

Assessment 4

Nicole Hinojosa, Sakeena Thapa, Vanshika

2024-10-03

Part 1: Importing files, data wrangling, mathematical operations, plots and saving code on GitHub.

Introduction

This report analyzes RNA-seq count data for gene expression and tree circumference measurements at two different sites over a 20-year period.

Task 1: RNA-seq count data for gene expression, high and low expression of 3 genes.

1.1 Read in the file “gene_expression.tsv”, making the gene identifiers the row names. Show a table of values for the first six genes.

1) Load libraries

```
#RNA-seq Count Data Analysis  
#Load necessary libraries  
library(R.utils)
```

```
## Loading required package: R.oo  
## Loading required package: R.methodsS3  
## R.methodsS3 v1.8.2 (2022-06-13 22:00:14 UTC) successfully loaded. See ?R.methodsS3 for help.  
## R.oo v1.26.0 (2024-01-24 05:12:50 UTC) successfully loaded. See ?R.oo for help.  
##  
## Attaching package: 'R.oo'  
## The following object is masked from 'package:R.methodsS3':  
##  
##      throw  
## The following objects are masked from 'package:methods':  
##  
##      getClasses, getMethods  
## The following objects are masked from 'package:base':  
##  
##      attach, detach, load, save  
## R.utils v2.12.3 (2023-11-18 01:00:02 UTC) successfully loaded. See ?R.utils for help.  
##  
## Attaching package: 'R.utils'
```

```

## The following object is masked from 'package:utils':
##
##     timestamp
## The following objects are masked from 'package:base':
##
##     cat, commandArgs, getOption, isOpen, nullfile, parse, warnings
#use BiocManager::install("Biostrings") if it is not already installed in your Rstudio
library(Biostrings)

## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
## Loading required package: S4Vectors
## Loading required package: stats4
##
## Attaching package: 'S4Vectors'
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
## The following object is masked from 'package:R.oo':
##
##     trim
## Loading required package: XVector
## Loading required package: GenomeInfoDb
##
## Attaching package: 'Biostrings'
## The following object is masked from 'package:base':
##
##     strsplit
library(seqinr)

```

```

##
## Attaching package: 'seqinr'
## The following object is masked from 'package:Biostrings':
##
##     translate
## The following object is masked from 'package:R.oo':
##
##     getName
library(dplyr)

##
## Attaching package: 'dplyr'
## The following object is masked from 'package:seqinr':
##
##     count
## The following objects are masked from 'package:Biostrings':
##
##     collapse, intersect, setdiff, setequal, union
## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect
## The following object is masked from 'package:XVector':
##
##     slice
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
library(ggplot2)
library(readr)
library(tidyr)

##
## Attaching package: 'tidyr'
## The following object is masked from 'package:S4Vectors':
##

```

```
##      expand
## The following object is masked from 'package:R.utils':
##
##      extract

2) Read in the gene expression data

#Download the data from the github link provided
URL = "https://raw.githubusercontent.com/ghazkha/Assessment4/refs/heads/main/gene_expression.tsv"
download.file(URL, destfile = "gene_expression.tsv")

# Read the downloaded TSV file into R
gene_expression <- read.table("gene_expression.tsv", header = TRUE, sep = "\t", row.names = 1)

3) 1st First 6 rows of the gene_expression data

# View the first few rows of the data
head(n=6, gene_expression)

##                                GTEX.1117F.0226.SM.5GZZ7 GTEX.1117F.0426.SM.5EGHI
## ENSG00000223972.5_DDX11L1                                0                      0
## ENSG00000227232.5_WASH7P                                187                    109
## ENSG00000278267.1_MIR6859-1                              0                      0
## ENSG00000243485.5_MIR1302-2HG                             1                      0
## ENSG00000237613.2_FAM138A                                0                      0
## ENSG00000268020.3_OR4G4P                                  0                      1
##                                GTEX.1117F.0526.SM.5EGHJ
## ENSG00000223972.5_DDX11L1                                0                      0
## ENSG00000227232.5_WASH7P                                143                     109
## ENSG00000278267.1_MIR6859-1                              1                      0
## ENSG00000243485.5_MIR1302-2HG                             0                      0
## ENSG00000237613.2_FAM138A                                0                      0
## ENSG00000268020.3_OR4G4P                                  0                      1
head( head (n=6, gene_expression))

##                                GTEX.1117F.0226.SM.5GZZ7 GTEX.1117F.0426.SM.5EGHI
## ENSG00000223972.5_DDX11L1                                0                      0
## ENSG00000227232.5_WASH7P                                187                    109
## ENSG00000278267.1_MIR6859-1                              0                      0
## ENSG00000243485.5_MIR1302-2HG                             1                      0
## ENSG00000237613.2_FAM138A                                0                      0
## ENSG00000268020.3_OR4G4P                                  0                      1
##                                GTEX.1117F.0526.SM.5EGHJ
## ENSG00000223972.5_DDX11L1                                0                      0
## ENSG00000227232.5_WASH7P                                143                     109
## ENSG00000278267.1_MIR6859-1                              1                      0
## ENSG00000243485.5_MIR1302-2HG                             0                      0
## ENSG00000237613.2_FAM138A                                0                      0
## ENSG00000268020.3_OR4G4P                                  0                      1
```

1.2 Make a new column which is the mean of the other columns. Show a table of values for the first six genes.

Calculate Mean Expression

```
# Calculate the mean across the samples and add as a new column
gene_expression <- gene_expression %>%
  mutate(mean_expression = rowMeans(select(., everything())))
# Show a table of values for the first six genes including the mean
head(n=6, gene_expression)
```

```
##                                GTEX.1117F.0226.SM.5GZZ7 GTEX.1117F.0426.SM.5EGHI
## ENSG00000223972.5_DDX11L1                                0                      0
## ENSG00000227232.5_WASH7P                                187                    109
## ENSG00000278267.1_MIR6859-1                              0                      0
## ENSG00000243485.5_MIR1302-2HG                             1                      0
## ENSG00000237613.2_FAM138A                                0                      0
## ENSG00000268020.3_OR4G4P                                  0                      1
##                                GTEX.1117F.0526.SM.5EGHJ mean_expression
## ENSG00000223972.5_DDX11L1                                0      0.0000000
## ENSG00000227232.5_WASH7P                                143     146.3333333
## ENSG00000278267.1_MIR6859-1                              1      0.3333333
## ENSG00000243485.5_MIR1302-2HG                             0      0.3333333
## ENSG00000237613.2_FAM138A                                0      0.0000000
## ENSG00000268020.3_OR4G4P                                  0      0.3333333
```

1.3 List the 10 genes with the highest mean expression.

Identify Top 10 Genes

```
# List the 10 genes with the highest mean expression
top_genes <- gene_expression %>%
  arrange(desc(mean_expression)) %>%
  head(10)
# Print the top genes
print(top_genes)
```

```
##                                GTEX.1117F.0226.SM.5GZZ7 GTEX.1117F.0426.SM.5EGHI
## ENSG00000198804.2_MT-CO1                                267250                    1101779
## ENSG00000198886.2_MT-ND4                                273188                    991891
## ENSG00000198938.2_MT-CO3                                250277                    1041376
## ENSG00000198888.2_MT-ND1                                243853                    772966
## ENSG00000198899.2_MT-ATP6                                141374                    696715
## ENSG00000198727.2_MT-CYB                                127194                    638209
## ENSG00000198763.3_MT-ND2                                159303                    543786
## ENSG00000211445.11_GPX3                                  464959                    39396
## ENSG00000198712.1_MT-CO2                                128858                    545360
## ENSG00000156508.17_EEF1A1                                317642                    39573
##                                GTEX.1117F.0526.SM.5EGHJ mean_expression
## ENSG00000198804.2_MT-CO1                                218923      529317.3
## ENSG00000198886.2_MT-ND4                                277628      514235.7
## ENSG00000198938.2_MT-CO3                                223178      504943.7
## ENSG00000198888.2_MT-ND1                                194032      403617.0
## ENSG00000198899.2_MT-ATP6                                151166      329751.7
## ENSG00000198727.2_MT-CYB                                141359      302254.0
## ENSG00000198763.3_MT-ND2                                149564      284217.7
## ENSG00000211445.11_GPX3                                  306070      270141.7
## ENSG00000198712.1_MT-CO2                                122816      265678.0
## ENSG00000156508.17_EEF1A1                                339347      232187.3
```

1.4 Determine the number of genes with a mean <10.

Count Genes with Low Expression (Mean < 10)

```
# Determine the number of genes with a mean < 10
num_genes_below_10 <- sum(gene_expression$mean_expression < 10)

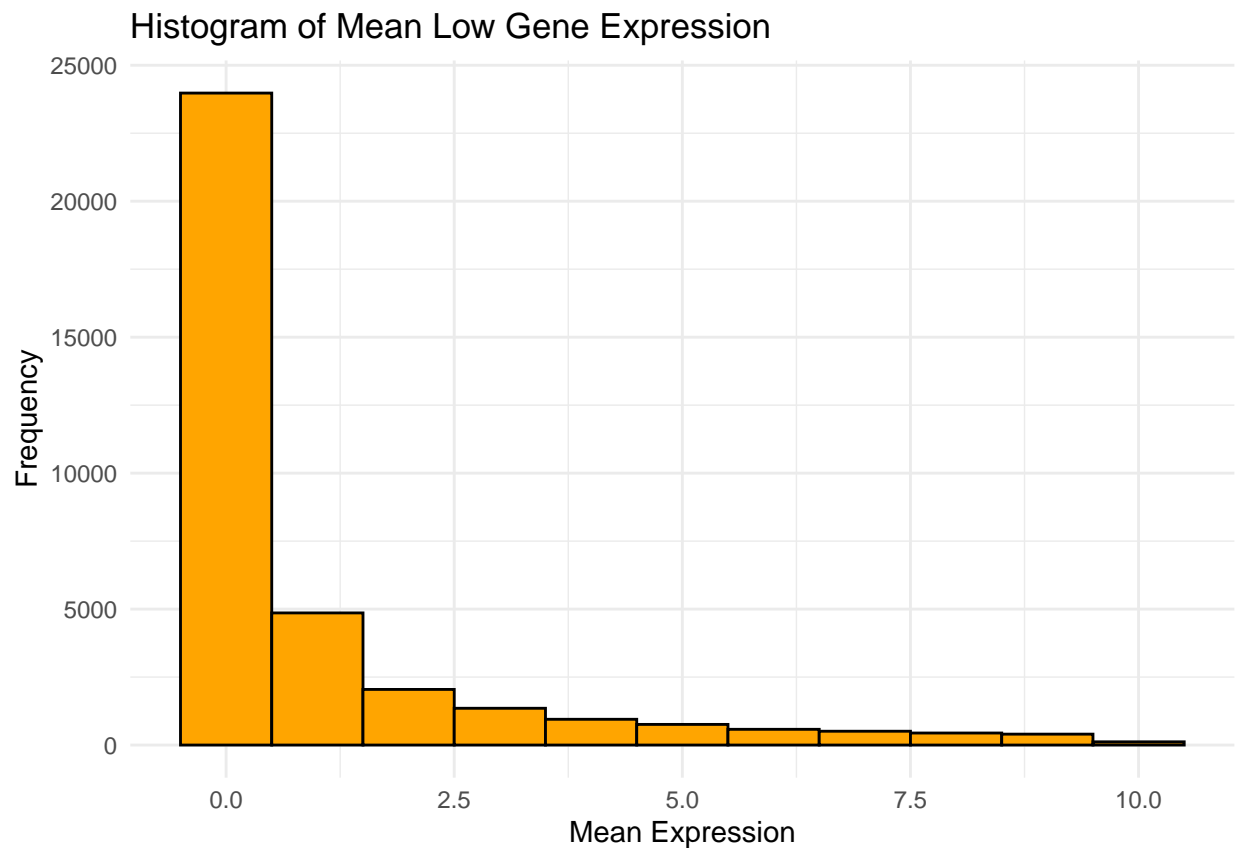
# Print the number of genes
print(num_genes_below_10)

## [1] 35988
```

1.5 Make a histogram plot of the mean values and include it into your report.

Histogram of <10, Mean Values

```
# Make a histogram plot of the mean values
filtered_data <- gene_expression[gene_expression$mean_expression < 10, ]
ggplot(filtered_data, aes(x = mean_expression)) +
  geom_histogram(binwidth = 1, fill = "orange", color = "black") +
  labs(title = "Histogram of Mean Low Gene Expression", x = "Mean Expression", y = "Frequency") +
  theme_minimal()
```



```
# Save the plot to your report
ggsave("histogram_mean_low_gene_expression.png")
```

```
## Saving 6.5 x 4.5 in image
```

Task 2: Tree circumference measurements over 20 years.

2.1 Import “growth_data.csv” file into an R object. What are the column names?

Read Data to perform a Tree Circumference Data Analysis

```
# Read in the growth data
#Download the data from the github link provided
URL = "https://raw.githubusercontent.com/ghazkha/Assessment4/refs/heads/main/growth_data.csv"
download.file(URL, destfile = "growth_data.csv")

# Read the downloaded TSV file into R
growth_data <- read.csv("growth_data.csv")

head(growth_data)
```

```
##           Site TreeID Circumf_2005_cm Circumf_2010_cm Circumf_2015_cm
## 1 northeast  A012           5.2           10.1           19.9
## 2 southwest  A039           4.9           9.6           18.9
## 3 southwest  A010           3.7           7.3           14.3
## 4 northeast  A087           3.8           6.5           10.9
## 5 southwest  A074           3.8           6.4           10.9
## 6 northeast  A008           5.9           10.0           16.8
## Circumf_2020_cm
## 1           38.9
## 2           37.0
## 3           28.1
## 4           18.5
## 5           18.4
## 6           28.4
```

```
# Show column names
cat("The column names are:", colnames(growth_data))
```

```
## The column names are: Site TreeID Circumf_2005_cm Circumf_2010_cm Circumf_2015_cm Circumf_2020_cm
```

2.2 Calculate the mean and standard deviation of tree circumference at the start and end of the study at both sites.

Statistics

```
# Calculate mean and standard deviation for tree circumference
summary_stats <- growth_data %>%
  summarise(mean_start_2005_southwest = mean(Circumf_2005_cm[Site == "southwest"]),
            sd_start_2005_southwest = sd(Circumf_2005_cm[Site == "southwest"]),
            mean_start_2005_northeast = mean(Circumf_2005_cm[Site == "northeast"]),
            sd_start_2005_northeast = sd(Circumf_2005_cm[Site == "northeast"]),
            mean_end_2020_southwest = mean(Circumf_2020_cm[Site == "southwest"]),
            sd_end_2020_southwest = sd(Circumf_2020_cm[Site == "southwest"]),
            mean_end_2020_northeast = mean(Circumf_2020_cm[Site == "northeast"]),
            sd_end_2020_northeast = sd(Circumf_2020_cm[Site == "northeast"])
  )

# Print summary statistics
print(summary_stats)
```

```
## mean_start_2005_southwest sd_start_2005_southwest mean_start_2005_northeast
## 1                4.862                1.147471                5.292
```

```
##      sd_start_2005_northeast mean_end_2020_southwest sd_end_2020_southwest
## 1              0.9140267              45.596              17.87345
##      mean_end_2020_northeast sd_end_2020_northeast
## 1              54.228              25.22795
```

2.3 Make a box plot of tree circumference at the start and end of the study at both sites.

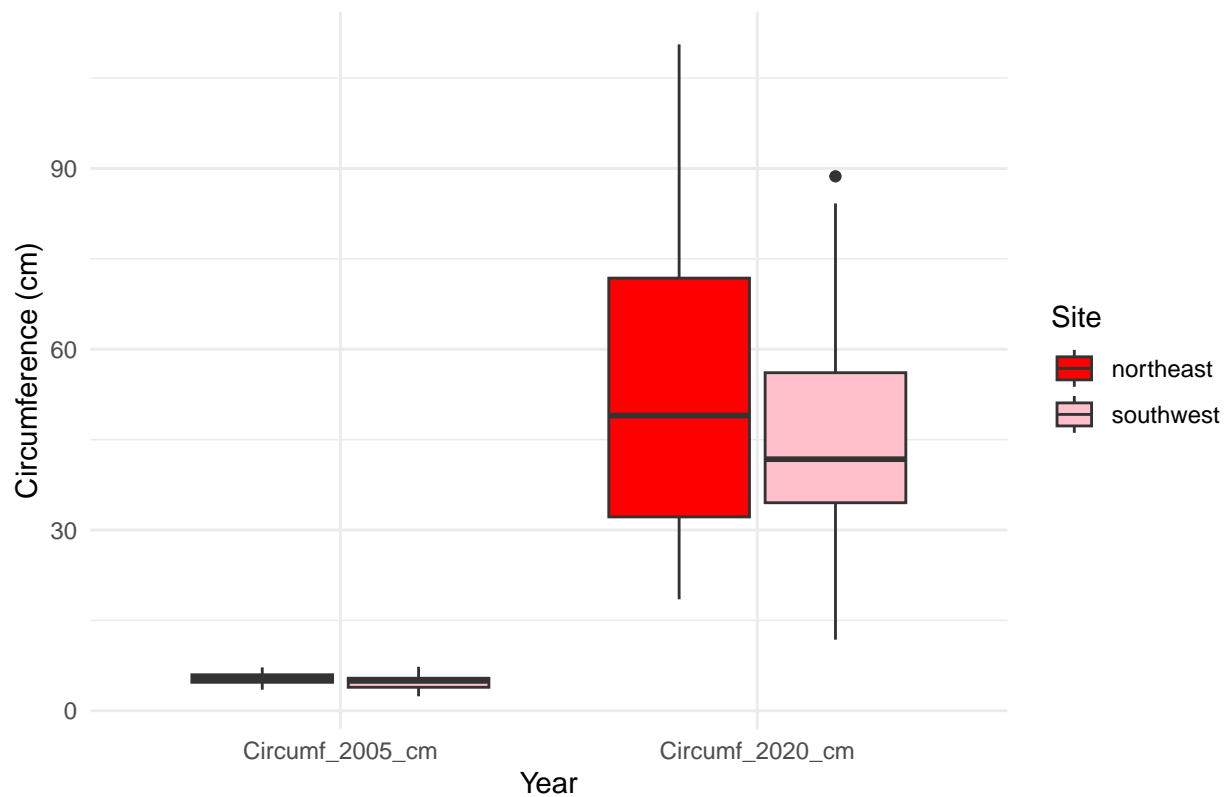
Boxplot of circumferences

```
# Reshape data from wide to long format for Circumf_2005_cm and Circumf_2020_cm
long_data <- growth_data %>%
  select(Site, TreeID, Circumf_2005_cm, Circumf_2020_cm) %>%
  pivot_longer(cols = starts_with("Circumf"),
               names_to = "Year",
               values_to = "Circumference")

# Filter for only the start and end years
long_data <- long_data %>%
  filter(Year %in% c("Circumf_2005_cm", "Circumf_2020_cm"))

# Create a box plot
ggplot(long_data, aes(x = Year, y = Circumference, fill = Site)) +
  geom_boxplot() +
  labs(title = "Tree Circumference at Start (2005) and End (2020) of Study",
       x = "Year",
       y = "Circumference (cm)") +
  scale_fill_manual(values = c("northeast" = "red", "southwest" = "pink")) +
  theme_minimal()
```


Tree Circumference at Start (2005) and End (2020) of Study



```
# Save the box plot to your report
ggsave("boxplot_tree_circumference.png")
```

```
## Saving 6.5 x 4.5 in image
```

2.4 Calculate the mean growth over the last 10 years at each site.

Mean Growth Calculation

```
# Calculate growth over the last 10 years for each tree
growth_data <- growth_data %>%
  mutate(Growth_10_years = Circumf_2020_cm - Circumf_2010_cm)

# Calculate mean growth at each site
mean_growth <- growth_data %>%
  group_by(Site) %>%
  summarise(mean_growth = mean(Growth_10_years, na.rm = TRUE), .groups = 'drop')

# Print the mean growth
print(mean_growth)
```

```
## # A tibble: 2 x 2
##   Site      mean_growth
##   <chr>      <dbl>
## 1 northeast    42.9
## 2 southwest    35.5
```

2.5 Use the t.test to estimate the p-value that the 10 year growth is different at the two sites.

T-Test for Growth Difference

```
# Perform t-test to compare growth between sites
t_test_result <- t.test(Growth_10_years ~ Site, data = growth_data)

# Print t-test results
print(t_test_result)

##
## Welch Two Sample t-test
##
## data: Growth_10_years by Site
## t = 1.8882, df = 87.978, p-value = 0.06229
## alternative hypothesis: true difference in means between group northeast and group southwest is not equal to 0
## 95 percent confidence interval:
## -0.3909251 15.2909251
## sample estimates:
## mean in group northeast mean in group southwest
## 42.94 35.49
```

Interpretation: p-value: The p-value of 0.06229 suggests that the difference in mean growth between the two sites is not statistically significant at the conventional alpha level of 0.05. However, it is close to this threshold, indicating a potential trend toward significance.

Mean Comparison: The mean growth in the northeast (42.94 cm) is higher than that in the southwest (35.49 cm). This suggests that trees in the northeast experienced greater growth compared to those in the southwest over the last 10 years.

Confidence Interval: The confidence interval includes zero, which means we cannot conclusively say that there is a true difference in growth between the two sites. The upper limit (15.29 cm) indicates that, while the northeast shows higher growth, it is possible that the actual difference might be minimal or even negative.