Chiang_review2

2025-03-21

Peer Review of `Sristi.Rmd` by Lillian Chiang

Hi Sristi! Overall, I think you did a great job on this assignment. For this review, I will discuss each section of the assignment (data inspection, data processing, and data visualization) and provide feedback.

Data Inspection

Both data sets (`fang` & `snp`) were described well with the code used. I believe all the information you gathered (data set dimensions, column names, variable types, etc.) are important for your data analysis and would be sufficient to complete the assignment goals.

One suggestion I have for this section, and the following sections as well, is to split your code into separate code chunks. By formatting your document in smaller sections, you can create headers, sub headers, and captions that will enhance your analysis and the readability of your output Rmd/pdf file. This will also make running and troubleshooting your code much quicker since you wouldn't have to either run all your code repeatedly or find the specific code and highlight the text to run.

I also noticed that there is no discussion of the data output in this section, which may be helpful to include either as a reminder to yourself or for any reader viewing your code without a solid background of these type of data sets or commands. One example of this could be

```
dim(genotypes) # 2782 observations,  986 variables
dim(snp_position) # 983 observations, 15 variables

# having this info easily visible would be nice, especially when ensuring filtering or other commands w
```

Data Processing

Overall, your code ran after some alterations and output the correct files. The format you currently have does not run all at once and requires the reader to write commands to upload the files to complete the rest of your analysis.

```
#Extract chromosome 1-10 data from merged_data_subset file
write.table(rbind(colnames(merged_data_subset), merged_data_subset[merged_data_subset[, 2] == 1, ]), fil
write.table(rbind(colnames(merged_data_subset), merged_data_subset[merged_data_subset[, 2] == 2, ]), fil
write.table(rbind(colnames(merged_data_subset), merged_data_subset[merged_data_subset[, 2] == 3, ]), fil
write.table(rbind(colnames(merged_data_subset), merged_data_subset[merged_data_subset[, 2] == 4, ]), fil
write.table(rbind(colnames(merged_data_subset), merged_data_subset[merged_data_subset[, 2] == 5, ]), fil
write.table(rbind(colnames(merged_data_subset), merged_data_subset[merged_data_subset[, 2] == 6, ]), fil
write.table(rbind(colnames(merged_data_subset), merged_data_subset[merged_data_subset[, 2] == 7, ]), fil
write.table(rbind(colnames(merged_data_subset), merged_data_subset[merged_data_subset[, 2] == 8, ]), fil
write.table(rbind(colnames(merged_data_subset), merged_data_subset[merged_data_subset[, 2] == 9, ]), fil
write.table(rbind(colnames(merged_data_subset), merged_data_subset[merged_data_subset[, 2] == 10, ]), fi

# might be missing this section as the following sections already have "maize_chr1" read in (I needed t
maize_chr1 <- read.table("maize_chr1.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)
maize_chr2 <- read.table("maize_chr2.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)
maize_chr3 <- read.table("maize_chr3.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)
```

```
maize_chr4 <- read.table("maize_chr4.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)
maize_chr5 <- read.table("maize_chr5.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)
maize_chr6 <- read.table("maize_chr6.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)
maize_chr7 <- read.table("maize_chr7.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)
maize_chr8 <- read.table("maize_chr8.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)
maize_chr9 <- read.table("maize_chr9.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)
maize_chr10 <- read.table("maize_chr10.txt", header = TRUE, sep = "\t", stringsAsFactors = FALSE)
```

The biggest standout in this section is the repeating nature of your code. For example, the data frame creation for each chromosome extraction is a line of code; however, you could create a function or command loop that could quickly accomplish this code in much fewer lines. Furthermore, you could combine the data processing steps (filtering per group, per chromosome, converting data types, sorting positions, and creating and saving output files) into a function or command loop, resulting in fewer lines of code to review and alter. I have provided the function I used for my analysis if you would like to see an example:

```
# ~~~ FUNCTION CREATION ~~~

process_chr_data <- function(data, chr_num, output_prefix, replace_na, sort_order = "asc", output_dir =
  # create file output directory (unless it already exists)
  if(!dir.exists(output_dir)) {
    dir.create(output_dir, recursive = FALSE)
  }

  # filter data
  filtered_data <- data |>
    filter(Chromosome == chr_num, !grepl("unknown|multiple", Chromosome)) |>
    mutate(
      across(everything(), ~ gsub("\\?/\\?", replace_na, .)),
      Position = as.numeric(as.character(Position))
      ) |>
    arrange(if (sort_order == "asc") Position else desc(Position))

  # write to file
  output_file <- file.path(output_dir, sprintf("%s_chr%d_%s.txt", output_prefix, chr_num, sort_order))
  write_tsv(filtered_data, output_file)
}

# ~~~ maize ascending ~~~
lapply(1:10, function(chr) { # chromosomes 1 - 10
  process_chr_data(maize_join, chr, "maize", "?/?", "asc", "maize_data")
})

# ~~~ maize descending ~~~
lapply(1:10, function(chr) { # chromosomes 1 - 10
  process_chr_data(maize_join, chr, "maize", "-/-", "desc", "maize_data")
})
```

I think the inclusion of viewing the `maize_chr7_sorted` data frame within your code was a beneficial addition. This allowed the reader to see what type of format the output file would be in.

Data Visualization

Your figures are neat and display the information outlined in the assignment instructions. I have a couple suggestions for figure aesthetics that I have listed below:

- All SNP bar charts

1. Chromosomes on x axis are not in order (10 is before 2, consider converting chromosome to factor and manually ordering)
2. remove legend; the x-axis label clearly labels the chromosomes, the colors are all unique so no need for color legend, and the legend is out of order (Chr 10 before 2)
3. angle x axis labels so "multiple" and "unknown" don't overlap

```
## 1)
# ordering chromosomes, example code
chromosome_order <- c(as.character(1:10), "multiple", "unknown")
df <- df |>
  mutate(Chromosome = factor(Chromosome, levels = chromosome_order))

## 2 & 3)
# your code
ggplot(snp_comparison, aes(x = Chromosome, y = SNP_Count, fill = Species)) +
geom_bar(stat = "identity", position = "dodge") +
labs(title = "SNPs per Chromosome in Maize vs Teosinte", x = "Chromosome", y = "Number of SNPs") +
theme_minimal()

# addition (#2)
+ theme(legend.position = "none")

# addition (#3)
+ theme(axis.text.x = element_text(angle = 45))
```

I also think it would be nice to include a brief description or caption describing what can be gathered from the figure (e.g. which chromosome has the most SNPs, which has the least, etc.)

Overall, I think you completed an organized analysis. The only suggestion I think might be necessary is to include the read-in code for the chr1-10.txt files, but other than that feel free to take any of my other suggestions into account, but even without them, I think your analysis is excellent!