PharmaSUG China 2023 - Paper CC-115

Generation of Geometric Mean Titer Plot in Immunogenicity from SAS and R

Qi Luo, Xiaofang Liu, Sinocelltech Ltd.

ABSTRACT

Generally, one of the primary objective of COVID-19 Vaccine trials is to evaluate the immunogenicity on Study Day 28 after vaccination of the study vaccines. The GMT (Geometric Mean Titer) plot is the effective and straightforward way to evaluate immunogenicity. We can use SAS GTL and ggplot2 package in R to create the GMT plot of neutralizing antibodies against SARS-CoV-2 variant and then compare the difference between the two methodology.

INTRODUCTION

This paper will discuss two ways to complete immunogenic analysis and provide the reader with details on how to represent the analysis data and demonstrate the plot increasing from baseline to Study Day 28 from SAS and R.

DATA DESCRIPTION AND GMT PLOT PRESENTATION

In this paper, we have generated available simulation data based on the data from the central laboratory. ADIS (Immunogenicity Analysis Dataset) generated per ADaM IG includes baseline and post baseline data regarding neutralization activities against variants.

The main variables in ADIS that will be used are listed as below.

Variable	Label
USUBJID	Unique Subject Identifier
IPPSFL	Immunogenicity Per-Protocol Set Flag
TRTAN	Actual Treatment (N)
COHORT	Cohort
PARAMCD	Parameter Code
ANL01FL	Analysis Record Flag 01
AVISITN	Analysis Visit (N)
AVAL	Analysis Value
BASE	Baseline Value

Table 1. Variable Description

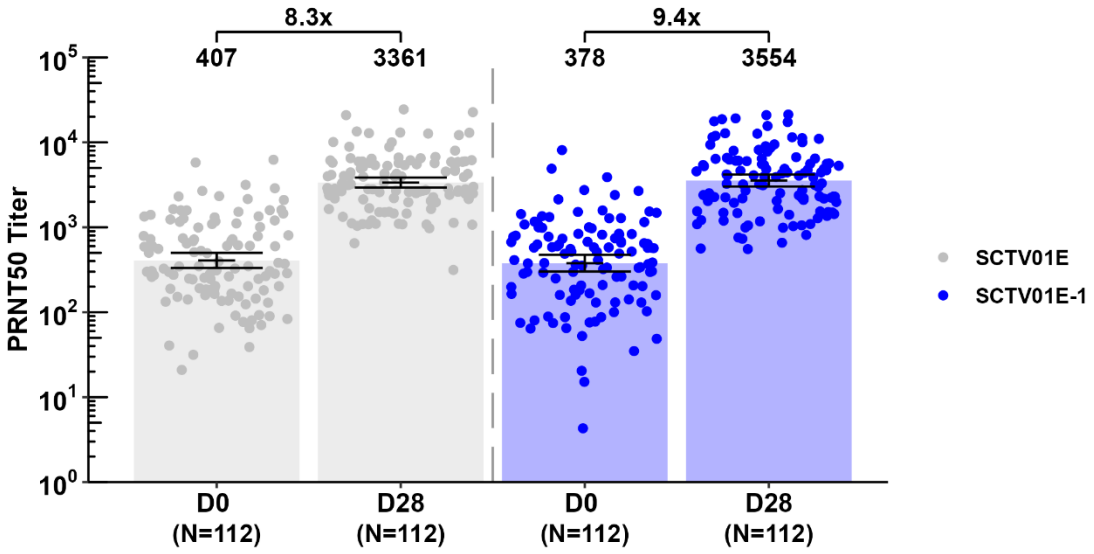

Figure 1. GMT Plot

As is shown to us the above plot, GMT of neutralizing antibody responses were measured using 50% plaque reduction neutralization test (PRNT50). Bars show the GMC (Geometric Mean Concentration) and GMT with 95% CIs at Day 0 and Day 28. Dots represent the values for individual participants. The Day 28 GMT of the neutralizing antibodies to variant were 3361 and 3554 with corresponding fold increase of 8.3 and 9.4.

SAS METHODOLOGY

SAS procedures are widely used to produce statistical results and graphs for clinical programmers.

COMPUTE GEOMETRIC MEAN BY PROC MEANS

Apply the common log to the AVAL and then get the new variable LOGAVAL. The MEANS procedure can help us to obtain the mean and 95% confidence interval. Furthermore, power of 10 will be used to obtain the final geometric mean and 95% CI:

```
proc means data=adis1 noprint;
  by trtan tptn avisitn;
  var logaval;
  output out=mean1 n=n mean=mean lclm=lclm uclm=uclm;
run;
```

COMPUTE GEOMETRIC MEAN BY PROC SURVEYMEANS

There is another SAS procedure to calculate the geometric mean and you don’t need to use log transformations:

```
ods output GeometricMeans=gmean;
proc surveymeans data=adis1 geomean gmclm;
  by trtan tptn avisitn;
  var aval;
run;
```

ADD ANNOTATION FOR GRAPH

Using annotation to customizing SAS graphs is necessary when you need to demonstrate clearly. Annotation dataset is recommended to generate batch annotations and each observation in dataset represents a command to draw an element:

```
data anno1;
  set mean1;
  length label $100 y1space x1space $20 textweight $10;
    where meanc ne '';
    retain function 'text' y1space 'graphpercent' x1space
          'datavalue' anchor 'Center' textcolor 'Black' textsize 10;
    width=200; textweight='Bold'; x1=tptn;
    y1=85;label=meanc; textcolor='black'; output;
    y1=4; label="(N="||strip(put(n, best.))||")"; output;
    y1=8; label="D"||strip(put(avisitn, best.)); output;
run;
```

DEFINE SAS GTL (GRAPH TEMPLATE LANGUAGE) TEMPLATE

HIGHLOWPLOT creates a vertical bar based on geometric mean concentration for each treatment group and visit. TYPE option should be set to “bar”, otherwise it will present a line. SCATTERPLOT creates scatter dots and it’s crucial to set the JITTER option if you want to jitter data markers. Another SCATTERPLOT creates dots with zero size in order to add the error bars:

```
highlowplot x=tptn low=low high=high /name='bar' group=backgroup
    display=(fill) fillattrs=(transparency=0.8) barwidth=0.6
    intervalbarwidth=60 groupdisplay=cluster
    clusterwidth=0.9 type=bar;
scatterplot x=tptn y=aval / name='scatter' markerattrs=( size=3
    symbol=circleFilled ) group=backgroup jitter=auto;
scatterplot x=tptn y=means / yerrorupper=yerru yerrorlower=yerrl
    markerattrs=(size=0) errorbarattrs=(color=CX000000
    thickness=1.5px ) errorbarcapscale=2;
```

R METHODOLOGY

As a clinical programmer, I have been more familiar with using SAS. So how can I transform my identity as a R programmer to complete the analysis? Let's demonstrate step by step.

PREPARATION

The following packages from R should be loaded in advance:

```
library(plyr)
library(tidyverse)
library(haven)
library(scales)
```

“read_sas” function in the haven package is to read a sas7bdat dataset if you have generated the ADaM.ADIS in advance by SAS:

```
adis <- read_sas("adis.sas7bdat")
```

HOW TO COMPUTE GEOMETRIC MEAN IN R

Then tidyverse is a set of packages and a powerful tool for every R user. “filter” and “mutate” function in dplyr package are used to subset a data frame and derive new variables. “summarise” and “group_by”

are also used to calculate the descriptive statistics by groups. “rbind.fill” function in plyr package is used to combine datasets as request. All of these functions come together to help us handle data gracefully:

```
adis %>%
  filter(IPPSFL == "Y", PARAMCD == "SAR2NAB", ANL01FL == "Y", COHORT ==
"Cohort 2") %>%
  mutate(TPTN = case_when(AVISITN == 0 ~ 1+(TRTAN-1)*2,
                          AVISITN %in% c(28, 42) ~ 2+(TRTAN-1)*2),
         LOGAVAL = log10(AVAL),
         LOGFOLD = log10(AVAL/BASE)
           )

adis1 %>%
  group_by(TRTAN, TPTN, AVISITN) %>%
  select(TRTAN, TPTN, AVISITN, LOGAVAL, LOGFOLD) %>%
  dplyr::summarise(n = n(),
            mean1 = mean(LOGAVAL),
            mean2 = mean(LOGFOLD),
            sd1 = sd(LOGAVAL),
            sd2 = sd(LOGFOLD)) %>%
  mutate(se = sd1/sqrt(n),
         lclm = mean1 - 1.96 * se,
         uclm = mean1 + 1.96 * se,
         yerrl = 10**lclm,
         yerru = 10**uclm,
         means = 10**mean1,
         foldmean = ifelse(AVISITN != 0,10**mean2, NA),
         nlabel = str_c("(N=",as.character(n),")"),
         foldlabel = str_c(as.character(round(foldmean,1)),"x"),
         ymin = 10^5*1.5,
         ymax = 10^5*2)
```

HOW TO ADD ANNOTATION FOR GRAPH IN R

The ggplot2 package which is known to us is used to realize data visualization. “geom_text” is the general function to add a text annotation to a plot. It’s very convenient to setup the font, size, bold and so on:

```
geom_text(aes(x = TPTN, y = 10^5,
              label = as.character(round(means))),
              na.rm = TRUE, family = "WRYH",
              fontface = "bold", size = 4)
```

HOW TO DEFINE GRAPH IN R

Compared with SAS, “geom_col”, “geom_jitter” and “geom_errorbar” are used to create the vertical bar, scatter dots and error bars respectively:

```
geom_col(aes(x = TPTN, y = means, fill = factor(TRTAN)),
    alpha = 0.3, na.rm = TRUE)+
geom_jitter(aes(x = TPTN, y = AVAL, color = factor(TRTAN)),
    na.rm = TRUE)+
geom_errorbar(aes(x = TPTN, ymin = yerrl, ymax = yerru),
    width = .5, size = 0.5)
```

CONCLUSION

Obtaining GMT plot from SAS and R is not difficult but is quite different. In SAS you use the GTL and annotation. In R you can use the powerful ggplot2 package. I hope every clinical programmer can have more effective tools to work with high quality and efficiency and I think the results are worth it.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Qi Luo
Clinical Programming
Sinocelltech Ltd.
qi_luo@sinocelltech.com

Xiaofang Liu
Clinical Programming
Sinocelltech Ltd.
xiaofang_liu@sinocelltech.com