bcb546x_r_assignments

Zihao Zheng

10/9/2018

Part I

1. Data inspection

```
setwd("~/Documents/BCB546X-Fall2018/assignments/UNIX_Assignment/")
library(tidyverse)
library(reshape2)
library(ggrepel)
```

Check the size of files to be loaded

```
file.size("fang_et_al_genotypes.txt") %>% utils:::format.object_size("auto")
```

```
## [1] "10.5 Mb"
```

```
file.size("snp_position.txt") %>% utils:::format.object_size("auto")
```

```
## [1] "80.8 Kb"
```

load files to R

```
fang_et_al <- read_delim("fang_et_al_genotypes.txt",delim = "\t",col_names = T)
snp_position <- read_delim("snp_position.txt",delim = "\t",col_names = T)
```

number of columns and rows for each file

```
# fang_et_al column#
ncol(fang_et_al)
```

```
## [1] 986
```

```
# fang_et_al row#
nrow(fang_et_al)
```

```
## [1] 2782
```

```
head(fang_et_al)
```

```
## # A tibble: 6 x 986
##   Sample_ID JG_OTU   Group abph1.20 abph1.22 ae1.3 ae1.4 ae1.5 an1.4 ba1.6
##   <chr>     <chr>    <chr> <chr>    <chr>    <chr> <chr> <chr> <chr> <chr>
## 1 SL-15     T-aust-1 TRIPS ?/?      ?/?      T/T   G/G   T/T   C/C   ?/?
## 2 SL-16     T-aust-2 TRIPS ?/?      ?/?      T/T   ?/?   T/T   C/C   A/G
## 3 SL-11     T-brav-1 TRIPS ?/?      ?/?      T/T   G/G   T/T   ?/?   G/G
## 4 SL-12     T-brav-2 TRIPS ?/?      ?/?      T/T   G/G   T/T   C/C   G/G
## 5 SL-18     T-cund   TRIPS ?/?      ?/?      T/T   G/G   T/T   C/C   ?/?
## 6 SL-2      T-dact-1 TRIPS ?/?      ?/?      T/T   G/G   T/T   C/C   A/G
## # ... with 976 more variables: ba1.9 <chr>, bt2.5 <chr>, bt2.7 <chr>,
## #   bt2.8 <chr>, Fea2.1 <chr>, Fea2.5 <chr>, id1.3 <chr>, lg2.11 <chr>,
## #   lg2.2 <chr>, pbf1.1 <chr>, pbf1.2 <chr>, pbf1.3 <chr>, pbf1.5 <chr>,
## #   pbf1.6 <chr>, pbf1.7 <chr>, pbf1.8 <chr>, PZA00003.11 <chr>,
```

```
## #   PZA00004.2 <chr>, PZA00005.8 <chr>, PZA00005.9 <chr>,
## #   PZA00006.13 <chr>, PZA00006.14 <chr>, PZA00008.1 <chr>,
## #   PZA00010.5 <chr>, PZA00013.10 <chr>, PZA00013.11 <chr>,
## #   PZA00013.9 <chr>, PZA00015.4 <chr>, PZA00017.1 <chr>,
## #   PZA00018.5 <chr>, PZA00029.11 <chr>, PZA00029.12 <chr>,
## #   PZA00030.11 <chr>, PZA00031.5 <chr>, PZA00041.3 <chr>,
## #   PZA00042.2 <chr>, PZA00042.5 <chr>, PZA00043.7 <chr>,
## #   PZA00045.1 <chr>, PZA00047.2 <chr>, PZA00049.12 <chr>,
## #   PZA00050.9 <chr>, PZA00051.2 <chr>, PZA00058.5 <chr>,
## #   PZA00058.6 <chr>, PZA00060.2 <chr>, PZA00061.1 <chr>,
## #   PZA00065.2 <chr>, PZA00069.4 <chr>, PZA00070.5 <chr>,
## #   PZA00078.2 <chr>, PZA00079.1 <chr>, PZA00081.17 <chr>,
## #   PZA00084.2 <chr>, PZA00084.3 <chr>, PZA00086.8 <chr>,
## #   PZA00088.3 <chr>, PZA00090.2 <chr>, PZA00092.1 <chr>,
## #   PZA00092.5 <chr>, PZA00093.2 <chr>, PZA00096.26 <chr>,
## #   PZA00097.13 <chr>, PZA00098.14 <chr>, PZA00100.10 <chr>,
## #   PZA00100.12 <chr>, PZA00100.14 <chr>, PZA00100.9 <chr>,
## #   PZA00103.20 <chr>, PZA00106.9 <chr>, PZA00107.18 <chr>,
## #   PZA00108.12 <chr>, PZA00108.14 <chr>, PZA00108.15 <chr>,
## #   PZA00109.3 <chr>, PZA00109.5 <chr>, PZA00111.2 <chr>,
## #   PZA00111.4 <chr>, PZA00111.5 <chr>, PZA00111.6 <chr>,
## #   PZA00111.8 <chr>, PZA00114.3 <chr>, PZA00116.2 <chr>,
## #   PZA00119.4 <chr>, PZA00120.4 <chr>, PZA00123.1 <chr>,
## #   PZA00125.2 <chr>, PZA00131.14 <chr>, PZA00132.17 <chr>,
## #   PZA00132.18 <chr>, PZA00132.3 <chr>, PZA00135.6 <chr>,
## #   PZA00137.2 <chr>, PZA00139.14 <chr>, PZA00140.10 <chr>,
## #   PZA00140.6 <chr>, PZA00140.9 <chr>, PZA00142.6 <chr>,
## #   PZA00148.2 <chr>, PZA00153.3 <chr>, ...
```

```
# snp_position column#
ncol(snp_position)
```

```
## [1] 15
```

```
# snp_position row#
nrow(snp_position)
```

```
## [1] 983
```

```
head(snp_position)
```

```
## # A tibble: 6 x 15
##   SNP_ID cdv_marker_id Chromosome Position alt_pos mult_positions amplicon
##   <chr>          <int> <chr>      <chr>    <chr>   <chr>          <chr>
## 1 abph1~          5976 2          27403404 <NA>    <NA>           abph1
## 2 abph1~          5978 2          27403892 <NA>    <NA>           abph1
## 3 ae1.3           6605 5          1678897~ <NA>    <NA>           ae1
## 4 ae1.4           6606 5          1678896~ <NA>    <NA>           ae1
## 5 ae1.5           6607 5          1678898~ <NA>    <NA>           ae1
## 6 an1.4           5982 1          2404985~ <NA>    <NA>           an1
## # ... with 8 more variables: cdv_map_feature.name <chr>, gene <chr>,
## #   `candidate/random` <chr>, Genaissance_daa_id <int>,
## #   Sequenom_daa_id <int>, count_amplicons <int>, count_cmf <int>,
## #   count_gene <int>
```

2. Data processing

split `fang_et_al` into maize group and teosinte group

```
maize.snp <- fang_et_al %>% filter(Group %in% c("ZMMIL","ZMMLR","ZMMMR"))
teosinte.snp <- fang_et_al %>% filter(Group %in% c("ZMPBA","ZMPIL","ZMPJA"))
```

transpose genotypic data, and merge with the `snp_position`

```
# maize group

maize.tmp <- data.frame(t(maize.snp)) %>% tibble::rownames_to_column()
colnames(maize.tmp) <- maize.tmp[1, ]
maize.tmp <- maize.tmp[-(1:3),]
colnames(maize.tmp)[1] <- "SNP_ID"
merged.maize.genotype <- merge(snp_position,maize.tmp,by = "SNP_ID")

# teosinte group
teosinte.tmp <- data.frame(t(teosinte.snp)) %>% tibble::rownames_to_column()
colnames(teosinte.tmp) <- teosinte.tmp[1, ]
teosinte.tmp <- teosinte.tmp[-(1:3),]
colnames(teosinte.tmp)[1] <- "SNP_ID"
merged.teosinte.genotype <- merge(snp_position,teosinte.tmp,by = "SNP_ID")
```

For maize group, generate 10 files (1 for each chromosome) with SNPs ordered based on increasing position values and with missing data encoded by this symbol: ?

```
new_names <- c("maize_chr1","maize_chr10","maize_chr2","maize_chr3",
               "maize_chr4","maize_chr5","maize_chr6","maize_chr7",
               "maize_chr8","maize_chr9","maize_multiple","maize_unknown")

maize_split_incr <- split(merged.maize.genotype, merged.maize.genotype$Chromosome)

for (i in 1:10) {
  maize_split_incr[[i]]<-
    maize_split_incr[[i]][order(as.numeric(maize_split_incr[[i]]$Position)),]
  write_delim(maize_split_incr[[i]],paste0(new_names[i],"_incr.txt"),delim = "\t")
 }
```

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

```
merged.maize.genotype[merged.maize.genotype== "?/?"]<- "-/-"

maize_split_decr <- split(merged.maize.genotype, merged.maize.genotype$Chromosome)

for (i in 1:10) {
  maize_split_decr[[i]]<-
    maize_split_decr[[i]][order(as.numeric(maize_split_decr[[i]]$Position)),]
  write_delim(maize_split_decr[[i]],paste0(new_names[i],"_decr.txt"),delim = "\t")
 }
```

For teosinte group, generate 10 files (1 for each chromosome) with SNPs ordered based on increasing position values and with missing data encoded by this symbol: ?

```
new_names <- c("teosinte_chr1","teosinte_chr10","teosinte_chr2","teosinte_chr3",
               "teosinte_chr4","teosinte_chr5","teosinte_chr6","teosinte_chr7",
               "teosinte_chr8","teosinte_chr9","teosinte_multiple","teosinte_unknown")
```

```
teosinte_split_incr <- split(merged.teosinte.genotype, merged.teosinte.genotype$Chromosome)

for (i in 1:10) {
  teosinte_split_incr[[i]]<-
    teosinte_split_incr[[i]][order(as.numeric(teosinte_split_incr[[i]]$Position)),]
  write_delim(teosinte_split_incr[[i]],paste0(new_names[i],"_incr.txt"),delim = "\t")
 }
```

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

```
merged.teosinte.genotype[merged.teosinte.genotype== "?/?"]<- "-/-"

teosinte_split_decr <- split(merged.teosinte.genotype, merged.teosinte.genotype$Chromosome)

for (i in 1:10) {
  teosinte_split_decr[[i]]<-
    teosinte_split_decr[[i]][order(as.numeric(teosinte_split_decr[[i]]$Position)),]
  write_delim(teosinte_split_decr[[i]],paste0(new_names[i],"_decr.txt"),delim = "\t")
 }
```

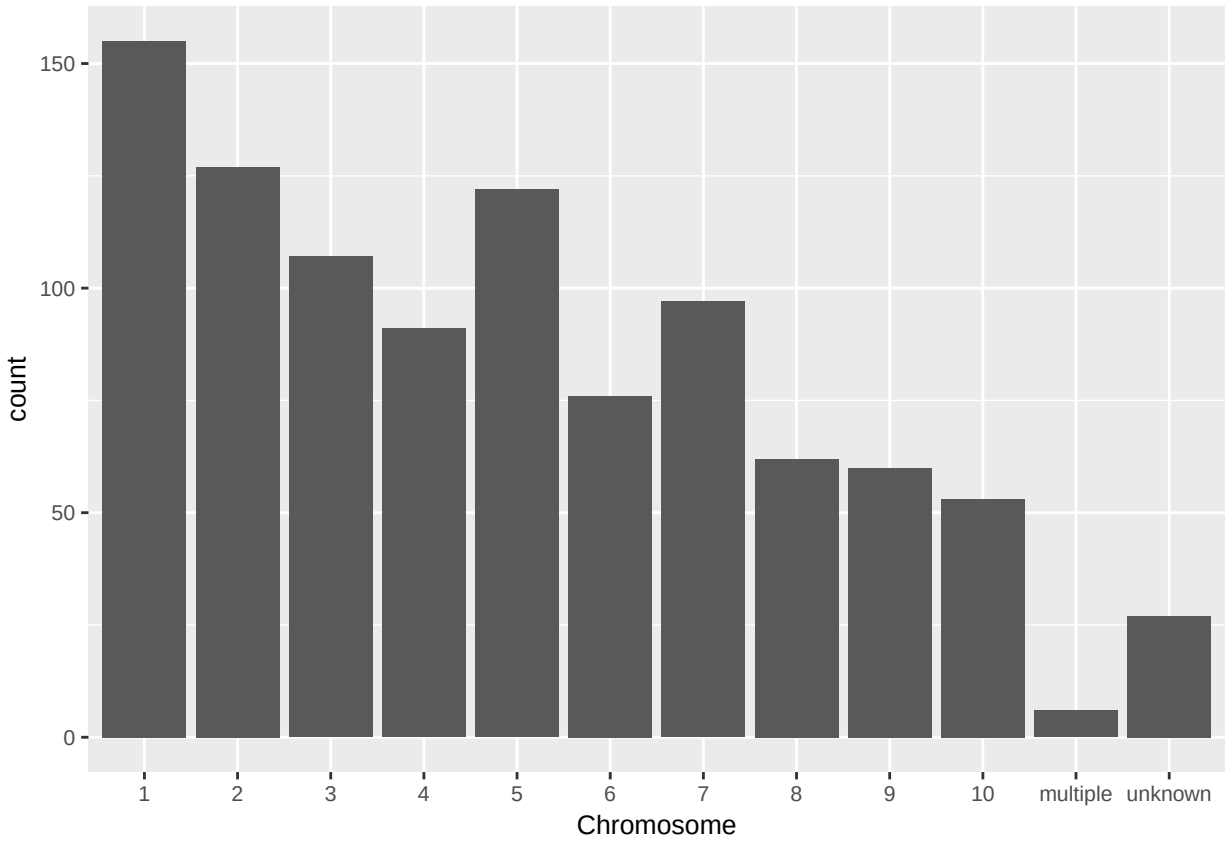
Part II

Merge `fang_et_al` and `snp_position` to get a master file

```
genotype.tmp <- data.frame(t(fang_et_al)) %>% tibble::rownames_to_column()
colnames(genotype.tmp) <- genotype.tmp[1, ]
genotype.tmp <- genotype.tmp[-(1:3),]
colnames(genotype.tmp)[1] <- "SNP_ID"
merged.genotype <- merge(snp_position,genotype.tmp,by = "SNP_ID")
```

1(a). Plot the total number of SNPs in our dataset on each chromosome.

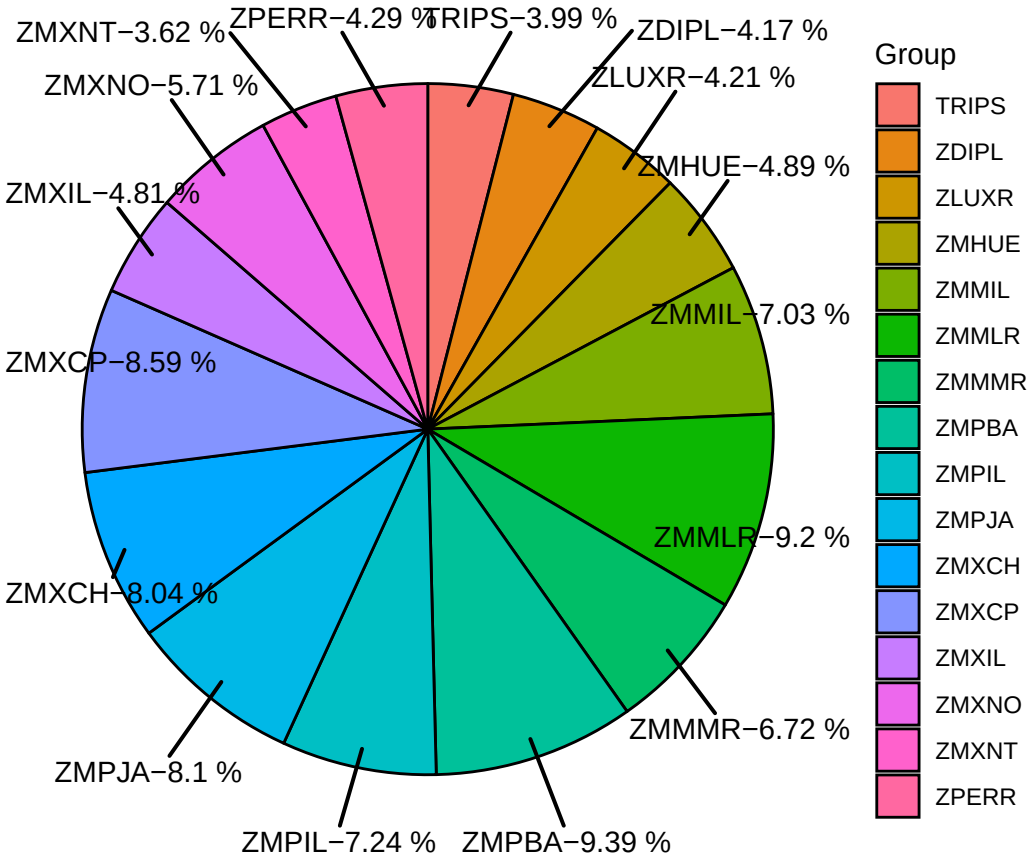
```
p1 <- ggplot(data = merged.genotype) + geom_bar(mapping = aes(x=Chromosome)) +
      scale_x_discrete(limits=c(1:10,"multiple","unknown")) +
      theme(text = element_text(size=10))
p1
```



1(b).What groups contribute most of these SNPs?

```
fang_et_al_short <- fang_et_al %>% select(-c(Sample_ID,JG_OTU)) %>%
  melt(id.vars = "Group",variable.name = "SNP_ID") %>% unique()

snp_position.tmp <- snp_position %>% select(SNP_ID,Chromosome) %>%
  merge(fang_et_al_short,by = "SNP_ID")
snp_position.tmp[snp_position.tmp=="?/?"] <- NA
snp_position.tmp <- na.omit(snp_position.tmp)

snp_group_stat <- snp_position.tmp %>% count(Group)
snp_group_stat$Label <- paste(snp_group_stat$Group,
                        paste(round(((snp_group_stat$n/sum(snp_group_stat$n))*100),2),"%"), sep="-")

snp_group_stat$pos = (cumsum(c(0, snp_group_stat$n))
                      + c(snp_group_stat$n / 2, .01))[1:nrow(snp_group_stat)]


p2 <- ggplot(snp_group_stat, aes(1, n, fill = Group)) +
    geom_col(color = 'black',
             position = position_stack(reverse = TRUE),
             show.legend = TRUE) +
    geom_text_repel(aes(x = 1.4, y = pos, label = Label),
                    nudge_x = .3,
                    segment.size = .7,
```

```
                    show.legend = FALSE) +
    coord_polar('y') +
    theme_void()
p2
```



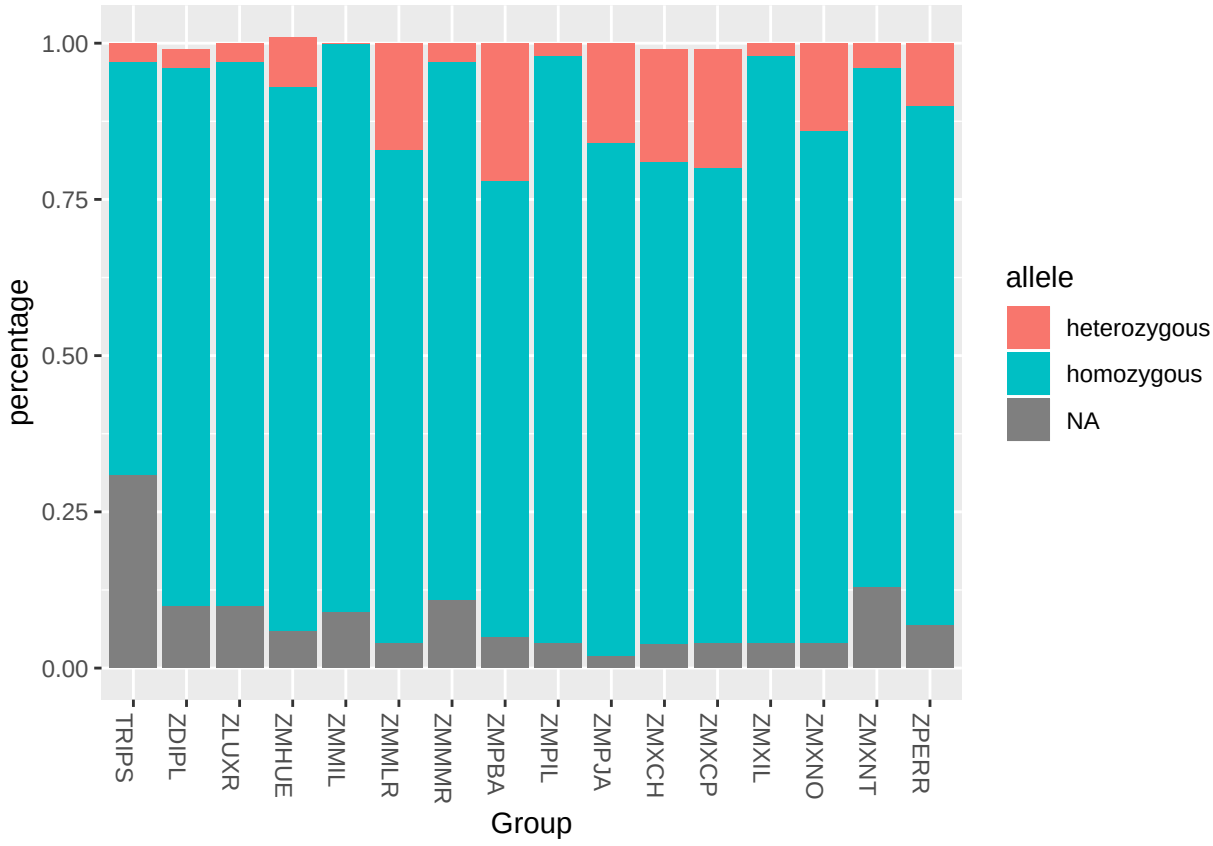
2. Missing data and amount of heterozygosity

```
fang_et_al_short2 <- fang_et_al %>% select(-one_of("JG_OTU")) %>%
                     melt(id.vars = c("Group","Sample_ID"),variable.name = "SNP_ID")
fang_et_al_short2[fang_et_al_short2=="?/?"] <- NA

fang_et_al_short2$allele[is.na(fang_et_al_short2$value)] <- NA
fang_et_al_short2$allele[fang_et_al_short2$value %in% c("A/A","C/C","G/G","T/T")] <- "homozygous"
fang_et_al_short2$allele[!fang_et_al_short2$value %in% c("A/A","C/C","G/G","T/T",NA)] <- "heterozygous"


allele_stat_group <- fang_et_al_short2 %>% group_by(Group,allele) %>%
  summarise(n = n()) %>% mutate(countT= sum(n)) %>%
  mutate(percentage=round(n/countT,2))

p3 <- ggplot(allele_stat_group) +
      geom_bar(aes(y = percentage, x = Group, fill = allele),stat="identity") +
      theme(axis.text.x=element_text(angle = -90, hjust = 0))
```

```
allele_stat_sample <- fang_et_al_short2 %>% group_by(Sample_ID,allele) %>%
  summarise(n = n()) %>% mutate(countT= sum(n)) %>%
  mutate(percentage=round(n/countT,2))

p4 <- ggplot(allele_stat_sample) +
  geom_bar(aes(y = percentage, x = Sample_ID, fill = allele), stat="identity") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

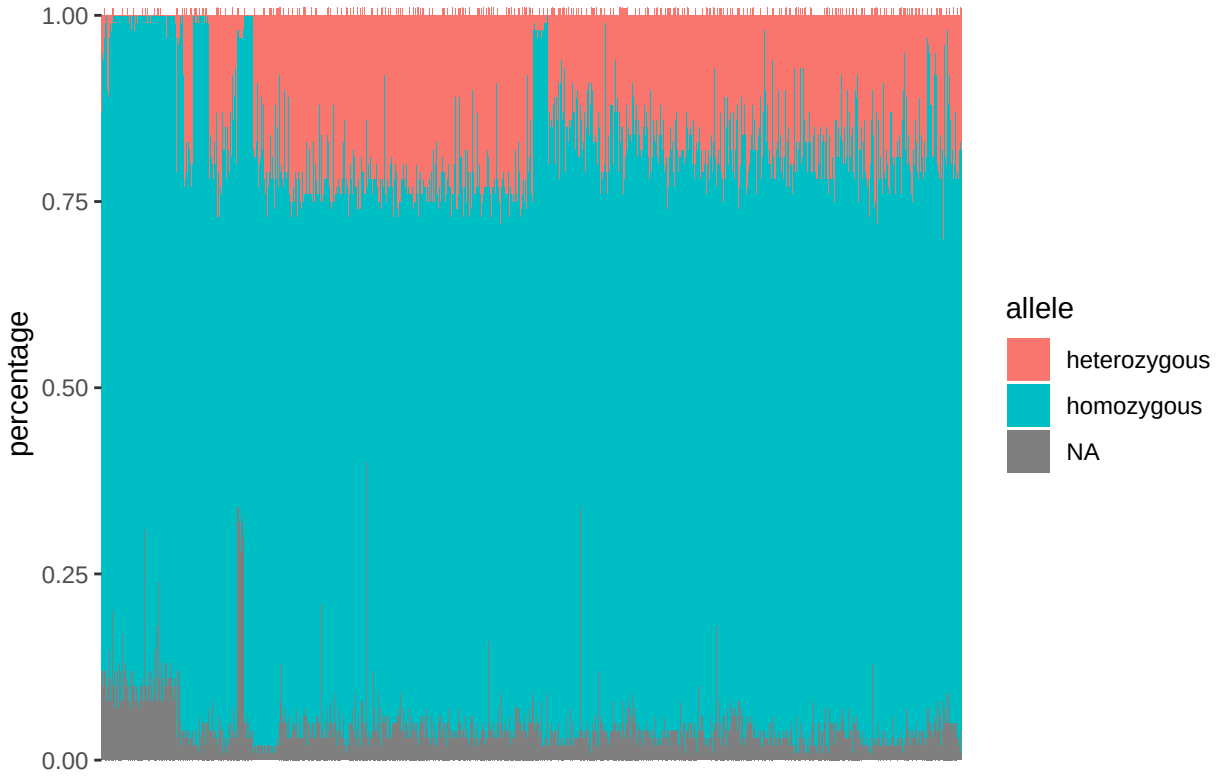
2(a).Missing data and amount of heterozygosity by group

```
p3
```



2(b).Missing data and amount of heterozygosity by individual

```
p4
```



3.Visualization of Minor Allele Frequency (MAF) of maize group and teosinte group

Calculate MAF for each group

```
MAF.tmp <- fang_et_al_short2 %>% filter(allele=="homozygous") %>%
  group_by(Group,SNP_ID,value) %>%
  summarise(n = n()) %>% mutate(countT= sum(n)) %>%
  mutate(allele_freq=round(n/countT,3)) %>%
  filter(allele_freq < 0.5) %>%
  select(Group,SNP_ID,allele_freq) %>%
  group_by(Group,SNP_ID) %>% top_n(-1)

colnames(MAF.tmp)[3] <- "MAF"

MAF.tmp$species <- NA
MAF.tmp$species[MAF.tmp$Group %in% c("ZMMIL","ZMMLR","ZMMMR")] <- "maize"
MAF.tmp$species[MAF.tmp$Group %in% c("ZMPBA","ZMPIL","ZMPJA")] <- "teosinte"
MAF.tmp <- na.omit(MAF.tmp)

p5<- ggplot(MAF.tmp, aes(x=MAF, fill=Group)) +
    geom_histogram(binwidth=0.01, alpha=.5, position="identity") +
    facet_grid(species ~ .)
p5
```


count
100
75
50
25
0
maize
teosinte
0.0
0.1
0.2
0.3
0.4
0.5
MAF
Group
ZMMIL
ZMMLR
ZMMMR
ZMPBA
ZMPIL
ZMPJA