Introduction to R part 2

## Learning objectives

* Data wrangling
* Basic plotting and statistics

library(knitr)  
getwd()

## [1] "/Users/anastasia/Desktop/BioinfomraticsGenomics"

##Open a data file and look at the data Download Gene\_expreesion\_matrix.txt from Moodle

This is a real RNA-Seq counts data file. Where we took gene expression from 2 human fibroblasts, and then did a serum challenge that can mimic wound healing and gene expression patterns seen in some cancers. h1 and h2 are the biological replicates (different individuals) and P is Pre-challenge, and T12 is 12 hours after the challenge.

There are many options to open files in R, but R studio makes it easy through the regular open function. To add the data matrix to the environment, go to the right top panel and import dataset from base. IMPORTANT, click yes in the heading row. This will give the columns the correct names.

Did it work? UNIX commands also work in R. Here use head() to look at the top of the data

Gene\_expression\_matrix<-read.table("Gene\_expression\_matrix.txt", header=T, sep="\t")

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :  
## number of items read is not a multiple of the number of columns

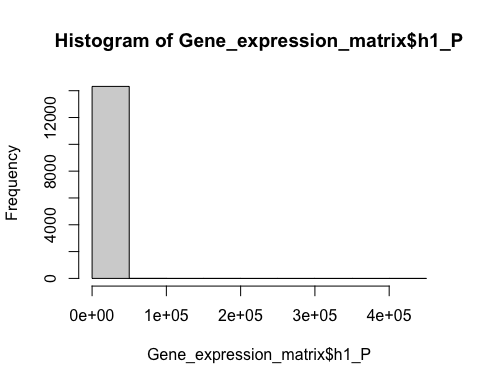
head(Gene\_expression\_matrix)

## geneName h1\_P h2\_P h1\_T12 h2\_T12  
## 1 A1BG 4 15 13 6  
## 2 A1CF 0 0 0 0  
## 3 A2LD1 0 2 1 0  
## 4 A2M 117 404 43 176  
## 5 A2ML1 1 0 1 0  
## 6 A4GALT 174 351 93 163

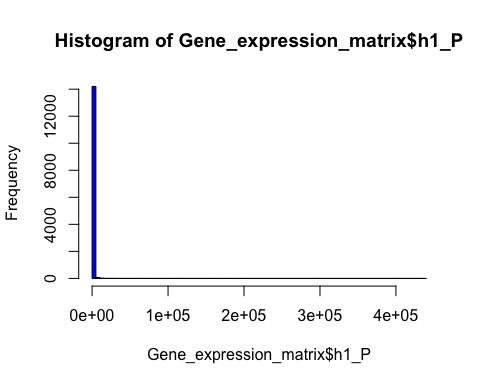
One of the most important things we need to do when looking at large datasets it to look at and understand the underlying distribution. Here we use the histogram tool to do that, hist(). We’ll only plot one of the columns at a time today.

###Analyizing parts of a larger dataset A really important shorthand in R is the $. The dollar sign $ lets you grab one element of a matrix (or other object). Here, if we want to plot the first data column we can use the matrix$column\_name, here Gene\_expression\_matrix$h1\_P.

hist(Gene\_expression\_matrix$h1\_P)

 Not super informative. is it? How can we see more about the data? Let’s break it up more.

hist(Gene\_expression\_matrix$h1\_P, breaks=100, col="blue")

 still not much there. There seems to be a lot in the left blue bar, but the x-axis keeps going, indicating that there are data points that far off. We need to look at a summary of our data.

What is the distribution of the count numbers? With Big Data and Genomics, sometimes the best way to start to understand the data is with summary statistics (seemingly boring things like the max, min, mean….)

summary(Gene\_expression\_matrix)

## geneName h1\_P h2\_P h1\_T12   
## Length:14331 Min. : 0.0 Min. : 0 Min. : 0.0   
## Class :character 1st Qu.: 1.0 1st Qu.: 1 1st Qu.: 1.0   
## Mode :character Median : 53.0 Median : 58 Median : 55.0   
## Mean : 411.4 Mean : 423 Mean : 447.6   
## 3rd Qu.: 257.0 3rd Qu.: 271 3rd Qu.: 295.0   
## Max. :438819.0 Max. :423133 Max. :347220.0   
## NA's :1 NA's :1 NA's :1   
## h2\_T12   
## Min. : 0.0   
## 1st Qu.: 1.0   
## Median : 45.0   
## Mean : 389.1   
## 3rd Qu.: 247.0   
## Max. :237213.0   
## NA's :1

##Subset the data So, we have a lot of low expression values that we might be less confident of since we only have two replicates. One next step to to look at the data again after removing these counts. Subset is a powerful function to subset data to analyze separately and better understand how changing the underlying distribution changes downstream analyses. We can use logical operators <, > & (and), | (or), == (equals), or != (does not equal) to ask for the data under a subset of conditions. Here all column counts need to be greater than or equal to 10. Make sure you can find where we specify that in the code before running this section.

data\_new<- subset(Gene\_expression\_matrix, Gene\_expression\_matrix$h1\_P >= 10 & Gene\_expression\_matrix$h2\_P >= 10 & Gene\_expression\_matrix$h1\_T12 >= 10 & Gene\_expression\_matrix$h2\_T12 >= 10)  
  
summary(data\_new)

## geneName h1\_P h2\_P h1\_T12   
## Length:8545 Min. : 10.0 Min. : 10.0 Min. : 10.0   
## Class :character 1st Qu.: 78.0 1st Qu.: 86.0 1st Qu.: 85.0   
## Mode :character Median : 197.0 Median : 208.0 Median : 225.0   
## Mean : 687.2 Mean : 706.8 Mean : 748.2   
## 3rd Qu.: 479.0 3rd Qu.: 495.0 3rd Qu.: 556.0   
## Max. :438819.0 Max. :423133.0 Max. :347220.0   
## h2\_T12   
## Min. : 10.0   
## 1st Qu.: 70.0   
## Median : 185.0   
## Mean : 650.9   
## 3rd Qu.: 453.0   
## Max. :237213.0

##Questions: 1. Subset the data to exclude really large counts. Use the information from summary (and the x-axis of your graph) to get an idea of what that threshold might be. 1000? 10,000?

data\_new2<- subset(Gene\_expression\_matrix, Gene\_expression\_matrix$h1\_P <= 100000 & Gene\_expression\_matrix$h2\_P <= 100000 & Gene\_expression\_matrix$h1\_T12 <= 100000 & Gene\_expression\_matrix$h2\_T12 <= 100000)  
  
summary(data\_new2)

## geneName h1\_P h2\_P h1\_T12   
## Length:14326 Min. : 0.0 Min. : 0.0 Min. : 0   
## Class :character 1st Qu.: 1.0 1st Qu.: 1.0 1st Qu.: 1   
## Mode :character Median : 53.0 Median : 57.0 Median : 55   
## Mean : 348.3 Mean : 359.8 Mean : 390   
## 3rd Qu.: 256.0 3rd Qu.: 271.0 3rd Qu.: 294   
## Max. :77523.0 Max. :92030.0 Max. :87051   
## h2\_T12   
## Min. : 0.0   
## 1st Qu.: 1.0   
## Median : 45.0   
## Mean : 344.6   
## 3rd Qu.: 247.0   
## Max. :89458.0

1. What did the subsetting in data\_new do to the data matrix (e.g. total gene count, min, max, and mean)?

Subsetting reduced the total gene count. Min, max and mean all decreased as there is now a constraint of 100,000 on each count.

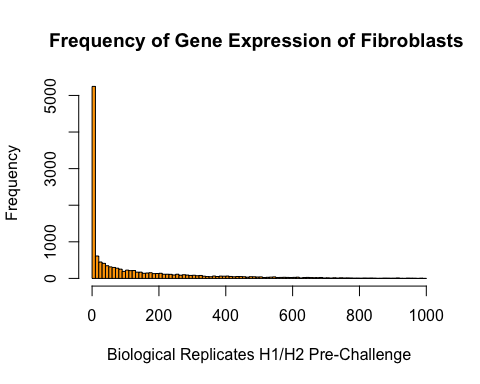
1. Edit the subset code above and use summary and hist to describe and plot your new subsetted data matrix. Be sure to add axis labels and a title (and color).

data\_new3 <- subset(Gene\_expression\_matrix, Gene\_expression\_matrix$h1\_P <= 1000 & Gene\_expression\_matrix$h2\_P <= 1000 & Gene\_expression\_matrix$h1\_T12 <= 1000 & Gene\_expression\_matrix$h2\_T12 <= 1000)  
  
summary(data\_new3)

## geneName h1\_P h2\_P h1\_T12   
## Length:12999 Min. : 0 Min. : 0.0 Min. : 0.0   
## Class :character 1st Qu.: 1 1st Qu.: 0.0 1st Qu.: 0.0   
## Mode :character Median : 35 Median : 37.0 Median : 34.0   
## Mean :122 Mean :126.8 Mean : 137.6   
## 3rd Qu.:175 3rd Qu.:187.0 3rd Qu.: 199.0   
## Max. :994 Max. :998.0 Max. :1000.0   
## h2\_T12   
## Min. : 0.0   
## 1st Qu.: 0.0   
## Median : 27.0   
## Mean : 116.7   
## 3rd Qu.: 165.0   
## Max. :1000.0

hist(data\_new3$h1\_P, data\_new3$h2\_P, breaks=100, main = "Frequency of Gene Expression of Fibroblasts ", xlab = "Biological Replicates H1/H2 Pre-Challenge",ylab = "Frequency", col="orange")

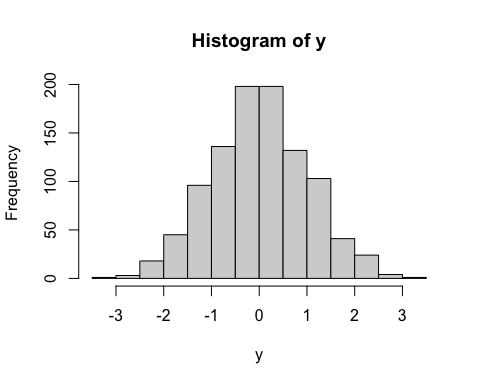
## Warning in if (freq) x$counts else x$density: the condition has length > 1 and  
## only the first element will be used

 ## Distributions and Statistics

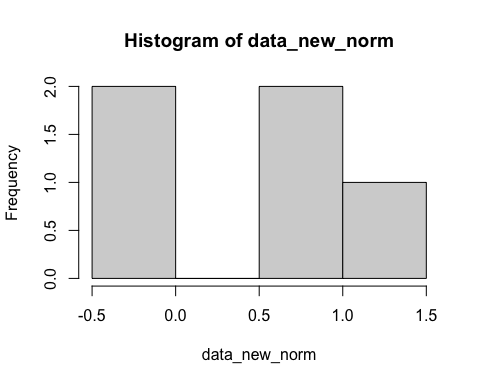
One of the challenges with real biological data is that it is not alway normal. Many statistical test default assume that there is a normal distribution.

Here’s some generated normal data

y <- rnorm(1000)  
hist(y)

 2. Let us plot the normal data against our subseted data. If they have the same distribution, we will see a straight line. For normally distributed data, we use parametric statistics. a) use the command qqplot to plot (#your data column, y). Plot it here.

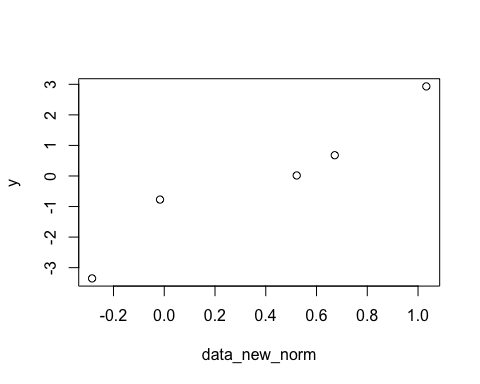
data\_new3 <- subset(Gene\_expression\_matrix, Gene\_expression\_matrix$h1\_P <= 1000 & Gene\_expression\_matrix$h2\_P <= 1000 & Gene\_expression\_matrix$h1\_T12 <= 1000 & Gene\_expression\_matrix$h2\_T12 <= 1000)  
y <- rnorm(1000)  
  
data\_new\_norm <- rnorm(data\_new3)  
hist(data\_new\_norm)



head(data\_new\_norm)

## [1] 0.52198872 -0.01667737 -0.28433138 0.67189376 1.03263467

#data\_new\_norm#H1\_P throws error  
qqplot(data\_new\_norm, y)



1. So, if the data is not normally distributed we need to use non-parametric statistics. Sometimes this only has a slight difference, sometimes more. Let’s look at the differences in correlations using cor.test
2. perform cor.test between sample h2 at the P and T12 timepoints

cor.test(data\_new3$h2\_P, data\_new3$h2\_T12)

##   
## Pearson's product-moment correlation  
##   
## data: data\_new3$h2\_P and data\_new3$h2\_T12  
## t = 201.69, df = 12997, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.8663236 0.8746509  
## sample estimates:  
## cor   
## 0.8705496

1. perform a non-parametric cor.test between the same two columns, using cor.test(“a”, “b”, method=“spearman”)

cor.test(data\_new3$h2\_P, data\_new3$h2\_T12, method="spearman")

## Warning in cor.test.default(data\_new3$h2\_P, data\_new3$h2\_T12, method =  
## "spearman"): Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: data\_new3$h2\_P and data\_new3$h2\_T12  
## S = 1.514e+10, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.958644