Base R: HackBio Wk 1

Qahhar

2023-10-30

The HackBio *DataScience4Life* contest required it's participant to complete a weekly task

"In this section, you are provided with a dataset and final figures. Your task is to reproduce the figures using the dataset alone". These words mark the exact instructions to be followed for this aspect of the contest

We have also been advised to "Use only base R functions. Do not use any library or package such as ggplot2 to solve the tasks"

In that light, here is my approach:

PROBLEM 1: FIG 1 A-E

Firstly:

Ensure you're working in your preferred directory

```
getwd()
```

```
## [1] "/cloud/project/Hackbio"
```

Secondly

We retrieve the dataset to be worked on, in said directory

```
url <- "https://raw.githubusercontent.com/HackBio-Internship/public_datasets/main/R/datasets/Contests/Vi

#Downloading the dataset, from the url , and using destfile to assign the file name
download.file(url, destfile = "fig_1_a_e.dat")
```

Thirdly:

Read the retrieved data into your environment. To read in this file, we utilize the read.table function, given a few arguments

- fig_1_a_e.dat: The name of our dataset
- header: This command when sets to true(T) keeps the header of any given txt file when being read in as a table or a data-frame
- sep: Every plain txt file to be read into R's console has a delimeter, The very thing that defines their structure, For our .dat file, the delimeter is a tab, thus the '' notation

```
fig_data <- read.table("fig_1_a_e.dat", header= T, sep = '\t')

head(fig_data)
```

```
##   depth        tech TSS_enrichment Unique_nr_frag_in_regions
## 1     5 10xmultiome       23.82273                    935.75
## 2     5       10xv1       25.89851                   1018.50
## 3     5      10xv11       20.49923                   1094.50
## 4     5     10xv11c       21.58124                   1770.25
## 5     5       10xv2       23.67305                   2466.75
## 6     5       ddseq       34.11716                   1498.50
##   X._unique_nr_frag_in_regions_in_cells median_cell_type_pred_score fc__B_cell
## 1                             0.2513718                   0.5454979   8.155354
## 2                             0.2208300                   0.6108661   7.986556
## 3                             0.2559485                   0.6161470  12.019779
## 4                             0.3817725                   0.6798223         NA
## 5                             0.5144501                   0.7028342  17.622997
## 6                             0.3005138                   0.5647128   5.312306
```

Dataset Previewing

Understanding the dimensions and structure of the dataset being worked with gives the analyst as good idea on what to expect as he/she goes down on the process

```r
dim(fig_data) # Returns the dimension of our tibble
```

```
## [1] 80  7
```

```r
colnames(fig_data) #This returns the no of columns in the given dataset
```

```
## [1] "depth"
## [2] "tech"
## [3] "TSS_enrichment"
## [4] "Unique_nr_frag_in_regions"
## [5] "X._unique_nr_frag_in_regions_in_cells"
## [6] "median_cell_type_pred_score"
## [7] "fc__B_cell"
```

```r
summary(fig_data) #Returns a descriptive summary of the entire dataset
```

```
##      depth           tech           TSS_enrichment   Unique_nr_frag_in_regions
##  Min.   : 5.00   Length:80          Min.   : 3.697   Min.   :  169.5
##  1st Qu.:13.75   Class :character   1st Qu.:21.307   1st Qu.: 1457.0
##  Median :22.50   Mode  :character   Median :22.646   Median : 3211.2
##  Mean   :22.50                      Mean   :23.327   Mean   : 3593.0
##  3rd Qu.:31.25                      3rd Qu.:27.607   3rd Qu.: 4852.4
##  Max.   :40.00                      Max.   :34.824   Max.   :10765.0
##
##  X._unique_nr_frag_in_regions_in_cells median_cell_type_pred_score
##  Min.   :0.04135                       Min.   :0.3855
##  1st Qu.:0.12403                       1st Qu.:0.6106
##  Median :0.18326                       Median :0.7234
##  Mean   :0.19684                       Mean   :0.6955
##  3rd Qu.:0.25594                       3rd Qu.:0.7959
##  Max.   :0.51445                       Max.   :0.8819
##
##    fc__B_cell
##  Min.   : 1.773
##  1st Qu.: 7.551
##  Median :12.809
```

```
##  Mean   :13.846
##  3rd Qu.:20.558
##  Max.   :31.550
##  NA's   :17
```

We get a statistical overview of the dataset.

- A majority of the columns are of number types with only one column/variable being a character/string type denoted by **chr**

In order to explore the 'tech' column, we can use "count" to give us a high end description of the columns values

```
length(unique(fig_data$tech))
```

```
## [1] 10
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
summarize(group_by(fig_data, tech), N = n())
```

```
## # A tibble: 10 x 2
##    tech             N
##    <chr>        <int>
##  1 10xmultiome      8
##  2 10xv1            8
##  3 10xv11           8
##  4 10xv11c          8
##  5 10xv2            8
##  6 ddseq            8
##  7 hydrop           8
##  8 mtscatac         8
##  9 mtscatacfacs     8
## 10 s3atac           8
```

From this, we find that the tech column has 10 unique values each of which are 8 in number, in line with the length of the dataset.

Plotting Figure 1

As stated previously, the task is to replicate template graphs. Here is the link for reference

Setting up a theme e.g color, font size and whatnot saves a ton of stress

```
tech_colors <- c("#00FFFF","#0066FF","#3300FF","#CC00FF","#FF0099","#FF0000","#FF9900","#CCFF00","#33FF
```

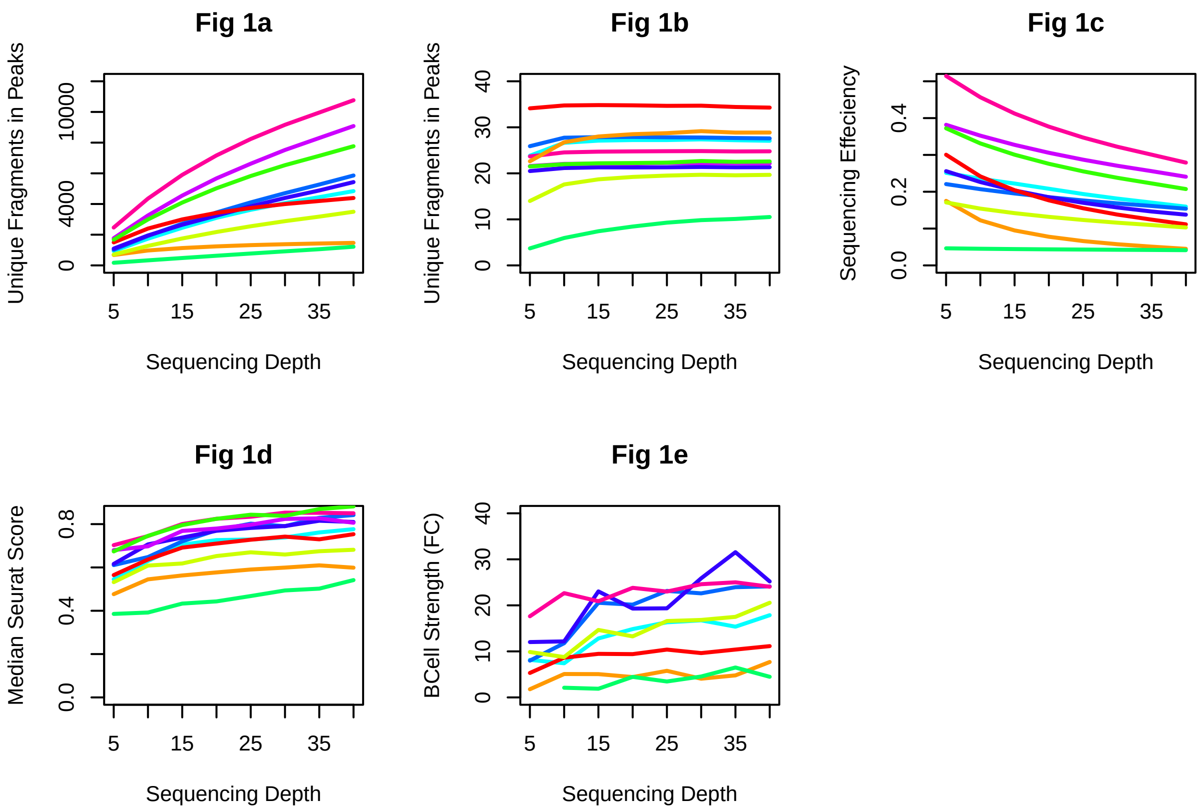
**We would also be dealing with the tech column alot in this problem, so it's best to store the unique values somewhere

```
technology <- c(unique(fig_data$tech)) # This extracts the unique values of tech in which we are intere
```

We would also create a layout for subsequent plots to fit into, and then have an output of a single panel file

```r
# Set up a 2x3 layout
par(mfrow = c(2, 3))


#Fig a
#setting up a blank plot
 plot(0,
     type = "n",
     xlim = c(5,40),
     ylim = c(0,12000),
     xlab= "Sequencing Depth",
     ylab = "Unique Fragments in Peaks",
     main = "Fig 1a")

for (i in 1:10) {
  filtered_data <- fig_data[fig_data$tech == technology[i],]
 # This command basically filters a dataset where tech would be equal to i index in technology

  #her telling progam to assign the values accordingly and set the colors individually
  lines(filtered_data$depth,filtered_data$Unique_nr_frag_in_regions, col = tech_colors[i], lwd = 2)
}

# Fig B
#setting up a blank plot
plot(0,
     type = "n",
     xlim = c(5,40),
     ylim = c(0,40),
     xlab= "Sequencing Depth",
     ylab = "Unique Fragments in Peaks",
     main = "Fig 1b")

for (i in 1:10) {
  filtered_data <- fig_data[fig_data$tech == technology[i],]
 # This command basically filters a dataset where tech would be equal to i index in technology

  #her telling progam to assign the values accordingly and set the colors individually
  lines(filtered_data$depth,filtered_data$TSS_enrichment, col = tech_colors[i], lwd = 2)
}

#Fig C
 plot(0,
     type = "n",
     xlim = c(5,40),
     ylim = c(0,0.5),
     xlab= "Sequencing Depth",
     ylab = "Sequencing Effeciency",
     main = "Fig 1c")

for (i in 1:10) {
  filtered_data <- fig_data[fig_data$tech == technology[i],]
 # This command basically filters the dataset where tech would be equal to i index in technology
```

```
  #her telling progam to assign the values accordingly and set the colors individually
  lines(filtered_data$depth,filtered_data$X._unique_nr_frag_in_regions_in_cells, col = tech_colors[i], 
}

# Fig D
#setting up a blank plot
 plot(0,
     type = "n",
     xlim = c(5,40),
     ylim = c(0,0.85),
     xlab= "Sequencing Depth",
     ylab = "Median Seurat Score",
     main = "Fig 1d")

for (i in 1:10) {
  filtered_data <- fig_data[fig_data$tech == technology[i],]
 # This command basically filters the dataset where tech would be equal to i index in technology

  #her telling progam to assign the values accordingly and set the colors individually
  lines(filtered_data$depth,filtered_data$median_cell_type_pred_score, col = tech_colors[i], lwd = 2)
}

# Fig E
 plot(0,
     type = "n",
     xlim = c(5,40),
     ylim = c(0,40),
     xlab= "Sequencing Depth",
     ylab = "BCell Strength (FC)",
     main = "Fig 1e")

for (i in 1:10) {
  filtered_data <- fig_data[fig_data$tech == technology[i],]
 # This command basically filters the dataset where tech would be equal to i index in technology

  #her telling progam to assign the values accordingly and set the colors individually
  lines(filtered_data$depth,filtered_data$fc__B_cell, col = tech_colors[i], lwd = 2)
}
```



Putting all the plots together, it is easier to draw comparisons from the relationships/trend of sequencing depth and other variables in the data frame

PROBLEM 2

These series of problems require datasets to be retrieved

```
#For Figure F
url <- "https://github.com/HackBio-Internship/public_datasets/raw/main/R/datasets/Contests/Viz/datasets/

# Read in the table
df_f <- read.table(url, header = TRUE, sep = "\t")

# For Figure G
url <- "https://github.com/HackBio-Internship/public_datasets/raw/main/R/datasets/Contests/Viz/datasets/

df_g <- read.table(url, header = TRUE, sep = "\t")

# For Figure H
url <- "https://raw.githubusercontent.com/HackBio-Internship/public_datasets/main/R/datasets/Contests/Vi

df_h <- read.table(url, header = TRUE, sep = "\t")

# For Figure J
url <- "https://raw.githubusercontent.com/HackBio-Internship/public_datasets/main/R/datasets/Contests/Vi

df_j <- read.table(url, header = TRUE, sep = "\t")
```

We would be using a different color theme, custom

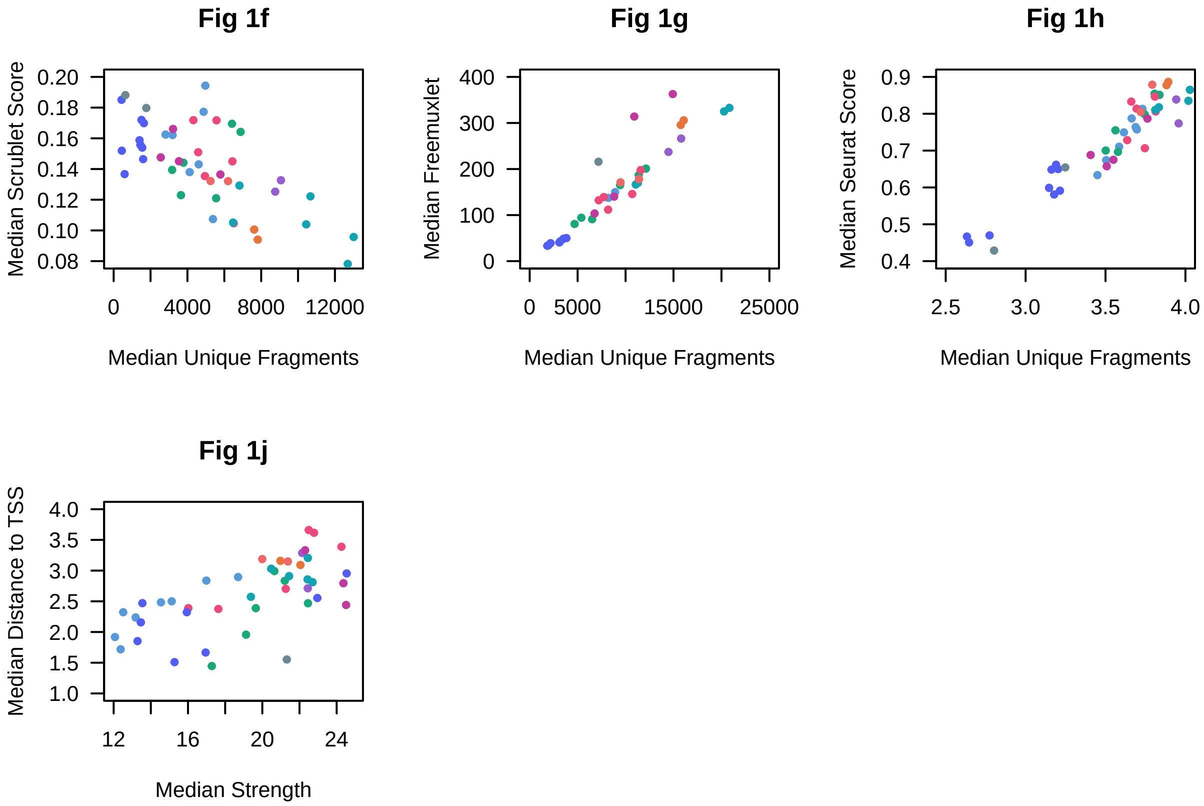
```
new_tech_colors <- c(lightblue = '#5899DA',
               lightorange = '#E8743B',
               lightgreen = "#19A979",
               lightred = "#ED4A7B",
               lightviolet = '#945ECF',
               lightturq = "#13A4B4",
               lightpurple = "#525DF4",
               lightmagenta = "#BF399E",
               grey = "#6C8893",
               lightamber = "#EE6868",
               deepblue = "#2F6497")
```

Plotting figure 2

```
# Set up a 2x3 layout
par(mfrow = c(2, 3))


# Fig F
technology_f <- c(unique(df_f$technology)) # This extracts the unique values of tech in which we are in


plot(0,
     type = "n",
     xlim = c(0, 13000),
     ylim = c(0.08, 0.20),
     xlab= "Median Unique Fragments",
     ylab = "Median Scrublet Score",
     main = "Fig 1f",
     las = 1)


for (i in 1:10) {
  filtered_f<- df_f[df_f$technology == technology_f[i],]
 # This command basically filters the dataset where tech would be equal to i index in technology

  points(filtered_f$Median_Unique_nr_frag_in_regions,
         filtered_f$Mean_scrublet_doublet_scores_fragments, col = new_tech_colors[i], pch = 20)
}

#Fig G

technology_g <- c(unique(df_g$technology)) # This extracts the unique values of tech in which we are in


 plot(0,
     type = "n",
     xlim = c(0, 25000),
     ylim = c(0, 400),
     xlab= "Median Unique Fragments",
     ylab = "Median Freemuxlet",
     main = "Fig 1g",
     las = 1)
```

```r
for (i in 1:10) {
  filtered_g<- na.omit(df_g[df_g$technology == technology_g[i],])
 # This command basically filters the dataset where tech would be equal to i index in technology

  points(filtered_g$Median_unique_nr_frag,
         filtered_g$fmx_delta_donor_llk, col = new_tech_colors[i], pch = 20)
}

# Figure H
technology_h <- c(unique(df_h$technology)) # This extracts the unique values of tech in which we are in


 plot(0,
     type = "n",
     xlim = c(2.5, 4.0),
     ylim = c(0.4, 0.9),
     xlab= "Median Unique Fragments",
     ylab = "Median Seurat Score",
     main = "Fig 1h",
     las = 1)


for (i in 1:10) {
  filtered_h<- df_h[df_h$technology == technology_h[i],]
 # This command basically filters the dataset where tech would be equal to i index in technology

  points(filtered_h$log_median_unique_nr_frag_in_regions,
         filtered_h$seurat_score, col = new_tech_colors[i], pch = 20)
}

# For Figure J
unq_tech <- unique(df_j$technology)

#blank plot
plot(1,
    type = "n",
    xlim = c(12 , 24.9),
    ylim = c(1.0,4.0),
    xlab = "Median Strength",
    ylab = "Median Distance to TSS",
    main = "Fig 1j",
    las=1)

# grid()

for (i in 1:10) {
 filtered_j <- df_j[df_j$technology == unq_tech[i],]

#Checking if the value in 'i'th position in technology matches to the same in the dataframe

# Telling program to assign the values accordingly and set the colors individually
```

```
points(filtered_j$top2kdars_median_dar_tss_dist/1000,filtered_j$top2kdars_median_dar_logfc,col = new_te

}
```



PROBLEM 3

What this hopes to do is to plot the sample id against variable "Seurat Score", subsetting each plot by the mode of "tech" applicable to the data

Let's first retrieve the data

```
url <- "https://raw.githubusercontent.com/HackBio-Internship/public_datasets/main/R/datasets/Contests/V

df_k <- read.table(url, header = TRUE, sep = "\t")
```

There seems to be an unnecessary data so it would be beneficial to drop it

```
new_data <- df_k[, -1]
head(new_data)
```

```
##   seurat_cell_type_pred_score  tech              sample_id
## 1                   0.7941011 ddseq BIO_ddseq_1.FIXEDCELLS
## 2                   0.5816272 ddseq BIO_ddseq_1.FIXEDCELLS
## 3                   0.6803693 ddseq BIO_ddseq_1.FIXEDCELLS
## 4                   1.0000000 ddseq BIO_ddseq_1.FIXEDCELLS
## 5                   0.9169055 ddseq BIO_ddseq_1.FIXEDCELLS
## 6                   0.6013571 ddseq BIO_ddseq_1.FIXEDCELLS
```

Basically what the code above states is that, pick all rows of the previous df_k dataframe and do not select the first column using the (-) and index (1)

Note: - is not minus

Plotting Figure 3

```
unq_tech <- unique(new_data$tech)
unq_tech
```

```
##  [1] "ddseq"        "mtscatacfacs" "10xmultiome"  "10xv11"       "10xv11c"
##  [6] "10xv2"        "hydrop"       "mtscatac"     "s3atac"       "10xv1"
```

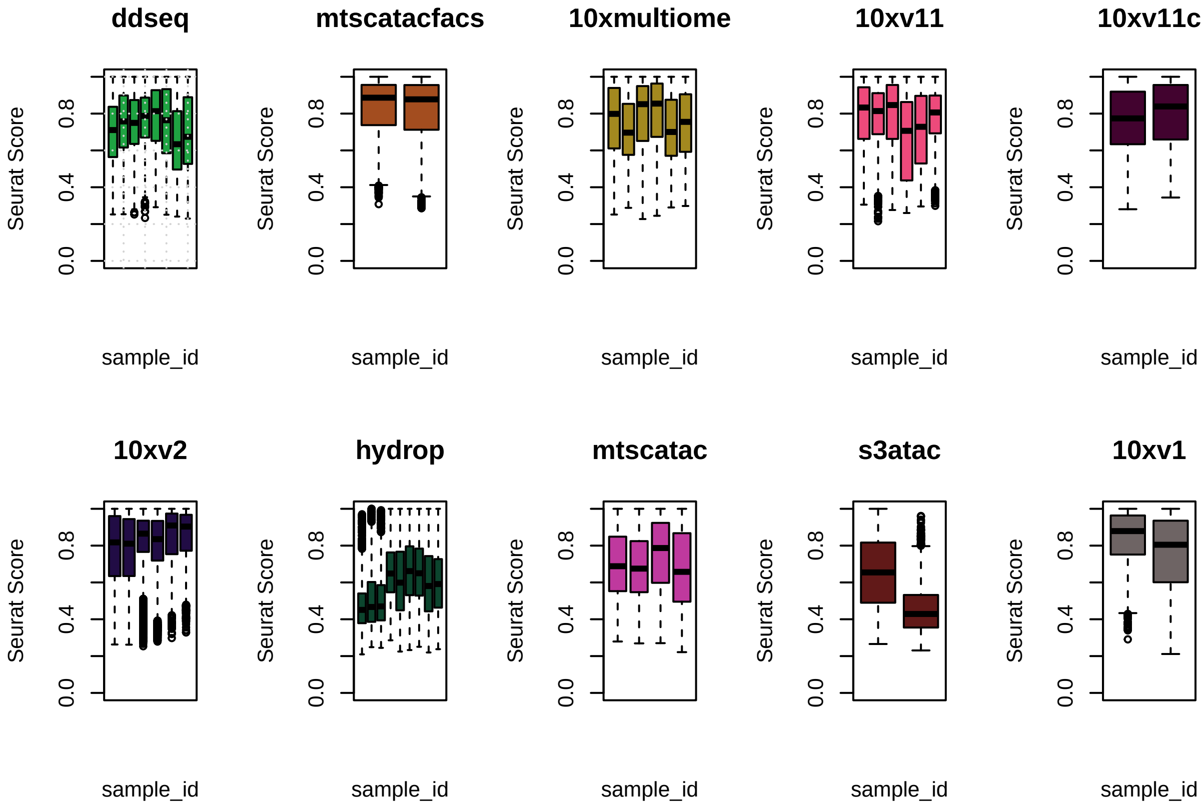
```
# Accessing the different "Data_for_" objects

for (tech in unq_tech) {
tech_data <- new_data[new_data$tech == tech, ] # Filter data for the current technology
assign(paste("Data_for_", tech, sep = ""), tech_data)
}

for (tech in unq_tech) {
  var_name <- paste("Data_for_", tech, sep = "")
  tech_data <- get(var_name)

  # Print the first few rows of each object
  cat("Data for", tech, ":\n")
  print(head(tech_data, n = 2))
}
```

```
## Data for ddseq :
##   seurat_cell_type_pred_score  tech              sample_id
## 1                   0.7941011 ddseq BIO_ddseq_1.FIXEDCELLS
## 2                   0.5816272 ddseq BIO_ddseq_1.FIXEDCELLS
## Data for mtscatacfacs :
##       seurat_cell_type_pred_score         tech                     sample_id
## 15926                   0.6849700 mtscatacfacs BRO_mtscatacfacs_1.FIXEDCELLS
## 15927                   0.6502055 mtscatacfacs BRO_mtscatacfacs_1.FIXEDCELLS
## Data for 10xmultiome :
##       seurat_cell_type_pred_score        tech                    sample_id
## 22651                   0.8009207 10xmultiome CNA_10xmultiome_1.FIXEDCELLS
## 22652                   0.5640099 10xmultiome CNA_10xmultiome_1.FIXEDCELLS
## Data for 10xv11 :
##       seurat_cell_type_pred_score   tech               sample_id
## 28898                   0.6166011 10xv11 CNA_10xv11_1.FIXEDCELLS
## 28899                   0.6945123 10xv11 CNA_10xv11_1.FIXEDCELLS
## Data for 10xv11c :
##       seurat_cell_type_pred_score    tech                sample_id
## 38432                   0.6947301 10xv11c CNA_10xv11c_1.FIXEDCELLS
## 38433                   0.6193778 10xv11c CNA_10xv11c_1.FIXEDCELLS
## Data for 10xv2 :
##       seurat_cell_type_pred_score  tech              sample_id
## 40264                   0.3841701 10xv2 CNA_10xv2_1.FIXEDCELLS
## 40265                   1.0000000 10xv2 CNA_10xv2_1.FIXEDCELLS
## Data for hydrop :
##       seurat_cell_type_pred_score   tech               sample_id
## 49705                   0.9306342 hydrop CNA_hydrop_1.FIXEDCELLS
## 49706                   0.4854532 hydrop CNA_hydrop_1.FIXEDCELLS
## Data for mtscatac :
##       seurat_cell_type_pred_score     tech                 sample_id
```

```
## 55005                   0.7185365 mtscatac CNA_mtscatac_1.FIXEDCELLS
## 55006                   0.8840439 mtscatac CNA_mtscatac_1.FIXEDCELLS
## Data for s3atac :
##       seurat_cell_type_pred_score   tech               sample_id
## 92035                   0.8681971 s3atac OHS_s3atac_1.FIXEDCELLS
## 92036                   0.8991045 s3atac OHS_s3atac_1.FIXEDCELLS
## Data for 10xv1 :
##        seurat_cell_type_pred_score  tech              sample_id
## 147459                   0.6939919 10xv1 VIB_10xv1_1.FIXEDCELLS
## 147460                   0.8984138 10xv1 VIB_10xv1_1.FIXEDCELLS
```

```r
par(mfrow = c(2,5)) #the first value is for rows and the second value is for columns
boxplot(Data_for_ddseq$seurat_cell_type_pred_score ~ Data_for_ddseq$sample_id,
main = "ddseq",
ylim = c(0.0, 1.0),
xlab = "sample_id",
ylab = "Seurat Score",
col = "#1fa342",
xaxt = "n") #xaxt = "n" to remove the ticks from the x-axis

grid()

boxplot (Data_for_mtscatacfacs$seurat_cell_type_pred_score ~
Data_for_mtscatacfacs$sample_id,
main = "mtscatacfacs",
ylim = c(0.0, 1.0),

xlab = "sample_id",
ylab = "Seurat Score",
col = "#a34d1f",
xaxt = "n")

boxplot (Data_for_10xmultiome$seurat_cell_type_pred_score ~
Data_for_10xmultiome$sample_id,
main = "10xmultiome",
ylim = c(0.0, 1.0),
xlab = "sample_id",
ylab = "Seurat Score",
col = "#a3891f",
xaxt = "n")

boxplot (Data_for_10xv11$seurat_cell_type_pred_score ~ Data_for_10xv11$sample_id,
main = "10xv11",
ylim = c(0.0, 1.0),
xlab = "sample_id",
ylab = "Seurat Score",
col = "#ED4A7B",
xaxt = "n")

boxplot (Data_for_10xv11c$seurat_cell_type_pred_score ~ Data_for_10xv11c$sample_id,
main = "10xv11c",
ylim = c(0.0, 1.0),
xlab = "sample_id",
ylab = "Seurat Score",
```

```
col = "#42032f",
xaxt = "n")

boxplot (Data_for_10xv2$seurat_cell_type_pred_score ~ Data_for_10xv2$sample_id,
main = "10xv2",

ylim = c(0.0, 1.0),
xlab = "sample_id",
ylab = "Seurat Score",
col = "#210c45",
xaxt = "n")

boxplot (Data_for_hydrop$seurat_cell_type_pred_score ~ Data_for_hydrop$sample_id,
main = "hydrop",
ylim = c(0.0, 1.0),
xlab = "sample_id",
ylab = "Seurat Score",
col = "#0c452e",
xaxt = "n")

boxplot (Data_for_mtscatac$seurat_cell_type_pred_score ~ Data_for_mtscatac$sample_id,
main = "mtscatac",
ylim = c(0.0, 1.0),
xlab = "sample_id",
ylab = "Seurat Score",
col = "#BF399E",
xaxt = "n")

boxplot (Data_for_s3atac$seurat_cell_type_pred_score ~ Data_for_s3atac$sample_id,
main = "s3atac",
ylim = c(0.0, 1.0),
xlab = "sample_id",
ylab = "Seurat Score",
col = "#611918",
xaxt = "n")

boxplot (Data_for_10xv1$seurat_cell_type_pred_score ~ Data_for_10xv1$sample_id,

main = "10xv1",
ylim = c(0.0, 1.0),
xlab = "sample_id",
ylab = "Seurat Score",
col = "#6e6464",
xaxt = "n")
```


ddseq
mtscatacfacs
10xmultiome
10xv11
10xv11c
10xv2
hydrop
mtscatac
s3atac
10xv1
Seurat Score
sample_id
0.0
0.4
0.8