~

The chromosomes need to be sorted to be plotted.

~~~
```{r}
snp_counts$Chromosome_Numeric <- as.numeric(snp_counts$Chromosome)
snp_counts$Chromosome_Numeric[snp_counts$Chromosome == "multiple"] <- 11
snp_counts$Chromosome_Numeric[snp_counts$Chromosome == "unknown"] <- 12
~~~

~~~
unique_chromosomes <- unique(snp_counts[, c("Chromosome", "Chromosome_Numeric")])

unique_chromosomes <- unique_chromosomes[order(unique_chromosomes$Chromosome_Numeric), ]

sorted_chromosome_levels <- unique_chromosomes$Chromosome


snp_counts$Chromosome <- factor(snp_counts$Chromosome, levels = sorted_chromosome_levels)


ggplot(snp_counts, aes(x = Chromosome, y = SNP_Count, fill = Group)) +

  geom_bar(stat = "identity", position = "dodge") +

  labs(title = "Distribution of SNPs Across Chromosomes",

       x = "Chromosome",

       y = "Number of SNPs") +

  scale_fill_manual(values = c("Maize" = "#E69F00", "Teosinte" = "#56B4E9")) +

  theme_minimal() +

  theme(legend.position = "top")

```
~~~

Step-2: Identifying homozygous and heterozygous sites

~~~
```{r,include=FALSE}

library(reshape)

library(data.table)

both_long <- filter(fang, Group == "ZMMIL" | Group == "ZMMLR" | Group == "ZMMMR" | Group == "ZMPBA" | Group == "ZMPIL" | Group == "ZMPJA")

both <- melt(as.data.table(both_long), measure.vars = colnames(fang)[4:986])
~~~

~~~
colnames(both)[4:5] <- c("SNP_ID", "Homozygous")
colnames(both)
```
```{r}
both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("A/A", "C/C", "G/G", "T/T"), TRUE, Homozygous))
both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("A/C", "A/G", "A/T", "C/G", "C/T", "G/T"), FALSE, Homozygous))
both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("?/?"), NA, Homozygous))
both <- arrange(both, Sample_ID, Group)
```



```{r}
ggplot(data = both) +
  geom_bar(mapping = aes(x = Group, fill = Homozygous), stat = "count") +
  ggtitle(label = "SNPs by  groups") +
  ylab(label = "Number of SNPs") +
  ggtitle(label = "SNPs across groups") +
  xlab(label = "Group") +
  ylab(label = "Number of SNPs") +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),  # Center the plot title
    axis.text = element_text(size = 11),
    axis.title = element_text(size = 11)
  )
~~~

~~~
```
~~~

~~~
```{r}
ggplot(data = both) +
 geom_bar(mapping = aes(x = Sample_ID, fill = Homozygous), stat = "count") +
 ggtitle(label = "SNPs by Ordered Sample_ID") +
 ylab(label = "Number of SNPs") +
 ggtitle(label = "SNPs across sample") +
 xlab(label = "Sample") +
 ylab(label = "Number of SNPs") +
 theme(
  plot.title = element_text(hjust = 0.5, size = 16),  # Center the plot title
  axis.title = element_text(size = 12)
 )
```
~~~

we can see that the proportion of homozygous sites are higher compared to heterozygous sites.

Step-3: Own Analysis

Reshaping the original data:

~~~
```{r}
fang_long <- pivot_longer(fang,
                          cols = -c(Sample_ID, JG_OTU, Group),
                          names_to = "SNP",
                          values_to = "Genotype")
~~~

~~~
```
~~~

Let us analyse the proportion of homozygous and heterozygous sites in all of the groups

~~~
```{r}
fang_long <- mutate(fang_long,Genotype_Type = case_when( Genotype == "?" ~ "Missing", str_detect(Genotype, "/") & str_sub(Genotype, 1, 1) ==str_sub(Genotype, 3, 3) ~ "Homozygous", str_detect(Genotype, "/") & str_sub(Genotype, 1, 1) != str_sub(Genotype, 3, 3) ~ "Heterozygous",TRUE ~ "Other" ))
```
~~~

~~~
```{r,include=FALSE}
summary_data <- fang_long %>%
 filter(Genotype_Type != "Missing") %>%
 group_by(Group, Genotype_Type) %>%
 summarise(Count = n()) %>%
 mutate(Proportion = Count / sum(Count))
```
```{r}
ggplot(summary_data, aes(x = Group, y = Proportion, fill = Genotype_Type)) +
 geom_bar(stat = "identity", position = "dodge") +
 labs(title = "Proportion of Homozygous vs. Heterozygous Genotypes by Group",
    x = "Group",
    y = "Proportion",
    fill = "Genotype Type") +
 theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, hjust = 1))
```library(tidyverse)
~~~

~~~~
library(tidyr)

```


# Data Inspection


```{r}

fang <- read.delim("C:/Users/risha/Downloads/fang_et_al_genotypes.txt")

fangdim=dim(fang) #return number of rows and columns

fang_info=(file.info('fang_et_al_genotypes.txt'))

```


```{r}

snp <- read.delim("C:/Users/risha/Downloads/snp_position (1).txt")

snpdim=dim(snp) #return number of rows and columns

snp_info=(file.info('snp_position.txt'))

```


# Data Processing


Use the transposed data before joining


```{r}

fang_t <- read.delim("C:/Users/risha/Downloads/transposed_genotypes.txt")


```
~~~~

From the genotype data, we remove the rows containing Sample_ID and JG_OTU, and arrange the table based on the GROUP row as header to facilitate merging and sorting

~~~
```{r}
fang_t <- as.data.frame(fang_t)
new_fang<-fang_t[-c(0,1),]
colnames(new_fang)<-as.character(new_fang[1,])
new_fang<-new_fang[-c(1),]
```
~~~

Joining the genotype data with the SNP data

~~~
```{r}
merged<-merge(snp,new_fang, by.x="SNP_ID",by.y="Group", all=TRUE  )
```
~~~

Removing columns other than SNP_ID, Chromosome and Position

~~~
```{r}
final <-merged[-c(2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,15)]
```
~~~

Maize Dataset

Find columns containing "ZMMIL", "ZMMLR", and "ZMMMR" and remove the rest

```{r, include=FALSE}
allcols<-colnames(final)
grep("ZMMIL",allcols)
grep("ZMMLR",allcols)
grep("ZMMMR",allcols)

```

```{r}
maize<-final[c(1,2,3,1213:2468, 2469:2495, 2496:2785)]
maize<-as.data.frame(maize)
```

We have all maize data now.

```{r}
maize_inc=maize
maize_inc[maize_inc=="?/?"]<-"?"
```

```{r}
inc_chr <- split(maize_inc, maize_inc$Chromosome)
```

sorting each list based on increasing position values

~~~
```{r,include=FALSE}
sorted_data <- lapply(inc_chr, function(df) {
  df[order(as.numeric(df$Position)),]
})

lapply(names(sorted_data), function(chr) {
  write.csv(sorted_data[[chr]], file=paste0("inc_chromosome_", chr, ".txt"),
row.names=FALSE)
})
```


```{r}
maize_dec=maize
maize_dec[maize_dec=="?/?"]<-"-"
```

```{r}
dec_chr<- split(maize_dec, maize_dec$Chromosome)
```

```{r,include=FALSE}
sorted_data <- lapply(dec_chr, function(df) df[order(as.numeric(df$Position), decreasing =
TRUE),])
lapply(names(sorted_data), function(chr) {write.csv(sorted_data[[chr]],
file=paste0("dec_chromosome_", chr, ".txt"), row.names=FALSE, quote=FALSE)
~~~

```
})
```

Thus we have the required 20 files.

Teosinte Dataset

```{r,include=FALSE}
grep("ZMPBA",allcols)
grep("ZMPIL",allcols)
grep("ZMPJA",allcols)
```

```{r}
teosinte=final[c(1, 2, 3, 77:976, 977:1010, 1166:1206)]
teosinte<-as.data.frame(teosinte)
```

Let's generate 10 files (1 for each chromosome) with SNPs ordered based on increasing position values and with missing data encoded by this symbol: ?

```{r,include=FALSE}
teosinte_inc=teosinte
teosinte_inc[teosinte_inc=="?/?"]<-"?"
inc_tchr <- split(teosinte_inc, teosinte_inc$Chromosome)
sorted_data <- lapply(inc_tchr, function(df) {
  df[order(as.numeric(df$Position)),]
```

~~~
})
lapply(names(sorted_data), function(chr) {
  write.csv(sorted_data[[chr]], file=paste0("teo_inc_chromosome_", chr, ".txt"), row.names=FALSE)
})
```
~~~

Next we generate 10 files (1 for each chromosome) with SNPs ordered based on decreasing position values and with missing data encoded by this symbol: -

~~~
```{r,include=FALSE}
teosinte_dec=teosinte
teosinte_dec[teosinte_dec=="?/?"]<-"-"
dec_tchr <- split(teosinte_dec, teosinte_inc$Chromosome)
sorted_data <- lapply(dec_tchr, function(df) {
  df[order(as.numeric(df$Position), decreasing = TRUE),]
})
lapply(names(sorted_data), function(chr) {
  write.csv(sorted_data[[chr]], file=paste0("teo_dec_chromosome_", chr, ".txt"), row.names=FALSE)
})
```
~~~

Thus we have all required files.

# Part II Visualization

Step-1: Plotting total number of SNPs per chromosome

~~~
```{r}
library(dplyr)
```
~~~

~~~
```{r}
maize_snp_count <- aggregate(SNP_ID ~ Chromosome, data = maize, FUN = length)
colnames(maize_snp_count)[2] <- "SNP_Count"
maize_snp_count$Group <- "Maize"
```
~~~

~~~
```{r}
teosinte_snp_count <- aggregate(SNP_ID ~ Chromosome, data = teosinte, FUN = length)
colnames(teosinte_snp_count)[2] <- "SNP_Count"
teosinte_snp_count$Group <- "Teosinte"
```
~~~

~~~
```{r}
snp_counts <- rbind(maize_snp_count, teosinte_snp_count)
```
~~~

The chromosomes need to be sorted to be plotted.

~~~
```{r}
snp_counts$Chromosome_Numeric <- as.numeric(snp_counts$Chromosome)
snp_counts$Chromosome_Numeric[snp_counts$Chromosome == "multiple"] <- 11
snp_counts$Chromosome_Numeric[snp_counts$Chromosome == "unknown"] <- 12
~~~

~~~
unique_chromosomes <- unique(snp_counts[, c("Chromosome",
"Chromosome_Numeric")])
unique_chromosomes <-
unique_chromosomes[order(unique_chromosomes$Chromosome_Numeric), ]
sorted_chromosome_levels <- unique_chromosomes$Chromosome


snp_counts$Chromosome <- factor(snp_counts$Chromosome, levels =
sorted_chromosome_levels)


ggplot(snp_counts, aes(x = Chromosome, y = SNP_Count, fill = Group)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Distribution of SNPs Across Chromosomes",
      x = "Chromosome",
      y = "Number of SNPs") +
  scale_fill_manual(values = c("Maize" = "#E69F00", "Teosinte" = "#56B4E9")) +
  theme_minimal() +
  theme(legend.position = "top")
```
~~~

Step-2: Identifying homozygous and heterozygous sites

~~~
```{r,include=FALSE}
library(reshape)
library(data.table)
both_long <- filter(fang, Group == "ZMMIL" | Group == "ZMMLR" | Group == "ZMMMR" |
Group == "ZMPBA" | Group == "ZMPIL" | Group == "ZMPJA")
both <- melt(as.data.table(both_long), measure.vars = colnames(fang)[4:986])
~~~

~~~
colnames(both)[4:5] <- c("SNP_ID", "Homozygous")
colnames(both)
```
```{r}
both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("A/A", "C/C", "G/G", "T/T"), TRUE, Homozygous))
both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("A/C", "A/G", "A/T", "C/G", "C/T", "G/T"), FALSE, Homozygous))
both <- mutate(both, Homozygous = ifelse(Homozygous %in% c("?/?"), NA, Homozygous))
both <- arrange(both, Sample_ID, Group)
```



```{r}
ggplot(data = both) +
 geom_bar(mapping = aes(x = Group, fill = Homozygous), stat = "count") +
 ggtitle(label = "SNPs by  groups") +
 ylab(label = "Number of SNPs") +
 ggtitle(label = "SNPs across groups") +
 xlab(label = "Group") +
 ylab(label = "Number of SNPs") +
 theme(
  plot.title = element_text(hjust = 0.5, size = 16),  # Center the plot title
  axis.text = element_text(size = 11),
  axis.title = element_text(size = 11)
 )
~~~

~~~
```

```{r}
ggplot(data = both) +
  geom_bar(mapping = aes(x = Sample_ID, fill = Homozygous), stat = "count") +
  ggtitle(label = "SNPs by Ordered Sample_ID") +
  ylab(label = "Number of SNPs") +
  ggtitle(label = "SNPs across sample") +
  xlab(label = "Sample") +
  ylab(label = "Number of SNPs") +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),  # Center the plot title
    axis.title = element_text(size = 12)
  )
```
~~~

we can see that the proportion of homozygous sites are higher compared to heterozygous sites.

Step-3: Own Analysis

Reshaping the original data:

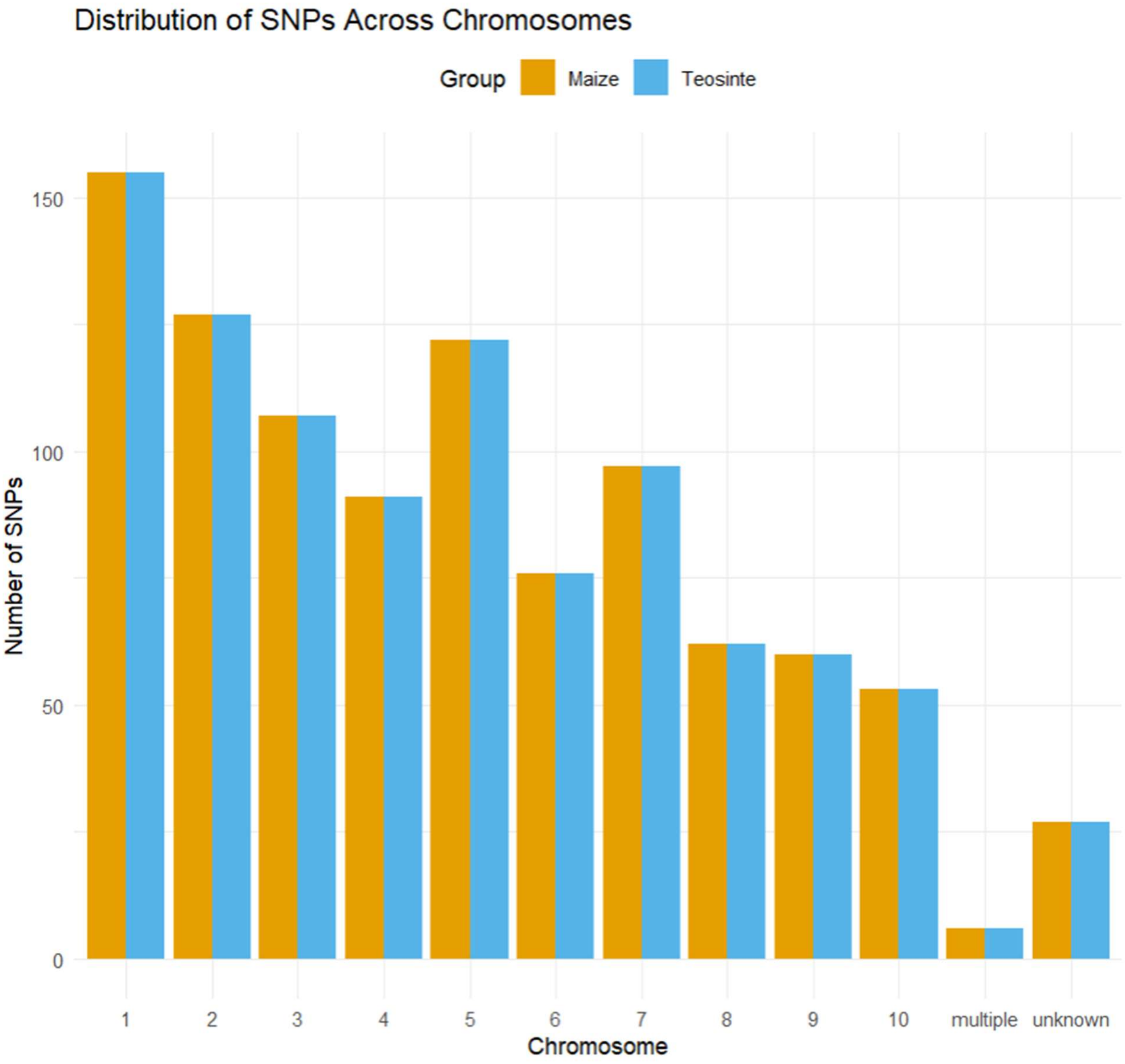
~~~
```{r}
fang_long <- pivot_longer(fang,
                          cols = -c(Sample_ID, JG_OTU, Group),
                          names_to = "SNP",
                          values_to = "Genotype")
~~~

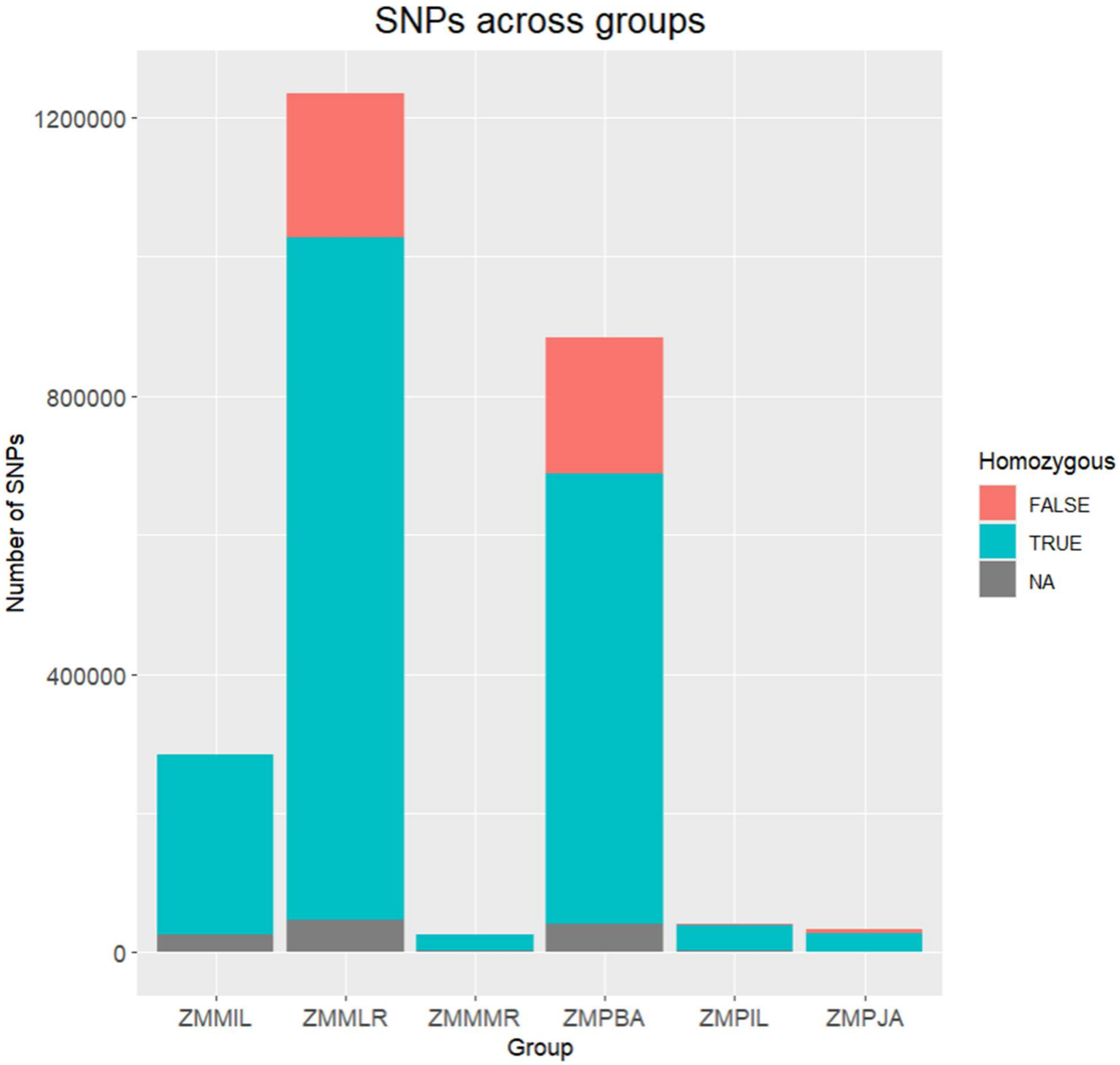
Let us analyse the proportion of homozygous and heterozygous sites in all of the groups

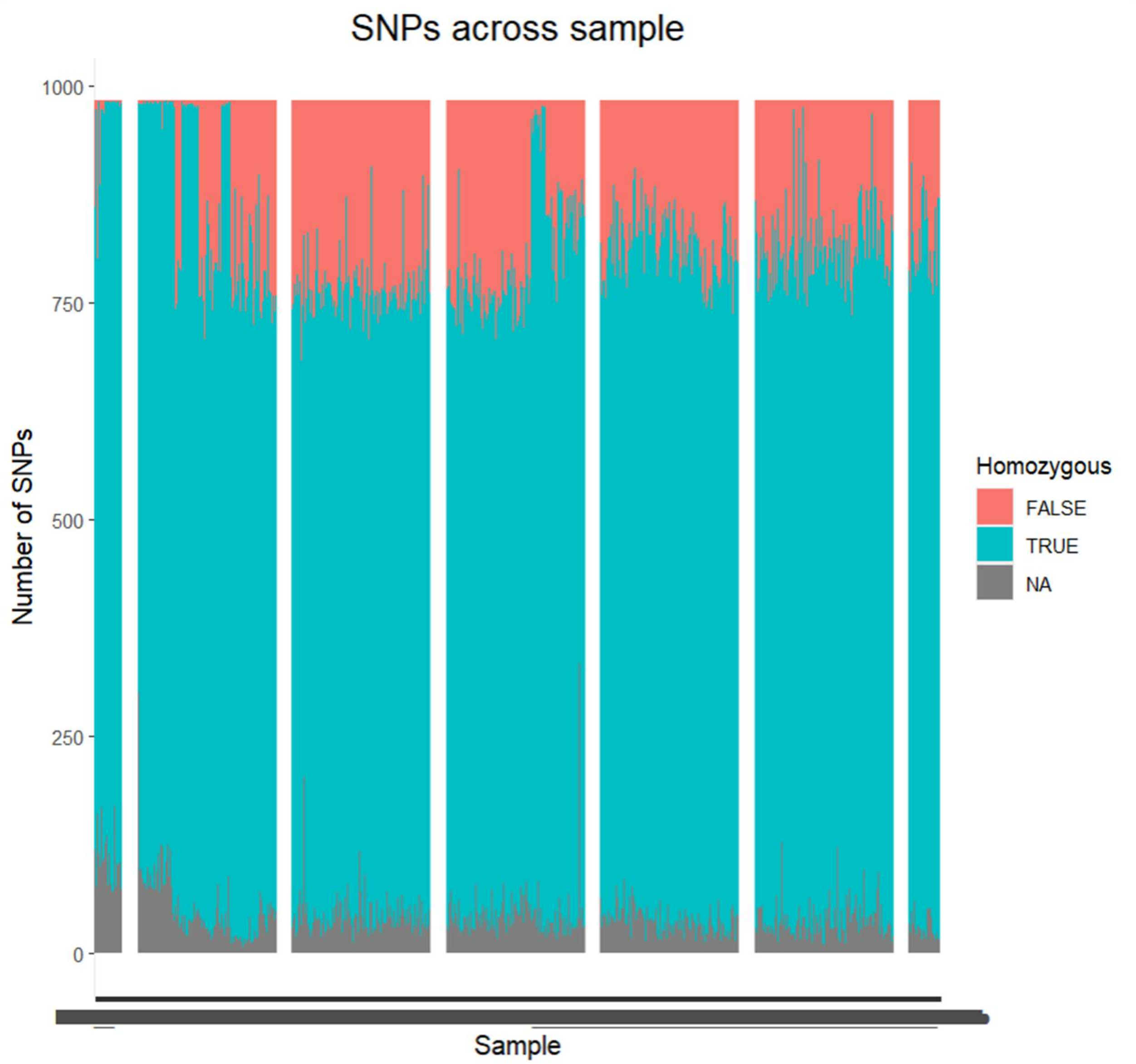
```{r}
fang_long <- mutate(fang_long,Genotype_Type = case_when( Genotype == "?" ~ "Missing", str_detect(Genotype, "/") & str_sub(Genotype, 1, 1) ==str_sub(Genotype, 3, 3) ~ "Homozygous", str_detect(Genotype, "/") & str_sub(Genotype, 1, 1) != str_sub(Genotype, 3, 3) ~ "Heterozygous",TRUE ~ "Other" ))
```

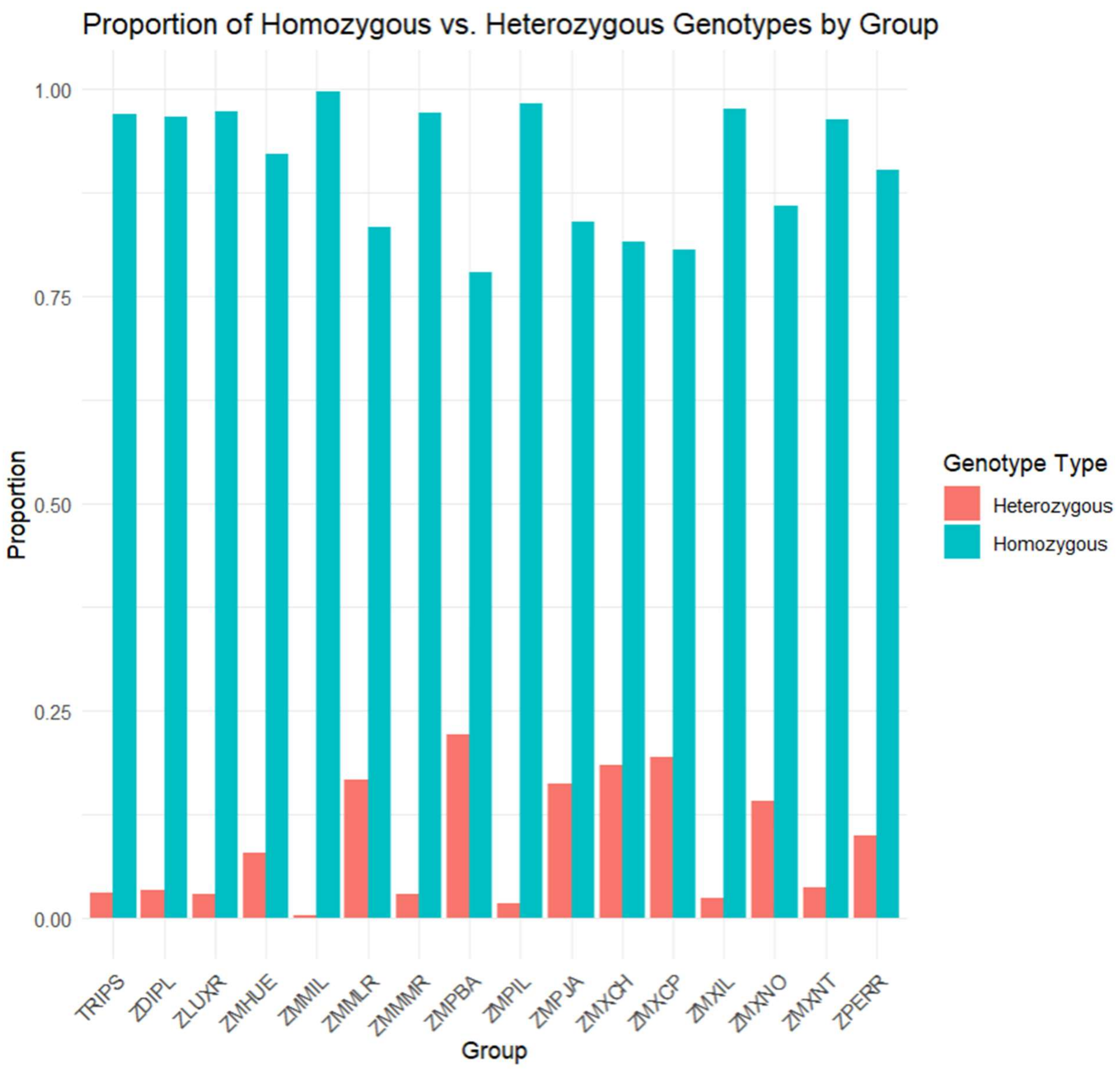
```{r,include=FALSE}
summary_data <- fang_long %>%
 filter(Genotype_Type != "Missing") %>%
 group_by(Group, Genotype_Type) %>%
 summarise(Count = n()) %>%
 mutate(Proportion = Count / sum(Count))
```

```{r}
ggplot(summary_data, aes(x = Group, y = Proportion, fill = Genotype_Type)) +
 geom_bar(stat = "identity", position = "dodge") +
 labs(title = "Proportion of Homozygous vs. Heterozygous Genotypes by Group",
    x = "Group",
    y = "Proportion",
    fill = "Genotype Type") +
 theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Distribution of SNPs Across Chromosomes
Group
Maize
Teosinte
Number of SNPs
150
100
50
0
1
2
3
4
5
6
7
8
9
10
multiple
unknown
Chromosome


SNPs across groups
Number of SNPs
1200000
800000
400000
0
ZMMIL
ZMMLR
ZMMMR
ZMPBA
ZMPIL
ZMPJA
Group
Homozygous
FALSE
TRUE
NA


SNPs across sample
Number of SNPs
1000
750
500
250
0
Sample
Homozygous
FALSE
TRUE
NA


Proportion of Homozygous vs. Heterozygous Genotypes by Group
Proportion
1.00
0.75
0.50
0.25
0.00
TRIPS
ZDIPL
ZLUXR
ZMHUE
ZMMIL
ZMMLR
ZMMMR
ZMPBA
ZMPIL
ZMPJA
ZMXCH
ZMXCP
ZMXIL
ZMXNO
ZMXNT
ZPERR
Group
Genotype Type
Heterozygous
Homozygous