Intro to Data Visualization and Statistics in R

Session #5

Genome Institute of Singapore

31st October 2019

Learning objectives for Session 5

Objectives:

Combine information from multiple sheets/files.

Learn the following concepts

1. Match using `%in%` and `which()`
2. Replacing strings using `gsub()`
3. List
4. The for() loop

Match using %in% and which()

```r
all_fruits <- c("Apple", "Banana", "Cherry",
                "Orange", "Rambutan")

red_fruits <- c("Cherry", "Rambutan", "Apple")

all_fruits %in% red_fruits     # Returns a logical vector
## [1]  TRUE FALSE  TRUE FALSE  TRUE

which(all_fruits %in% red_fruits)   # Position of matches
## [1] 1 3 5
```

Replacing strings using gsub()

You just realized that there green and red apples. Let's change the text to red apple to be more precise.

```r
gsub("Apple", "Red Apple", all_fruits)
# [1] "Apple" "Red Banana" "Cherry" "Orange" "Rambutan"
```

gsub() is like a flexible find-and-replace command.

List | Method 1 for list creation

```r
red_fruits    <- c("Cherry", "Rambutan", "Apple")
yellow_fruits <- c("Banana", "Mango")
green_fruits  <- c("Mango")

fruits <- list(red_fruits, yellow_fruits, green_fruits)
fruits
## [[1]]
## [1] "Cherry"   "Rambutan" "Apple"
##
## [[2]]
## [1] "Banana" "Mango"
##
## [[3]]
## [1] "Mango"
```

Note: this list does not have a name.

List | Method 2 for list creation

```
fruits <- list(NULL)
fruits[["red_fruits"]] <- c("Cherry", "Rambutan", "Apple")
fruits[["yellow_fruits"]] <- c("Banana", "Mango")
fruits[["green_fruits"]]  <- c("Mango")
fruits
## [[1]]
## NULL
##
## $red_fruits
## [1] "Cherry"   "Rambutan" "Apple"
##
## $yellow_fruits
## [1] "Banana" "Mango"
##
## $green_fruits
## [1] "Mango"
```

Note: This list has name but first element is empty. Solution later!

The for() loop

```
all_fruits
## [1] "Apple"    "Banana"   "Cherry"   "Orange"   "Rambut

for(fr in all_fruits){

  fr  <- tolower(fr)
  txt <- paste("I love", fr, "now!")
  print(txt)
  rm(txt)
}
## [1] "I love apple now!"
## [1] "I love banana now!"
## [1] "I love cherry now!"
## [1] "I love orange now!"
## [1] "I love rambutan now!"
```

"`fr`" is the variable name that gets incremented to next value when it reaches the end of the loop.

Exercise

Background

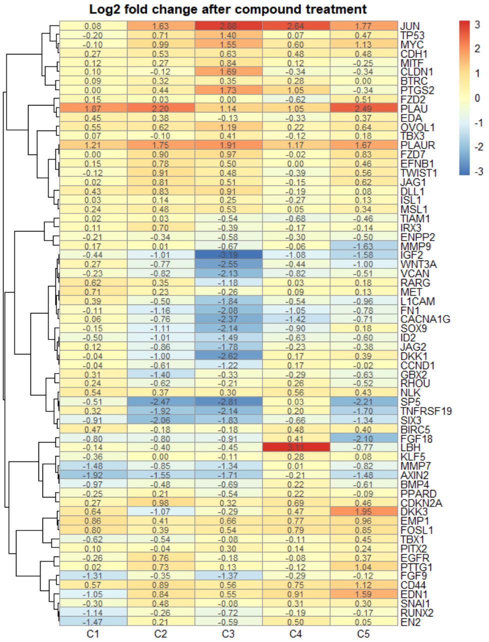
`RNAseq_treatment_effect.xlsx` contains the outputs from an experiment with the following groups:

- untreated cell lines (as controls)
- samples after treatment with C1 compound
- samples after treatment with C2 compound
- samples after treatment with C3 compound
- samples after treatment with C4 compound
- samples after treatment with C5 compound

Three biological replicates per group. 18 samples in total.

Your friendly neighbourhood bioinformation has compared each of the treated groups vs control. You also have a set of 66 genes that you like to investigate in more depth.

Objective | Generate heatmap of 66 genes



	C1	C2	C3	C4	C5
JUN	0.08	1.63	2.88	2.64	1.77
TP53	-0.20	0.71	1.40	0.07	0.47
MYC	-0.10	0.99	1.55	0.60	1.13
CDH1	0.27	0.53	0.63	0.48	0.48
MITF	0.12	0.27	0.84	0.12	-0.25
CLDN1	0.10	-0.12	1.69	-0.34	-0.34
BTRC	0.09	0.32	0.35	0.28	0.00
PTGS2	0.00	0.44	1.73	1.05	-0.34
FZD2	0.15	0.03	0.00	-0.62	0.51
PLAU	1.87	2.20	1.14	1.05	2.49
EDA	0.45	0.38	-0.13	-0.33	0.37
OVOL1	0.55	0.62	1.19	0.22	0.64
TBX3	0.07	-0.10	0.41	-0.12	0.18
PLAUR	1.21	1.75	1.91	1.17	1.67
FZD7	0.00	0.90	0.97	-0.02	0.83
EFNB1	0.15	0.78	0.50	0.00	0.46
TWIST1	-0.12	0.91	0.48	-0.39	0.56
JAG1	0.02	0.81	0.51	-0.15	0.62
DLL1	0.43	0.83	0.91	-0.19	0.08
ISL1	0.03	0.14	0.25	-0.27	0.13
MSL1	0.24	0.48	0.53	0.05	0.34
TIAM1	0.02	0.03	-0.54	-0.68	-0.46
IRX3	0.11	0.70	-0.39	-0.17	-0.14
ENPP2	-0.21	-0.34	-0.58	-0.30	-0.50
MMP9	0.17	0.01	-0.67	-0.06	-1.63
IGF2	-0.44	-1.01	-3.19	-1.08	-1.58
WNT3A	0.27	-0.77	-2.55	-0.44	-1.00
VCAN	-0.23	-0.82	-2.13	-0.82	-0.51
RARG	0.62	0.35	-1.18	0.03	0.18
MET	0.71	0.23	-0.26	0.09	0.13
L1CAM	0.39	-0.50	-1.84	-0.54	-0.96
FN1	-0.11	-1.16	-2.08	-1.05	-0.78
CACNA1G	0.06	-0.76	-2.37	-1.42	-0.71
SOX9	-0.15	-1.11	-2.14	-0.90	0.18
ID2	-0.50	-1.01	-1.49	-0.63	-0.60
JAG2	0.12	-0.86	-1.78	-0.23	-0.38
DKK1	-0.04	-1.00	-2.62	0.17	0.39
CCND1	-0.04	-0.61	-1.22	0.17	-0.02
GBX2	0.31	-1.40	-0.33	-0.29	-0.63
RHOU	0.24	-0.62	-0.21	0.26	-0.52
NLK	0.54	0.37	0.30	0.56	0.43
SP5	-0.51	-2.47	-2.81	0.03	-2.21
TNFRSF19	0.32	-1.92	-2.14	0.20	-1.70
SIX3	-0.91	-2.06	-1.83	-0.66	-1.34
BIRC5	0.47	-0.18	-0.18	0.48	0.40
FGF18	-0.80	-0.80	-0.91	0.41	-2.10
LBH	-0.14	-0.40	-0.45	3.11	-0.77
KLF5	-0.36	0.00	-0.11	0.28	0.08
MMP7	-1.48	-0.85	-1.34	0.01	-0.82
AXIN2	-1.92	-1.55	-1.71	-0.21	-1.48
BMP4	-0.97	-0.48	-0.69	0.22	-0.61
PPARD	-0.25	0.21	-0.54	0.22	-0.09
CDKN2A	0.27	0.98	0.32	0.69	0.46
DKK3	0.64	-1.07	-0.29	0.47	1.95
EMP1	0.86	0.41	0.66	0.77	0.96
FOSL1	0.80	0.39	0.54	0.79	0.85
TBX1	-0.62	-0.54	-0.08	-0.11	0.45
PITX2	0.10	-0.04	0.30	0.14	0.24
EGFR	-0.26	0.76	-0.18	-0.08	0.37
PTTG1	0.02	0.73	0.13	-0.12	1.04
FGF9	-1.31	-0.35	-1.37	-0.29	-0.12
CD44	0.57	0.89	0.56	0.75	1.12
EDN1	-1.05	0.84	0.55	0.91	1.59
SNAI1	-0.30	0.48	-0.08	0.31	0.30
RUNX2	-1.14	-0.26	-0.72	-0.19	-0.17
EN2	-1.47	0.21	-0.59	0.50	0.05

Step 1 | Genes of interest

1. Setup (setwd, packages, clear workspace)
2. Read in the the genes of interest as a vector called “`keep`”.
   Hint: pull()
3. Do you have 66 genes in your list?

Step 1 | Genes of interest

1. Setup (setwd, packages, clear workspace)
2. Read in the the genes of interest as a vector called "`keep`". Hint: pull()
3. Do you have 66 genes in your list?

```
setwd("C:/Users/aramasamy/Desktop/R_workshop")
pacman::p_load(tidyverse, readxl, pheatmap)
rm(list=ls())

fn <- "data/RNAseq_treatment_effect.xlsx"

## Identify the genes that we want to keep for plotting
keep <- read_excel(fn, sheet="My genes", col_names=F) %>% p
length(keep)
## [1] 66
```

Step 2A | Template processing one file

1. Read in the data from first compound.
2. Keep only the selected 66 genes.
3. Keep only the Gene and LFC columns.
4. Rename the “LFC” column as “C1”

Step 2A | Template processing one file

1. Read in the data from first compound.
2. Keep only the selected 66 genes.
3. Keep only the Gene and LFC columns.
4. Rename the "LFC" column as "C1"

```r
df <- read_excel(fn, sheet="C1") %>%
    filter(Gene %in% keep) %>%
    select(Gene, LFC)

colnames(df) <- gsub("LFC", "C1", colnames(df))
head(df, 3)
## # A tibble: 3 x 2
##   Gene      C1
##   <chr> <dbl>
## 1 AXIN2 -1.92
## 2 PLAU   1.87
## 3 RUNX2 -1.14
```

Step 2B | Generalize the template

1. Create a variable name `sh` and assign the value of "C1" to it.
2. Change all "C1" strings to the variable `sh`.

Step 2B | Generalize the template

1. Create a variable name `sh` and assign the value of "C1" to it.
2. Change all "C1" strings to the variable `sh`.

```r
sh <- "C1"

df <- read_excel(fn, sheet=sh) %>%
    filter(Gene %in% keep) %>%
    select(Gene, LFC)

colnames(df) <- gsub("LFC", sh, colnames(df))
head(df, 3)
## # A tibble: 3 x 2
##   Gene     C1
##   <chr> <dbl>
## 1 AXIN2 -1.92
## 2 PLAU   1.87
## 3 RUNX2 -1.14
```

Step 2C | Generate vector to loop over

1. Read in the sheet names of the Excel file.
2. Remove any unwanted sheet names from the vector.

Step 2C | Generate vector to loop over

1. Read in the sheet names of the Excel file.
2. Remove any unwanted sheet names from the vector.

```r
sheets <- excel_sheets(fn)
sheets
## [1] "C1"       "C2"       "C3"       "C4"       "C5"
## [7] "My genes"

sheets <- setdiff( sheets, c("README", "My genes") )
sheets
## [1] "C1" "C2" "C3" "C4" "C5"
```

Step 2D | Repeat with a `for()` loop

1. Generate an null list called “`store`”
2. Use the `sh` in a `for()` loop to read in multiple sheets
3. Save the data that you read in and processed into the “`store`” list

Step 2D | Repeat with a for() loop

1. Generate an null list called "store"
2. Use the sh in a for() loop to read in multiple sheets
3. Save the data that you read in and processed into the "store" list

```
store <- list(NULL)  ## A list to store each sheet

for(sh in sheets){

  df <- read_excel(fn, sheet=sh) %>%
    filter(Gene %in% keep) %>%
    select(Gene, LFC)

  colnames(df) <- gsub("LFC", sh, colnames(df))

  store[[ sh ]] <- df
  rm(df)
}
```

Step 2E | Remove the ghost element

It is an unintended feature in R to create an empty first element if you assign data into a list using names.

Here is one way to remove it:

```r
names(store)
## [1] ""   "C1" "C2" "C3" "C4" "C5"

store <- store[ sheets ]
names(store)
## [1] "C1" "C2" "C3" "C4" "C5"
```

Step 3 | Combine the data

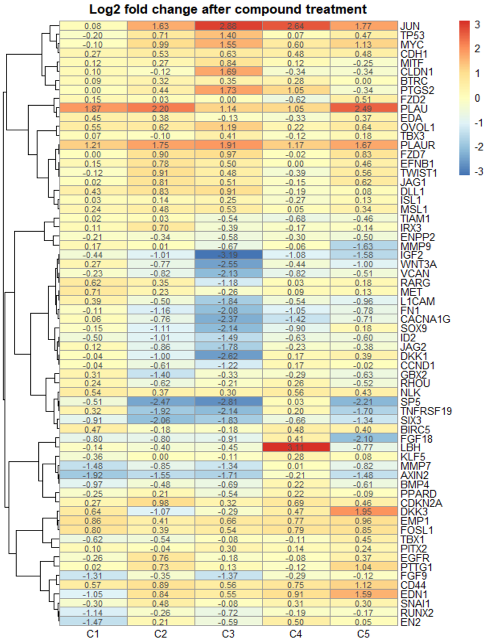
1. Combine it into one big dataframe.
2. Set the “Gene” column as the rownames.

Step 3 | Combine the data

1. Combine it into one big dataframe.
2. Set the "Gene" column as the rownames.

```r
comb <- plyr::join_all(store, type="full") %>%
  column_to_rownames("Gene")
```

Step 4 | Generate this heatmap



Step 4 | Generate this heatmap

```r
my_title <- "Log2 fold change after compound treatment"

pheatmap(comb,
         scale        = "none",
         angle_col    = 0,
         cluster_cols = FALSE,
         clustering_distance_rows = "correlation",
         main         = my_title,
         display_numbers = TRUE)
```