Bioinformatics Analysis and Visualisation of Medical Genomics Data

KI course number 5633, March 2023

Luuk Harbers

2023-03-10

Table of Contents

## Task 1 - Literature

Answer the following questions:

**1. What is the medically relevant insight from the article?**

The authors identified how insulin-regulated genes, pathways and Transcription Factors respond to changes in body weight. They observe that the transcription response in human White Adipose Tissue (WAT) is selective. Furthermore, they observe that the insulin response in WAT remains present in severely Obese Women (OB). Concluding that the supressive effects of insulin are attenuated, the stimulatory effects remain largely intact.

**2. Which genomics technology/technologies were used?**

The genomics technology used in this paper is CAGE.

Further related research questions:

**3. List and explain at least three questions/ hypotheses you can think of that extend the analysis presented in the paper.**

Three potential questions/hypotheses are:

1. Besides the gene enrichment analysis, One could also look at the network of genes that are upregulated/downregulated in the different conditions directly. It would be interesting to see how networks of the altered genes look like, whether in these networks you would also see three clearly distinct groups or whether these are more tightly connected.
2. It would also be interesting to see if genes that are implicated in either of the identified groups of differentially expressed genes have different usages of transcription start sites compared to background genes.
3. The authors mentioned that they collected samples relatively soon after the clamp procedure to ensure they do not measure the Secondary effects. However, it would be interesting to also look at the secondary effects if it were possible to collect an additional (slightly later) sample.

## Task 4 - R basic operations

1. What is the square root of 10?

sqrt(10) = 3.1622777

1. What is the logarithm of 32 to the base 2?

log2(32) = 5

1. What is the sum of the numbers from 1 to 1000?

sum(1:1000) = 500500

1. What is the sum of all even numbers from 2 to 1000?

sum(seq(2, 1000, 2)) = 250500

1. How many pairwise comparisons are there for 100 genes?

100 \* (100 - 1) / 2 = 4950

1. And how many ways to arrange 100 genes in triples?

factorial(100) / factorial(100-3) = 970200

## Task 5 - Using R example datasets

1. Use the R internal CO2 dataset (“data(CO2)”).

# Load in data  
data(CO2)

1. Describe briefly the content of the CO2 dataset using the help function.

# Show the help output in Rstudio (or base R)  
help(CO2)

The help function shows that CO2 is a data frame with 84 rows and 5 columns of daata from an experiment on the cold tolerance of the grass species Echinochloa crus-galli.

1. What is the average and median CO2 uptake of the plants from Quebec and Mississippi?

# Calculate mean  
mean\_qb = mean(CO2[CO2$Type == "Quebec", "uptake"])  
mean\_ms = mean(CO2[CO2$Type == "Mississippi", "uptake"])  
  
# Print results  
paste0("The mean uptake of plans in Quebec is: ", mean\_qb)

## [1] "The mean uptake of plans in Quebec is: 33.5428571428571"

paste0("The mean uptake of plans in Mississippi is: ", mean\_ms)

## [1] "The mean uptake of plans in Mississippi is: 20.8833333333333"

# Calculate median  
median\_qb = median(CO2[CO2$Type == "Quebec", "uptake"])  
median\_ms = median(CO2[CO2$Type == "Mississippi", "uptake"])  
  
# Print results  
paste0("The median uptake of plants in Quebec is: ", median\_qb)

## [1] "The median uptake of plants in Quebec is: 37.15"

paste0("The median uptake of plants in Mississippi is: ", median\_qb)

## [1] "The median uptake of plants in Mississippi is: 37.15"

1. [Optional] In the “airway” example data from Bioconductor, how many genes are expressed in each sample? How many genes are not expressed in any sample?

# Load the airway library or install it if it's not present  
if (!require(airway)) BiocManager::install("airway")  
  
# Load in airway data  
data("airway")  
  
# Check what kind of data airway is  
head(airway)

## class: RangedSummarizedExperiment   
## dim: 6 8   
## metadata(1): ''  
## assays(1): counts  
## rownames(6): ENSG00000000003 ENSG00000000005 ... ENSG00000000460  
## ENSG00000000938  
## rowData names(0):  
## colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521  
## colData names(9): SampleName cell ... Sample BioSample

# See that it is a RangedSummarizedExperiment so check the assays  
assays(airway)

## List of length 1  
## names(1): counts

# Get gene count data and transform it to data.table  
counts = assays(airway)$counts  
  
# Count the number of genes expressed per sample  
colSums(counts > 0)

## SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517 SRR1039520   
## 24633 24527 25699 23124 25508 25998 24662   
## SRR1039521   
## 23991

## Task 6 - R functions

1. Write a function that calculates the ratio of the mean and the median of a given vector. This is a helpful measure to detect data with outlying values. Note: See Reference for R language

# Function to calculate the ratio of mean / median of a vector  
calcRatio = function(vec) {  
 return(mean(vec) / median(vec))  
}

1. Write a function that ignores the lowest and the highest value from a given vector and calculate the mean.

# Function to calculate the mean of a vector without the highest and lowest value  
calcAdjMean = function(vec) {  
 vec = sort(vec) # Sort vector  
 vec = vec[c(-1, -length(vec))] # Remove first and last value  
 return(mean(vec)) # Return mean  
}

1. Read about piping from here:<https://r4ds.had.co.nz/pipes.html#pipes> (you don’t have to learn everything, a basic understanding of the usage is enough). Write a short (max. 300 characters, no spaces) explanation of why, how, and when not to use pipes.

One of the major advantages of piping is that it can make your code look easier to read and easier to interpret. When you want to chain functions you can use the pipe operator %>% (or |> in base R). However, you should not use pipes for (very) long sequences, multiple inputs/outputs or for complex relationships between the operations.

1. Familiarize yourself with the apply-family of functions (apply, lapply, sapply etc.) <http://uc-r.github.io/apply_family> Write a short explanation (max. 300 characters, no spaces) of why they could be useful in your work.

the apply-family can be very useful if you want to perform some type of operation over multiple columns/rows in a data.frame. Or, for me personally, you can use lapply() or sapply() when performing some function(s) over a list of patient samples. These can also easily be parallelized using mcapply().

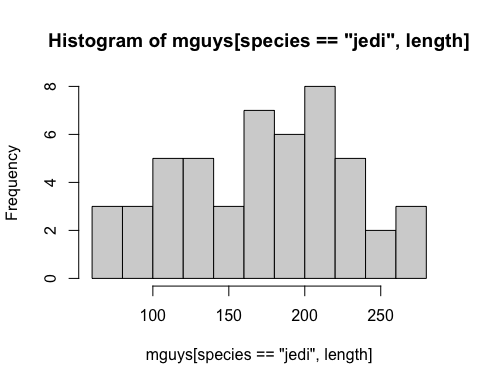
## Task 7 - Basic visualization with R

1. Compare the distributions of the body heights of the two species from the ‘magic\_guys.csv’ dataset graphically

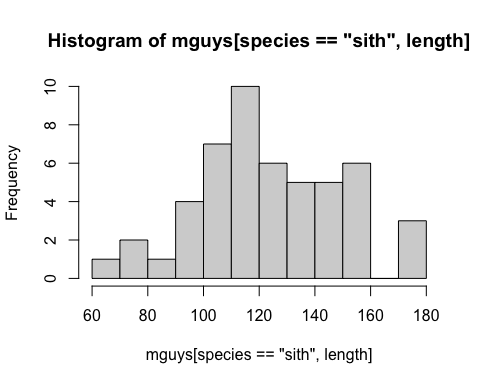
require(data.table) # Load data.table  
mguys = fread("~/Downloads/magic\_guys.csv") # Load in dataset

a) using the basic 'hist' function as well as ‘ggplot’ and ‘geom\_histogram’ functions from the ggplot2 package. Optimize the plots for example by trying several different 'breaks'. Note that ggplot2-based functions give you many more options for changing the visualization parameters, try some of them.  
b) Do the same comparison as in a. but with boxplots. If you want to use the ggplot2-package, use the functions ‘ggplot’ and ‘geom\_boxplot’.  
c) Save the plots with the ‘png’, ‘pdf’, and ‘svg’ formats. In which situation would you use which file format?

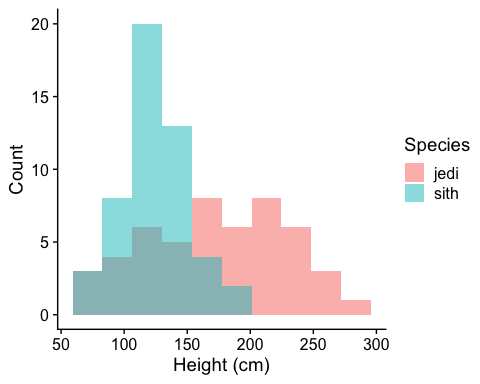
# Load libraries  
require(ggplot2)  
require(cowplot)  
  
# Set breaks  
breaks = 10  
  
# Plot hist using base R  
hist(mguys[species == "jedi", length], breaks = breaks)



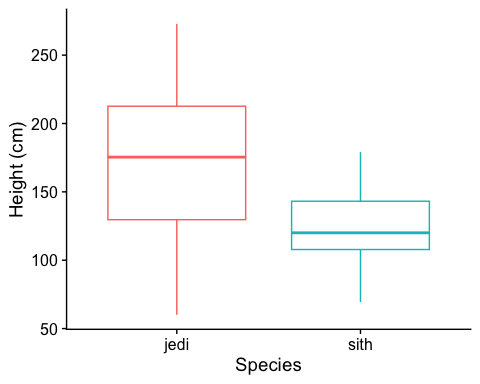
hist(mguys[species == "sith", length], breaks = breaks)



# Plot with ggplot2  
theme\_set(theme\_cowplot()) # Set theme  
  
# Plot  
plt\_hist = ggplot(mguys, aes(x = length, fill = species)) +  
 geom\_histogram(position = "identity", alpha = 0.5, bins = breaks) +  
 labs(y = "Count", x = "Height (cm)", fill = "Species")  
plt\_hist



# Plot boxplots  
plt\_box = ggplot(mguys, aes(x = species, y = length, color = species)) +  
 geom\_boxplot() +  
 labs(y = "Height (cm)", x = "Species") +  
 theme(legend.position = "none")  
plt\_box



# Save the histograms  
ggsave("~/Documents/KI-UT\_Course/plots/assignment\_1/height\_hist\_plot.png", plot = plt\_hist) #png for quick saving and sharing  
ggsave("~/Documents/KI-UT\_Course/plots/assignment\_1/height\_hist\_plot.pdf", plot = plt\_hist) #pdf to keep resolution  
ggsave("~/Documents/KI-UT\_Course/plots/assignment\_1/height\_hist\_plot.svg", plot = plt\_hist) #svg to be able to make changes in illustrator etc for publications  
  
# Save the boxplots  
ggsave("~/Documents/KI-UT\_Course/plots/assignment\_1/height\_boxplot.png", plot = plt\_box) #png for quick saving and sharing  
ggsave("~/Documents/KI-UT\_Course/plots/assignment\_1/height\_boxplot.pdf", plot = plt\_box) #pdf to keep resolution  
ggsave("~/Documents/KI-UT\_Course/plots/assignment\_1/height\_boxplot.svg", plot = plt\_box) #svg to be able to make changes in illustrator etc for publications

1. Load the gene expression data matrix from the ‘microarray\_data.tab’ dataset provided in the shared folder, it is a big tabular separated matrix.

array = fread("~/Downloads/microarray\_data.tab")

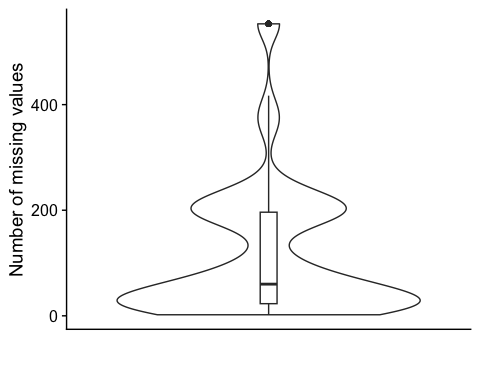
a) How big is the matrix in terms of rows and columns?  
b) Count the missing values per gene and visualize this result.  
c) Find the genes for which there are more than X% (X=10%, 20%, 50%)

missing values. d) Replace the missing values by the average expression value for the particular gene. (Note: Imputing data has to be used with caution!)

# To get the dimensions of the microarray  
dim(array)

## [1] 553 1000

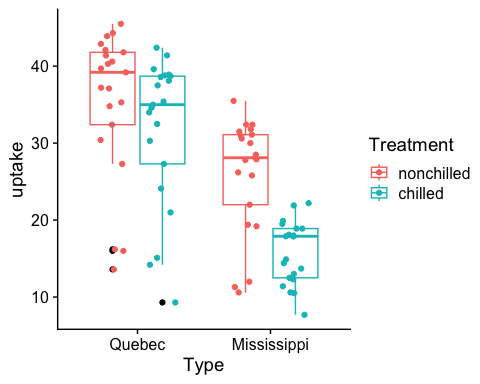
# Calculate missing values per gene  
missing = data.table(gene = colnames(array), n\_missing = colSums(is.na(array)))  
  
# Plot number of missing values per gene  
ggplot(missing, aes(x = "", y = n\_missing)) +  
 geom\_violin() +  
 geom\_boxplot(width = .05) +  
 labs(y = "Number of missing values", x = "") +  
 theme(axis.ticks.x = element\_blank())



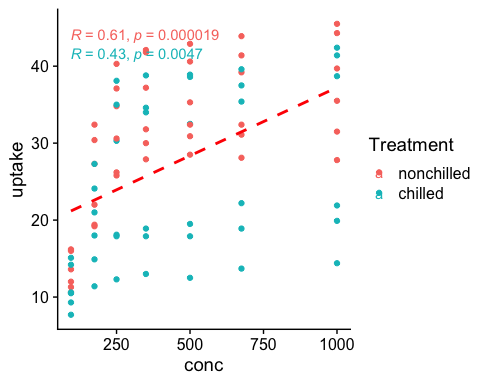
# Find genes where more than X% are missing values  
missing[, fraction := n\_missing / nrow(array)]  
miss\_10 = missing[fraction > .1, gene]  
miss\_20 = missing[fraction > .2, gene]  
miss\_50 = missing[fraction > .5, gene]  
  
# Impute values for missing data  
imputed\_array = apply(array, 2, function(x) {  
 avg = mean(x, na.rm = T)  
 nafill(x, fill = avg)  
})

1. Visualize the data in the CO2 dataset in a way that gives you a deeper understanding of the data. What do you see?

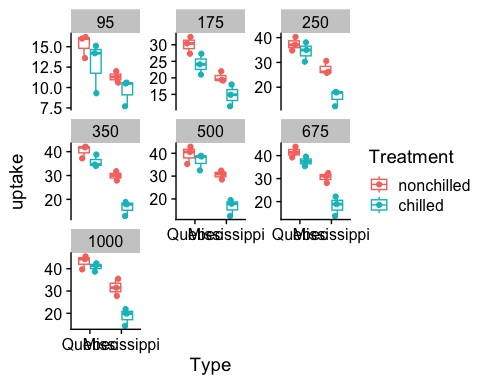
# Load libraries  
require(ggpubr)  
  
# Set DT  
CO2 = CO2  
setDT(CO2)  
  
# Plot uptake  
ggplot(CO2, aes(x = Type, y = uptake, color = Treatment)) +  
 geom\_boxplot(outlier.colour = "black") +  
 geom\_point(position = position\_jitterdodge())



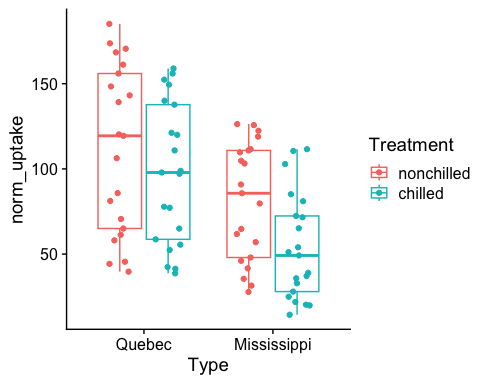
# Plot uptake vs ambient CO2 concentration  
ggplot(CO2, aes(x = conc, y = uptake, color = Treatment)) +  
 geom\_point() +  
 stat\_cor() +  
 geom\_smooth(method = "lm", se = FALSE, col = "red", linetype = 2)



# Plot uptake  
ggplot(CO2, aes(x = Type, y = uptake, color = Treatment)) +  
 geom\_boxplot() +  
 geom\_point(position = position\_jitterdodge()) +  
 facet\_wrap(~conc, scales = "free\_y")



# Normalizing linearly by concentration is not the best but will do so anyway..  
  
# Normalize for ambient CO2 concentration  
CO2[, norm\_uptake := uptake / conc \* 1000]  
  
# Plot norm uptake  
ggplot(CO2, aes(x = Type, y = norm\_uptake, color = Treatment)) +  
 geom\_boxplot(outlier.colour = "black") +  
 geom\_point(position = position\_jitterdodge())

 I see that Mississippi has a low overal uptake compared to Quebec, even when accounting for the CO2 concentration. Furthermore, Chilled treatment results in a low overal uptake compared to nonchilled treatment.

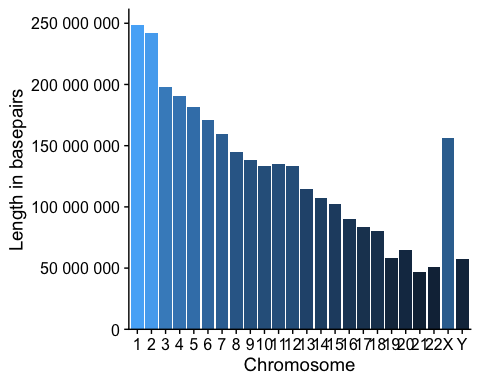
## Task 8 - Tidybiology

1. Install the Tidybiology package, which includes the data ‘chromosome’ and ‘proteins’ devtools::install\_github(“hirscheylab/tidybiology”)
2. Extract summary statistics (mean, median and maximum) for the following variables from the ‘chromosome’ data: variations, protein coding genes, and miRNAs. Utilize the tidyverse functions to make this as simply as possible.
3. How does the chromosome size distribute? Plot a graph that helps to visualize this by using ggplot2 package functions.
4. Does the number of protein coding genes or miRNAs correlate with the length of the chromosome? Make two separate plots to visualize these relationships.
5. Calculate the same summary statistics for the ‘proteins’ data variables length and mass. Create a meaningful visualization of the relationship between these two variables by utilizing the ggplot2 package functions. Play with the colors, theme- and other visualization parameters to create a plot that pleases you.

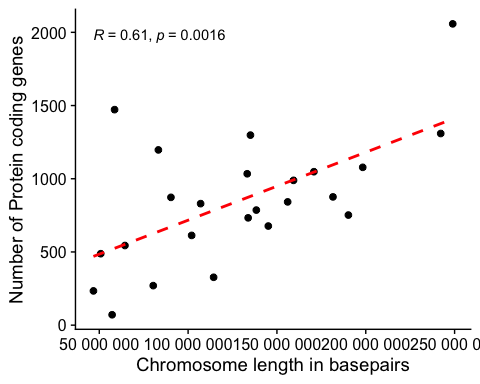
# Load libraries  
require(tidybiology)   
require(scales)  
require(patchwork)  
  
# Load data and setDT  
data(chromosome)  
setDT(chromosome)  
  
# Get summary for variations, protein coding genes and miRNAs  
summary(chromosome[, .(variations, protein\_codinggenes, mi\_rna)])

## variations protein\_codinggenes mi\_rna   
## Min. : 211643 Min. : 71.0 Min. : 15.00   
## 1st Qu.: 4395298 1st Qu.: 595.8 1st Qu.: 55.75   
## Median : 6172346 Median : 836.0 Median : 75.00   
## Mean : 6484572 Mean : 850.0 Mean : 73.17   
## 3rd Qu.: 8742592 3rd Qu.:1055.5 3rd Qu.: 92.00   
## Max. :12945965 Max. :2058.0 Max. :134.00

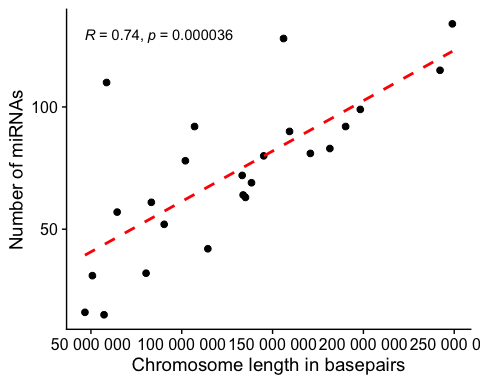
# Plot chromosome sizes  
ggplot(chromosome, aes(x = id, y = basepairs, fill = basepairs)) +  
 geom\_col() +  
 labs(y = "Length in basepairs", x = "Chromosome") +  
 scale\_y\_continuous(expand = expansion(mult = c(0, .05)),   
 labels = scales::number\_format()) +  
 theme(legend.position = "none")



# Plot chromosome length vs number of protein coding genes  
ggplot(chromosome, aes(x = basepairs, y = protein\_codinggenes)) +  
 geom\_point(size = 2) +  
 stat\_cor() +   
 labs(y = "Number of Protein coding genes", x = "Chromosome length in basepairs") +  
 geom\_smooth(method = "lm", se = FALSE, linetype = 2, color = "red") +  
 scale\_x\_continuous(labels = number\_format())



ggplot(chromosome, aes(x = basepairs, y = mi\_rna)) +  
 geom\_point(size = 2) +  
 stat\_cor() +   
 labs(y = "Number of miRNAs", x = "Chromosome length in basepairs") +  
 geom\_smooth(method = "lm", se = FALSE, linetype = 2, color = "red") +  
 scale\_x\_continuous(labels = number\_format())



# Yes the chromosome length does correlate with the number of protein coding genes and miRNAs  
  
# Get summary for protein length and mass  
data(proteins)  
setDT(proteins)  
summary(proteins[, .(length, mass)])

## length mass   
## Min. : 2.0 Min. : 260   
## 1st Qu.: 251.0 1st Qu.: 27940   
## Median : 414.0 Median : 46140   
## Mean : 557.2 Mean : 62061   
## 3rd Qu.: 669.0 3rd Qu.: 74755   
## Max. :34350.0 Max. :3816030

# Plot length vs mass  
ggplot(proteins, aes(x = length, y = mass, color = length)) +  
 geom\_point() +  
 labs(y = "Mass of protein", x = "Length of protein") +  
 scale\_y\_log10(labels = number\_format()) +   
 scale\_x\_log10(labels = number\_format()) +  
 stat\_cor() +   
 geom\_smooth(method = "lm", se = FALSE, linetype = 2, color = "red") +  
 scale\_color\_viridis\_c(oob = squish, limits = c(0, 5000)) +  
 theme(legend.position = "none")

