

Analyzing Human RNase P CRT Values for Handle and Swab Samples

SECTION 0: Required Packages and Files

The following packages must be installed and loaded into the R script prior to execution.

```
library(dplyr)
library(data.table)
library(ggplot2)
library(tidyr)
library(readr)
library(splitstackshape)
library(ggpubr)
```

The datasets required to do the RNase P Analysis can be downloaded from github (<https://github.com/bbi-lab>)

1. rnasep_crt_SCAN_SFS_dataset.csv
2. angry_swab_dataset.csv

SECTION 1: Creating Initial Dataframes

Remember to set your working directory to the appropriate path.

Loading the SCAN/SFS Dataset and the Handle Sample Dataset

Two datasets are loaded and read:

1. The raw dataset of all SCAN and SFS samples (both swab and handle) tested on the Open Array. This dataset omits failed and duplicated samples.

rnasep_crt_SCAN_SFS_dataset

2. The dataset of the samples where participants used the handle side for nasal collection

angry_swab_dataset

```
# load Angry (handle) Swab Data
dfAngry <- as.data.frame(read_csv("angry_swab_dataset.csv"))

# load Open Array Data
dfOA <- as.data.frame(read_csv("rnasep_crt_SCAN_SFS_dataset.csv"))
```

Two dataframes are created and saved in the Global Environment. 1. **dfOA** 2. **dfAngry**

In the next section, the dataframes for just handle samples and swab samples (minus handle samples) will be created.

SECTION 2: Creating Handle and Swab Dataframes

With dfAngry and dfOA, they can be merged together to obtain a handle dataframe. This dataframe will only include samples collected via handle.

```
# create a Handle dataframe
dfHandle <- inner_join(dfAngry, dfOA, by = "sample_id")
```

With dfAngry and dfOA, they can also be anti-joined to obtain a swab dataframe. This dataframe includes all swabs **minus** the handle samples.

```
# create a Swab dataframe for swabs across all racks
dfSwab <- anti_join(dfOA, dfAngry, by = "sample_id")
```

At the end of this section, two more dataframes are created and saved in the Global Environment.

1. **dfHandle**
2. **dfSwab**

In the next section, the dataframes will be prepared for data visualization.

SECTION 3: Preparation for Data Visualization

Obtaining Swab Samples within Same Latch Rack as Handle Samples

To prepare for graphing, swabs in the same latch rack as handle samples are pulled into a new dataframe.

To do so:

```
# pulls the name of the latch racks handle samples are in
angry_rack_list <- data.frame(rack_a_nickname = unique(dfHandle$rack_a_nickname.x))

# dataframe of all swabs in the same latch rack as handle samples
all_swabs_in_angry_racks <- merge(x = dfOA, y = angry_rack_list, by = "rack_a_nickname",
                                  all.x = FALSE)

# updated dataframe of swabs (minus handle) within same latch rack
dfSwab_within <- anti_join(all_swabs_in_angry_racks, dfAngry, by = "sample_id")
rm(angry_rack_list)
rm(all_swabs_in_angry_racks)
```

Obtaining RNase P CRT Values for Three Different Dataframes

The three dataframes of interest is:

1. only handle - `handle_crt`
2. all swabs - `swab_crt`
3. swabs within the same latch rack as handle - `swab_crt_within`

```
# data table for only handle and CRT values
handle_crt <- data.table(sample_id = dfHandle$sample_id, rack_id = dfHandle$rack_id,
                        handle = dfHandle$crt)
handle_crt$rack_id <- as.character(handle_crt$rack_id)

# data table for all swabs and CRT values
swab_crt <- data.table(sample_id = dfSwab$sample_id, rack_id = dfSwab$rack_id,
                      swab = dfSwab$crt)
swab_crt$rack_id <- as.character(swab_crt$rack_id)

#data table for only swabs within same latch rack as handle and CRT values
swab_crt_within <- data.table(sample_id = dfSwab_within$sample_id,
                             rack_id = dfSwab_within$rack_id, swab = dfSwab_within$crt)
swab_crt_within$rack_id <- as.character(swab_crt_within$rack_id)
```

The RNase P CRT values for handle versus swab can be melted and binded into one dataframe called `tidy_all_crt`.

```
# transforms handles' CRT value data
tidy_handle_crt <- melt(handle_crt, variable.name = "side_used",value.name = "crt")

# transform swabs' CRT value data
tidy_swab_crt <- melt(swab_crt, variable.name = "side_used",value.name = "crt")

# binds the two tidy data into one
tidy_all_crt <- bind_rows(tidy_handle_crt,tidy_swab_crt, .id = NULL)
```

SECTION 4: Data Analysis of Handle Versus Swab:

Calculating Mean with 95% Confidence Intervals for Handle Versus Swab

The mean along with a 95% confidence intervals can be calculated for handle samples and for swab samples.

```
# calculate mean RNase P CRTs
handle_mean <- mean(tidy_handle_crt$crt)
swab_mean <- mean(tidy_swab_crt$crt)

# calculate 95% confidence intervals
handle_ci = qnorm(0.975)*sd(tidy_handle_crt$crt)/sqrt(count(tidy_handle_crt)[[1,1]])
swab_ci <- qnorm(0.975)*sd(tidy_swab_crt$crt)/sqrt(tally(tidy_swab_crt)[[1,1]])
```

The mean for handle is **16.4** with 95% CI of **0.072**.

The mean for swab is **18.4** with 95% CI of **0.049**.

Calculating Standard Deviation for Handle Versus Swab

The standard deviation can be calculated for handle samples and for swab samples.

```
handle_error = sd(tidy_handle_crt$crt)/sqrt(tally(tidy_handle_crt))[[1,1]]
swab_error = sd(tidy_swab_crt$crt)/sqrt(tally(tidy_swab_crt))[[1,1]]
```

The 95% confidence interval for handle is **0.4**

The 95% confidence interval for swab is **0.03**

Calculating P-Value

The p-value to determine if the RNase P CRT values are significantly different for handle versus swab can be done via a two-tailed Welch's two-sample t-test.

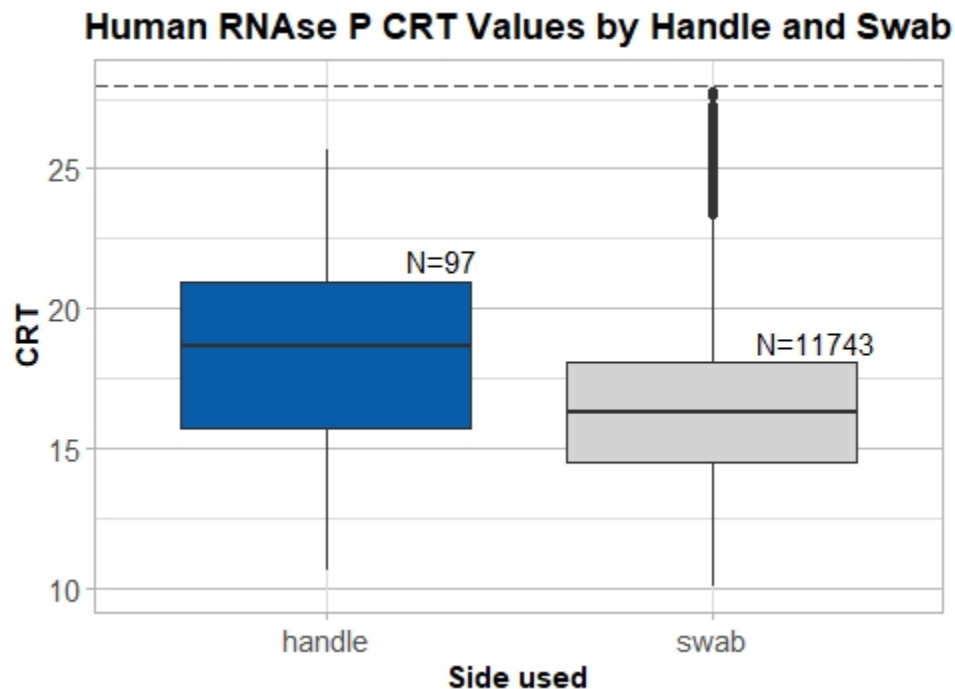
```
t.test(tidy_handle_crt$crt, tidy_swab_crt$crt)
```

The p-value is **< 0.01**.

SECTION 5: Creating a Box Plot

With the data analysis performed in Section 4, a box plot depicting the RNase P CRT values for handle versus swab can be generated.

```
boxplot <- tidy_all_crt %>%
  ggplot(aes(x=side_used, y=crt, fill=side_used))+
  geom_boxplot()+
  scale_fill_manual(values = c("swab" = "lightgrey",
                              "handle" = "#095DA8"))+
  theme_light()+
  theme(panel.background = element_rect(fill = "white"),
        panel.grid.major.y = element_line(size = 0.5,
                                           linetype = 'solid', color = "grey"),
        legend.position = "none",
        plot.title = element_text(face = "bold", hjust = 0.5),
        axis.title.x = element_text(face = "bold"),
        axis.title.y = element_text(face = "bold"),
        axis.text = element_text(size = 11))+
  ggtitle("Human RNase P CRT Values by Handle and Swab")+
  scale_y_continuous("CRT", breaks = c(10, 15, 20, 25, 30),
                    labels = c("10", "15", "20", "25", "30"))+
  geom_hline(yintercept=28, linetype = "longdash", color="#4D4D4D")+
  # annotate("rect", xmin= 1.26, xmax=1.74, ymin=27.5, ymax=28.5, fill="white")+
  # annotate("text", y=28, x=1.5, label="Limit of Detection", color = "black")+
  annotate("text", x=1.3, y=21.7, label = "N=97")+
  annotate("text", x=2.27, y=18.8, label = "N=11743")+
  xlab("Side used")+
  ylab("CRT")
print(boxplot)
```



SECTION 6: Creating a Dot Plot

In addition to the box plot, a dot plot displaying RNase P CRT values based on test date can be generated.

```
p <- ggplot()+
  geom_point(
    data=swab_crt_within,
    aes(x=as.character(rack_id),y=swab,color="swab"),
    size=1)+
  geom_point(
    data=handle_crt,
    aes(x=as.character(rack_id),y=handle,color="handle"),
    size=1)+
  theme_bw()+
  theme(panel.grid.minor.y = element_blank(),
        panel.border = element_rect(color="lightgrey"),
        plot.title = element_text(face = "bold", hjust = 0.5),
        axis.title.x = element_text(face = "bold"),
        axis.title.y = element_text(face = "bold"),
        axis.text.x = element_text(face = "bold",size = 7),
        axis.ticks = element_blank(),
        # plot.margin = unit(c(0.2,1,0.2,0.2), "in"),
        # legend.position = c(1.07,0.5),
        # legend.background = element_rect(color = "lightgrey"),
        axis.line.x = element_blank())+
  ylim(10,30)+
  scale_x_discrete(breaks = c("021","054","100","133","206"),
                  labels = c("Dec","Jan","Feb","Mar","Apr"))+
  ggtitle("Human RNase P CRT Values by Handle and Swab")+
```

```
# geom_text(aes(x="21",y=9.5, label="Dec 2019"))+
scale_color_manual(name = "Side Used", values = c("swab" = "lightgrey",
                                                    "handle" = "#095DA8"))+
xlab("Date")+
ylab("CRT")
print(p)
```

