# Project 4d

Margaret Dayhoff Team

This document serves as a comprehensive guide that encapsulates three distinct R scripts, each tailored to perform a specific function in the process of generating the bar plots depicted in Figure 1(b & c) of the research paper titled "A CRISPR-based base-editing screen for the functional assessment of BRCA1 variants". The paper can be accessed at the following link: https://www.nature.com/articles/s41388-019-0968-2#Sec11.

## 1. Conversion of Output Files:

The first script is designed to streamline the data processing workflow by converting all output files from the Cas-analyzer server, which are initially in .TXT format, into a more manageable .CSV format. This transformation facilitates easier data manipulation and analysis in the subsequent steps.

```r
# Set the directory path
directory_path <- "/Users/Hossam/Downloads/1B OUTPUT"

# Get a list of folders in the directory
folders <- list.dirs(directory_path, recursive = TRUE)

# Loop through each folder
for (folder in folders) {
  # Set the working directory to the current folder
  setwd(folder)

  # Get a list of .txt files in the current folder
  txt_files <- list.files(pattern = "\\.txt$")

  # Loop through each .txt file
  for (file in txt_files) {
    # Read all lines from the .txt file
    lines <- readLines(file)

    # Check if the file contains more than one line (excluding the header)
    if (length(lines) > 1) {
      # Read the first line from the .txt file
      first_line <- lines[1]

      # Read the rest of the .txt file as a data frame
      data <- read.table(text = lines[-1], header = FALSE, sep = "\t")

      # Set the header using the extracted first line
      colnames(data) <- unlist(strsplit(first_line, "\t"))

      # Extract the filename without extension
      filename <- gsub("\\.txt$", "", file)
      # Write the data to a .csv file
      write.csv(data, file = paste0(filename, ".csv"), row.names = FALSE)
    } else {
      # Print a message indicating that the file has only one line
      cat("Skipped file:", file, "as it contains only one line (excluding the header).\n")
    }
  }
}
```

## 2. Calculation of Required Parameters:

The second script is a robust computational tool that calculates all the necessary parameters required for plotting. It meticulously processes the data, computes the desired parameters, and stores these in a new .CSV file. The filename for this output file corresponds to the original file's name. Additionally, this script is equipped to calculate the average for each column in the file, providing a useful summary of the data.

```
# Load necessary library
library(dplyr)

# Set the working directory
setwd("/Users/Hossam/Downloads/1B OUTPUT/CCR5_14")

# Get a list of all CSV files in the directory
files <- list.files(pattern="*.csv")

# Initialize an empty data frame to hold all results
all_results <- data.frame()

# Loop over each file
for(file in files) {
  # Read the CSV file
  data <- read.csv(file)

  # Calculate the required values
  min_del <- min(data$Count[data$Type == "del"], na.rm = TRUE)
  max_del <- max(data$Count[data$Type == "del"], na.rm = TRUE)
  min_Ins <- min(data$Count[data$Type == "Ins"], na.rm = TRUE)
  max_Ins <- max(data$Count[data$Type == "Ins"], na.rm = TRUE)
  min_WT_Sub <- min(data$Count[data$Type == "WT or Sub"], na.rm = TRUE)
  max_WT_Sub <- max(data$Count[data$Type == "WT or Sub"], na.rm = TRUE)
  total_WT_Sub <- sum(data$Count[data$Type == "WT or Sub"], na.rm = TRUE)
  indels <- sum(data$Count[data$Type %in% c("Ins", "del")], na.rm = TRUE)
  total_count <- sum(data$Count, na.rm = TRUE)

  # Calculate the additional parameters
  Relative_del_max <- max_del / indels
  Relative_del_min <- min_del / indels
  Relative_Ins_max <- max_Ins / indels
  Relative_Ins_min <- min_Ins / indels
  Relative_Sub_max <- max_WT_Sub / total_WT_Sub
  Relative_Sub_min <- min_WT_Sub / total_WT_Sub
  Relative_WT_Sub <- total_WT_Sub / total_count
  Relative_Indels <- indels / total_count

  # Create a data frame to hold the results
  results <- data.frame(
    File = file,
    Relative_Del_Max = Relative_del_max,
    Relative_Del_Min = Relative_del_min,
    Relative_Ins_Max = Relative_Ins_max,
    Relative_Ins_Min = Relative_Ins_min,
    Relative_Sub_Max = Relative_Sub_max,
    Relative_Sub_Min = Relative_Sub_min,
    Relative_WT_Sub = Relative_WT_Sub,
    Relative_Indels = Relative_Indels
  )

  # Append the results to the all_results data frame
  all_results <- rbind(all_results, results)
}

# Calculate the average for each column (excluding the File column)
avg_results <- colMeans(all_results[, -1], na.rm = TRUE)

# Add the averages to the all_results data frame
all_results <- rbind(all_results, c("Average", avg_results))

# Write all results to a new CSV file
write.csv(all_results, file="CCR5_day14_results.csv")
```

## Manual Step - Data Collection:

Following the execution of the first two scripts, there is a manual step that involves gathering the plotting data from the generated CSV files. This data is then consolidated into a single CSV file named plotting.csv.

## 3. Plotting Bar Plots:

The third R script is the final piece of this process. It utilizes the data stored in plotting.csv to generate the desired bar plots, effectively visualizing the data in a manner consistent with Figure 1(b & c) of the referenced research paper.

### plotting fig 1.b

```
# Load the necessary libraries
library(ggplot2)
library(viridis)

# Set the working directory
setwd("/Users/Hossam/Downloads/1B OUTPUT")

# Read the data from the CSV file
data <- read.csv("plotting.csv", header = TRUE)
```

```r
# Reorder the factor levels of the 'Days' variable
data$Days <- factor(data$Days, levels = c("Day 0", "Day 7", "Day 14"))

# Generate the plot
ggplot(data, aes(x = Mutation,
                 y = Relative.Indels.Frequency,
                 ymin = Lower,
                 ymax = Upper,
                 fill = Days)) +
  geom_col(width = 0.55, position = position_dodge(0.6)) +
  geom_errorbar(width = 0.1, position = position_dodge(0.6)) +
  scale_fill_manual(values = c("274c77", "6096ba", "33415c")) +
  scale_y_continuous(expand = expansion(0), limits = c(0, 1.05)) +
  labs(y = "Relative Indels Frequency",
       x = NULL,
       fill = NULL,
       title = "Cell viability analysis of HAP1-
Cas9 cells transfected with two different gRNAs targeting BRCA1
using targeted deep sequencing. BRCA1 #1 and BRCA1 #2 indicate
each BRCA1-targeting gRNA, and the CCR5-targeting gRNA was
used as a negative control") +
  theme(plot.margin = unit(c(1, 1, 1, 1), "cm"),
        panel.background = element_blank(),
        plot.title = element_text(size = 20, face = "bold", hjust = 0.5, margin = margin(b = 15)),
        axis.line = element_line(color = "black"),
        axis.title = element_text(size = 20, color = "black", face = "bold"),
        axis.text = element_text(size = 20, color = "black"),
        axis.text.x = element_text(margin = margin(t = 10)),
        axis.text.y = element_text(size = 17),
        axis.title.y = element_text(margin = margin(r = 10)),
        axis.ticks.x = element_blank(),
        legend.position = c(0.98, 0.8),
        legend.text = element_text(size = 12),
        legend.margin = margin(t = 3, l = 3, r = 3, b = 3),
        legend.key = element_rect(color = NA, fill = NA)) +
  guides(fill = guide_legend(keywidth = 1.2, keyheight = 1.2, default.unit = "cm"))

# Save the plot
ggsave("fig_B.png", width = 10, height = 7, dpi = 300)
```

**The produced plot:**

**plotting fig 1.c**

```r
# Load the necessary libraries
library(ggplot2)
library(viridis)

# Set the working directory
setwd("/Users/Hossam/Downloads/1C OUTPUT")
```

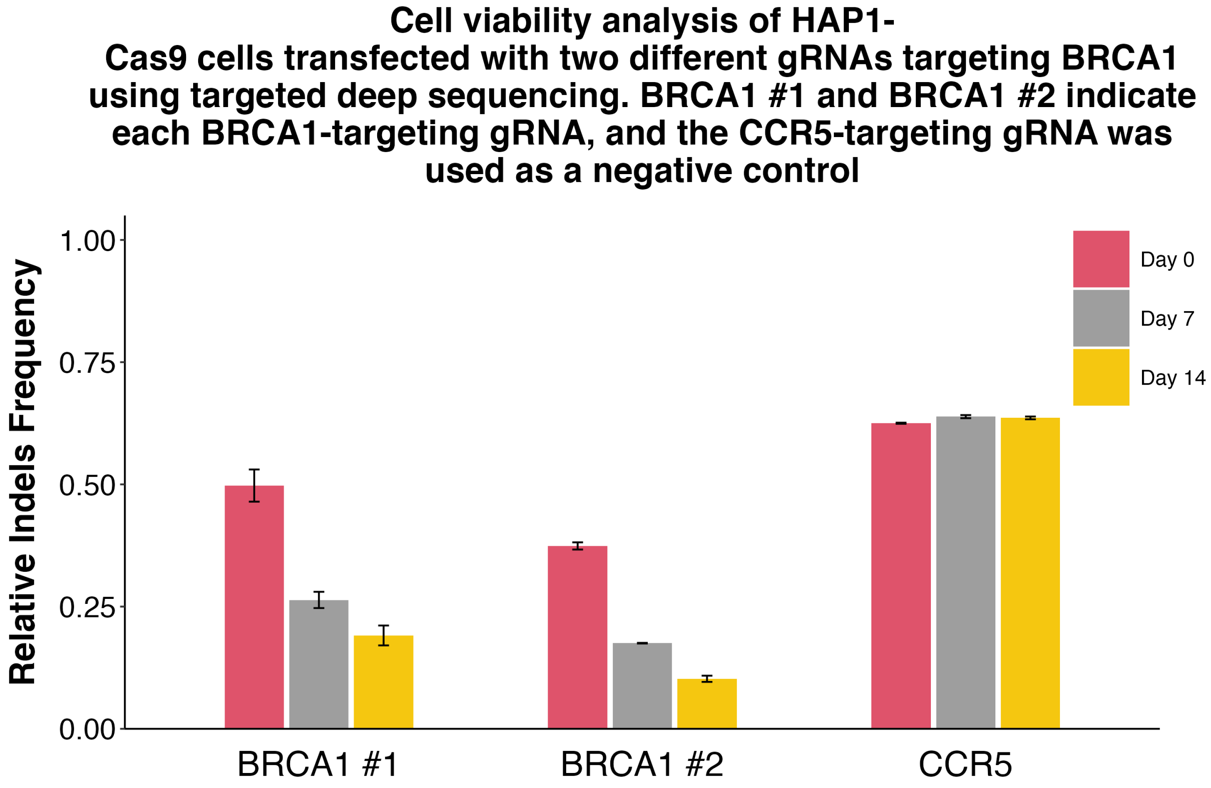

Figure 1: **Figure 1.b:** Cell viability analysis of HAP1- Cas9 cells transfected with two different gRNAs targeting BRCA1 using targeted deep sequencing. BRCA1 #1 and BRCA1 #2 indicate each BRCA1-targeting gRNA, and the CCR5-targeting gRNA was used as a negative control.

```
# Read the data from the CSV file
data <- read.csv("plotting.csv", header = TRUE)

# Reorder the factor levels of the 'Days' variable
data$Days <- factor(data$Days, levels = c("Day 0", "Day 7", "Day 21"))

# Generate the plot
ggplot(data, aes(x = Mutation,
                 y = Relative.Substitution.Frequency,
                 ymin = Lower,
                 ymax = Upper,
                 fill = Days)) +
  geom_col(width = 0.55, position = position_dodge(0.6)) +
  geom_errorbar(width = 0.1, position = position_dodge(0.6)) +
  scale_fill_manual(values = c("274c77", "6096ba", "33415c")) +
  scale_y_continuous(expand = expansion(0), limits = c(0, 1.05)) +
  labs(y = "Relative Substitution Frequency",
       x = NULL,
       fill = NULL,
       title = "Cell viability analysis of HAP1-BE3 cells
transfected with gRNAs targeting pathogenic mutations [c.81-1G>A
and c.191G>A (p.C64Y)] and a benign mutation [c.5252G>A
(p.R1751Q)] using targeted deep sequencing.") +
  theme(plot.margin = unit(c(1, 1, 1, 1), "cm"),
        panel.background = element_blank(),
        plot.title = element_text(size = 20, face = "bold", hjust = 0.5, margin = margin(b = 15)),
        axis.line = element_line(color = "black"),
        axis.title = element_text(size = 20, color = "black", face = "bold"),
        axis.text = element_text(size = 20, color = "black"),
        axis.text.x = element_text(margin = margin(t = 10)),
        axis.text.y = element_text(size = 17),
        axis.title.y = element_text(margin = margin(r = 10)),
        axis.ticks.x = element_blank(),
        legend.position = c(0.98, 0.8),
        legend.text = element_text(size = 12),
        legend.margin = margin(t = 3, l = 3, r = 3, b = 3),
        legend.key = element_rect(color = NA, fill = NA)) +
  guides(fill = guide_legend(keywidth = 1.2, keyheight = 1.2, default.unit = "cm"))

# Save the plot
ggsave("fig_C.png", width = 10, height = 7, dpi = 300)
```

**The produced plot:**

**REMEBER TO CHANGE THE DIRECTORIES AND THE NAMES OF THE OUTPUT FILES.**

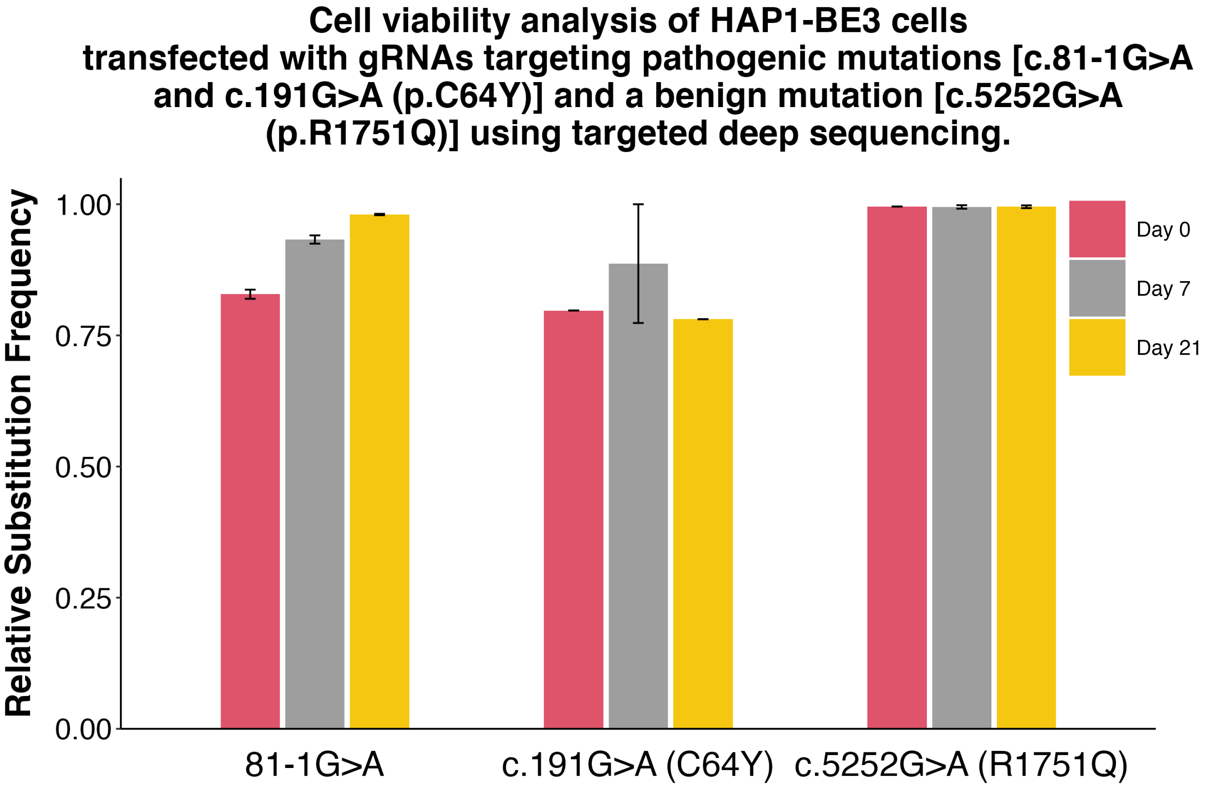

Figure 2: **Figure 1.c:** Cell viability analysis of HAP1-BE3 cells transfected with gRNAs targeting pathogenic mutations [c.81-1G>A and c.191G>A (p.C64Y)] and a benign mutation [c.5252G>A (p.R1751Q)] using targeted deep sequencing.