USC Department Of Translational Genomics
Keck Medicine of USC

TRGN 527: Applied Data Science and Bioinformatics

UNIT I. Introduction and Basic Data Science

Week 1 – Lecture 4

Enrique I. Velazquez Villarreal, M.D., Ph.D., M.P.H., M.S. | **Assistant Professor**

Dept. of Translational Genomics

USC | Keck School of Medicine | Norris Comprehensive Cancer Center

Leader of the USC Bioinformatics Core – *USC CaRE2 Health Equity Center*

David W. Craig, Ph.D. | **Professor and Vice Chair**

Dept. of Translational Genomics

USC | Keck School of Medicine | Norris Comprehensive Cancer Center

Co-Director, Institute of Translational Genomics

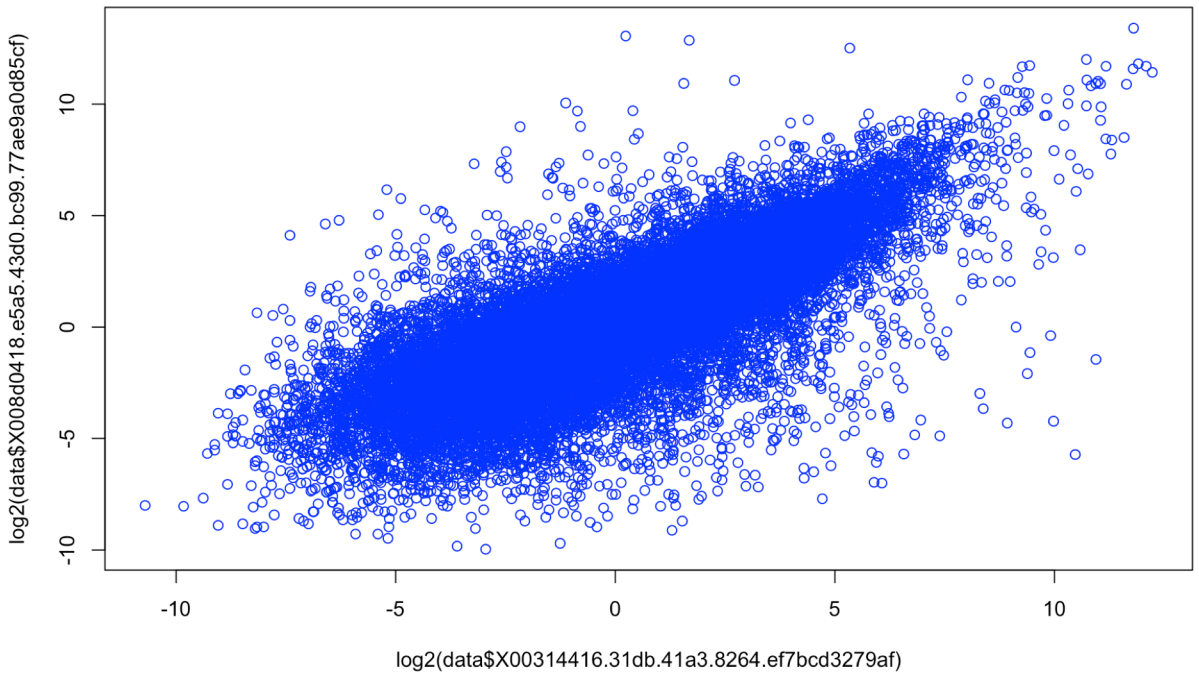
Topics

- Data aggregation and pivoting examples. Joining of tables. Loading in data, data frames, Data types (numerical, categorical, ordinal).

Reading TCGA Data Files

- Loading TCGA data file into R.
  - First, look at the content of the data “trgn599.rna.tsv” (download the file from BB)
    - The file contains gene expression data – breast cancer samples.
  - The file is a tab separated table.
  - Load the content of file “trgn599.rna.tsv” into the variable data.
    - The path is skipped because the file sits in the same folder from where R is called.
    - The argument header is set to TRUE, because the first line in the data file contains the column names.
    - As file separator, the tabulator is set with the argument sep.
  - Create a plot of the data.

````
---
title: "TRGN599_Week_1_Lecture_4"
author: "EIVV"
output: html_document
---

## Importing TCGA Data


```{R}
# Check your current working directory
getwd()

# Set your working directory
setwd("/Users/enriquevelazquez/Documents/R_working_directory")

# Upload your file
data <- read.table("trgn599.rna.tsv", header = TRUE, sep = "\t")

# Display the names of the header
# since it is a large table display the first five names

allheaders <- names(data)
head(allheaders)


# Plot the lo2 two of the first two samples of the dataset
plot(log2(data$X00314416.31db.41a3.8264.ef7bcd3279af),
log2(data$X008d0418.e5a5.43d0.bc99.77ae9a0d85cf), col="blue")
```
````



Writing TCGA Data Files

- Write TCGA data into a file.
  - Use arguments for write.table() function.

```R
# Write a table with column names and without row names
write.table(data, "output-data_1.csv", append=FALSE, col.names=TRUE,
          row.names = FALSE, quote = FALSE, sep = ",")

# Display first five rows and first two columns
data_1 <- read.table("output-data_1.csv", header = TRUE, sep = ",")
data_1[1:5,1:2]

# Write a table with column names, without row names and with quotes.
write.table(data, "output-data_2.csv", append=FALSE, col.names=TRUE,
            row.names = FALSE, quote = TRUE, sep = ",")

# Display first five rows and first two columns
data_2 <- read.table("output-data_2.csv", header = TRUE, sep = ",")
data_2[1:5,1:2]

# Write a table with col names and row names, and with quotes.
write.table(data, "output-data_3.csv", append=FALSE, col.names=TRUE,
            row.names = TRUE, quote = TRUE, sep = ",")

# Display first five rows and first two columns
data_3 <- read.table("output-data_3.csv", header = TRUE, sep = ",")
data_3[1:5,1:2]

# Write a table without col names and row names, and without quotes.
write.table(data, "output-data_4.csv", append=FALSE, col.names=FALSE,
            row.names = FALSE, quote = FALSE, sep = ",")

# Display first five rows and first two columns
data_4 <- read.table("output-data_4.csv", header = TRUE, sep = ",")
data_4[1:5,1:2]

# Write a table without col names and row names, and without quotes of the first sample of the data file
write.table(data$X00314416.31db.41a3.8264.ef7bcd3279af, "output-data_5.csv", append=FALSE, col.names=FALSE,
            row.names = FALSE, quote = FALSE, sep = ",")

# Display first five rows since this file has only one column - one sample.
data_5 <- read.table("output-data_5.csv", header = TRUE, sep = ",")
head(data_5)

# Write a table without col names and row names, and without quotes of a list of the first and second samples of the
data file
write.table(list(data$X00314416.31db.41a3.8264.ef7bcd3279af,data$X008d0418.e5a5.43d0.bc99.77ae9a0d85cf),
"output-data_6.csv", append=FALSE, col.names=FALSE, row.names = FALSE, quote = FALSE, sep = ",")

# Display first five rows and first two columns
data_6 <- read.table("output-data_6.csv", header = TRUE, sep = ",")
data_6[1:5,1:2]
```

Checking TCGA data

- DEFINITION: Lists are arrays with mixed data types.
- Use the function read.table() to import data
  - This means that although the first column contains text type data, it is allowable to perform calculation with the data in column two.

```
85 ```{R}
86 clinical_data <- read.table("trgn599.clinical.tsv", header = TRUE, sep = "\t")
87
88 # Check number of rows of the dataset
89 nrow(clinical_data)
90
91 # Check number of columns of the dataset
92 ncol(clinical_data)
93
94 # Display the first row and all columns of the dataset
95 clinical_data[1,]
96
97 # display the first column and all rows of the dataset
98 clinical_data[,1]
99 ```
```

Summarize TCGA data

- The summary function in R provide the mean, median, 25th and 75th quartiles, min, max of the variables in a dataset:

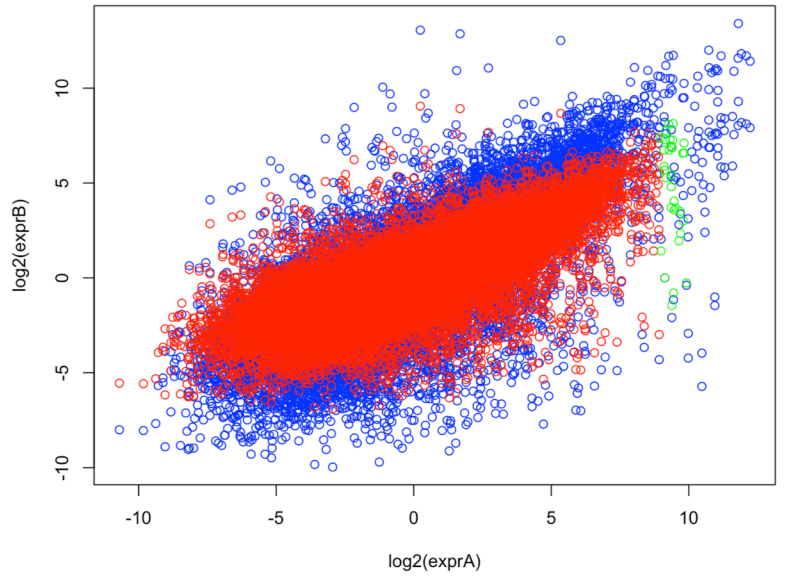
```
101 ## Summarize TCGA Data
102
103 ```{R}
104 # Display the descriptive statistics of the TCGA Data
105 summary(clinical_data)
106
107 # Display the headers of the TCGA Data
108 colnames(clinical_data)
109
110 # Display the descriptive statistics of the header: "days_to_last_follow_up"
111 summary(clinical_data$days_to_last_follow_up)
112
```

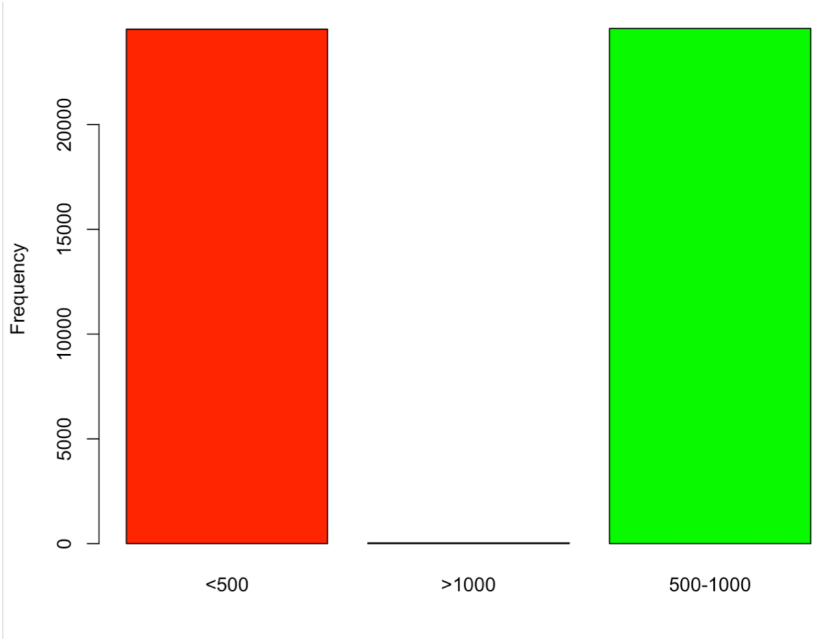
```
            Sample.Type  cigarettes_per_day     weight           height          gender    year_of_birth
Primary Tumor      :409   Min.   : 0.05479   Min.   : 0.05479   Min.   : 8.00   female:167   Min.   :1915
Solid Tissue Normal: 56   1st Qu.: 1.53425   1st Qu.: 1.53425   1st Qu.:29.50   male  :298   1st Qu.:1931
                          Median : 2.46575   Median : 2.46575   Median :40.00                Median :1938
                          Mean   : 2.67333   Mean   : 2.67333   Mean   :37.58                Mean   :1939
                          3rd Qu.: 3.28767   3rd Qu.: 3.28767   3rd Qu.:48.00                3rd Qu.:1946
                          Max.   :13.15068   Max.   :13.15068   Max.   :64.00                Max.   :1972
                          NA's   :117        NA's   :117        NA's   :290                  NA's   :12
```

Programming Structures

- R is a full-fledged programming language.
- It was designed to write complex programs.
- Here is a demonstration of using two fundamental functions: loops and conditionals
- The basic structure of a loop is: *for(index in vector) {…}*
- Below is an example where a *for* construct loops through all element indices of vector exprA, where the expression value fulfills a cerntain condition.

```
## Composite plots

```{R}
# Upload your file
data <- read.table("trgn599.rna.tsv", header = TRUE, sep = "\t")

# Check colnames
colnames(data)

# generate two vectors of first two columns of the dataset
exprA <- data$X00314416.31db.41a3.8264.ef7bcd3279af
exprB <- data$X008d0418.e5a5.43d0.bc99.77ae9a0d85cf


# Plot the log 2 of the first two samples of the dataset according with a specific gene expression ranges
plot(log2(exprA), log2(exprB), col= "blue")
for(i in which(exprA<500)) {points(log2(exprA[i]),log(exprB[i]), col="red")}
for(i in which(exprA>1000)) {points(log2(exprA[i]),log(exprB[i]), col="blue")}
for(i in which(exprA>=500 & exprA <=1000)) {points(log2(exprA[i]),log(exprB[i]), col="green")}
```
```



Programming Structures

- Write a complex one liner in order to count the number of genes with expression values of a particular range.
- Use the *if* function.

```
## Composite plots 2

```{R}
a = 0; b = 0; c = 0;
for(i in exprA) {
  if(i<500) {a=a+1};
  if(i>1000){b=b+1}else{c=c+1}
};
barplot(c(a,b,c),
        names=c("<500",">1000","500-1000"),
        col=c("red", "blue", "green"),
        yla="Frequency")
```
```



Exercise

- Exercise
  - Upload the file trgn599.clinical.tsv in your R program and display how many females and males are in the dataset.

Entering Data from Keyboard

- Answer:

```
## Exercise

```{R}
# Exercise
# Upload the file trgn599.clinical.tsv and display how many females and males are in the dataset.

summary(clinical_data$gender)

```
```

R Markdown

file:///Users/enriquevelazquez/Documents/R_working_directory/TRGN599_Week_1_Lecture_4.html

TRGN599_Week_1_Lecture_4

EIVV

Importing TCGA Data

```
# Check your current working directory
getwd()
```

```
## [1] "/Users/enriquevelazquez/Documents/R_working_directory"
```

```
# Set your working directory
setwd("/Users/enriquevelazquez/Documents/R_working_directory")

# Upload your file
data <- read.table("trgn599.rna.tsv", header = TRUE, sep = "\t")

# Display the names of the header
# since it is a large table display the first five names

allheaders <- names(data)
head(allheaders)
```

```
## [1] "gene"
## [2] "X00314416.31db.41a3.8264.ef7bcd3279af"
## [3] "X008d0418.e5a5.43d0.bc99.77ae9a0d85cf"
## [4] "X01aa3eb0.11d3.4248.90ec.f8bdc0c44b1a"
## [5] "X0209bb79.7f97.4f44.a645.b1fa9ed3b3ef"
## [6] "X02145fba.e5e7.41c3.a671.1e371aac8498"
```

```
# Plot the lo2 two of the first two samples of the dataset
plot(log2(data$X00314416.31db.41a3.8264.ef7bcd3279af), log2(data$X008d0418.e5a5.43d0.bc99.77ae9a0d85cf), col="blue")
```