Visualizing and standardizing data

2025-05-22

Introduction

The purpose of this document is to show how specific samples are determined to seropositive or seronegative using external information about a particular pathogen and antigen (e.g., an internationally recognized threshold of protective immunity). This document then describes how to aggregate serostatus information about each sample to calculate a population level seroprevalence. Upon completing this lab you should be able to:

- Read in control data if available
- Visualize MFI distributions on the appropriate scale
- Calculate serostatus for each sample using a predetermined cutoff
- Calculate a population seroprevalence.

General housekeeping

Before we start, let's navigate to the appropriate working directory. You can accomplish this by navigating to the "Session" tab of Rstudio, and choosing "Set Working Directory" –> "Choose Directory" and using your file browser to navigate to the Data folder within the seroanalytics_workshop folder. Alternatively, you can modify the code below as appropriate for your files to get to the Data folder in the seroanalytics_workshop folder.

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message=FALSE)

# set my_path to be the working directory location of where the *seroanalytics_workshop* folder is stor
my_path <- "/Users/soniahegde/Library/CloudStorage/OneDrive-JohnsHopkins/seroanalytics_workshop"

source(paste(my_path, "Source/utils.R", sep="/"))
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
##
## 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
```

```
## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'MASS' was built under R version 4.1.2

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

## Warning: package 'mclust' was built under R version 4.1.2

## Package 'mclust' version 6.0.0
## Type 'citation("mclust")' for citing this R package in publications.
```

Reading in data

Note this is the same procedure as in lab 3.

```
control_data <- read.csv(paste(my_path, "Data/simulated_control_long_training_data.csv", sep="/"))
sample_data <- read.csv(paste(my_path, "Data/simulated_sample_wide_training_data.csv", sep="/"))

#this converts sample_data from a wide to a long dataframe. edit the column names to your data.
sample_long <- reshape(
  sample_data,
  varying = setdiff(names(sample_data), c("id", "age", "sex")),
  v.names = "mfi",
  timevar = "antigen",
  times = setdiff(names(sample_data), c("id", "age", "sex")),
  idvar = "id",
  direction = "long"
)
rownames(sample_long) <- NULL
```

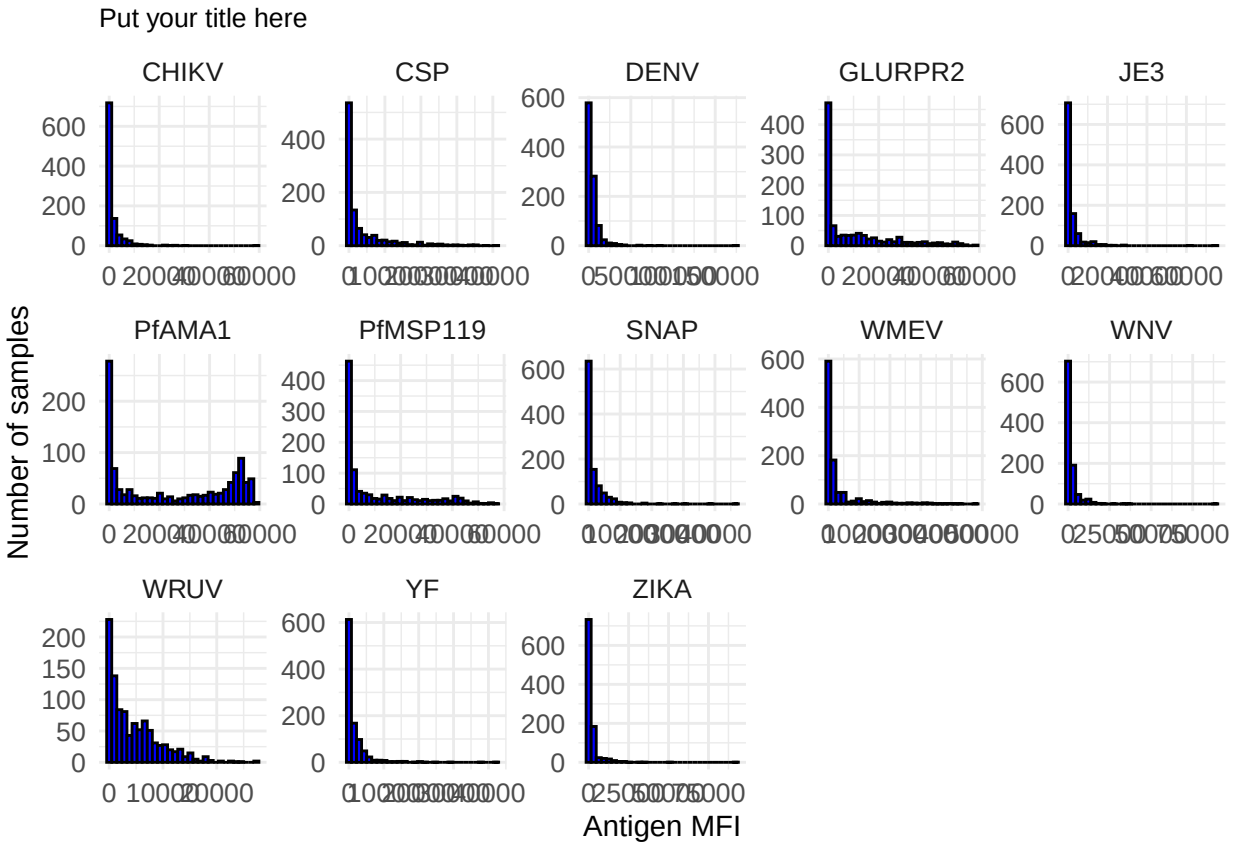
Visualizing your controls

General visalization

1. Adjust the code below to make a histogram of sample MFI values in an untransformed and log scale.

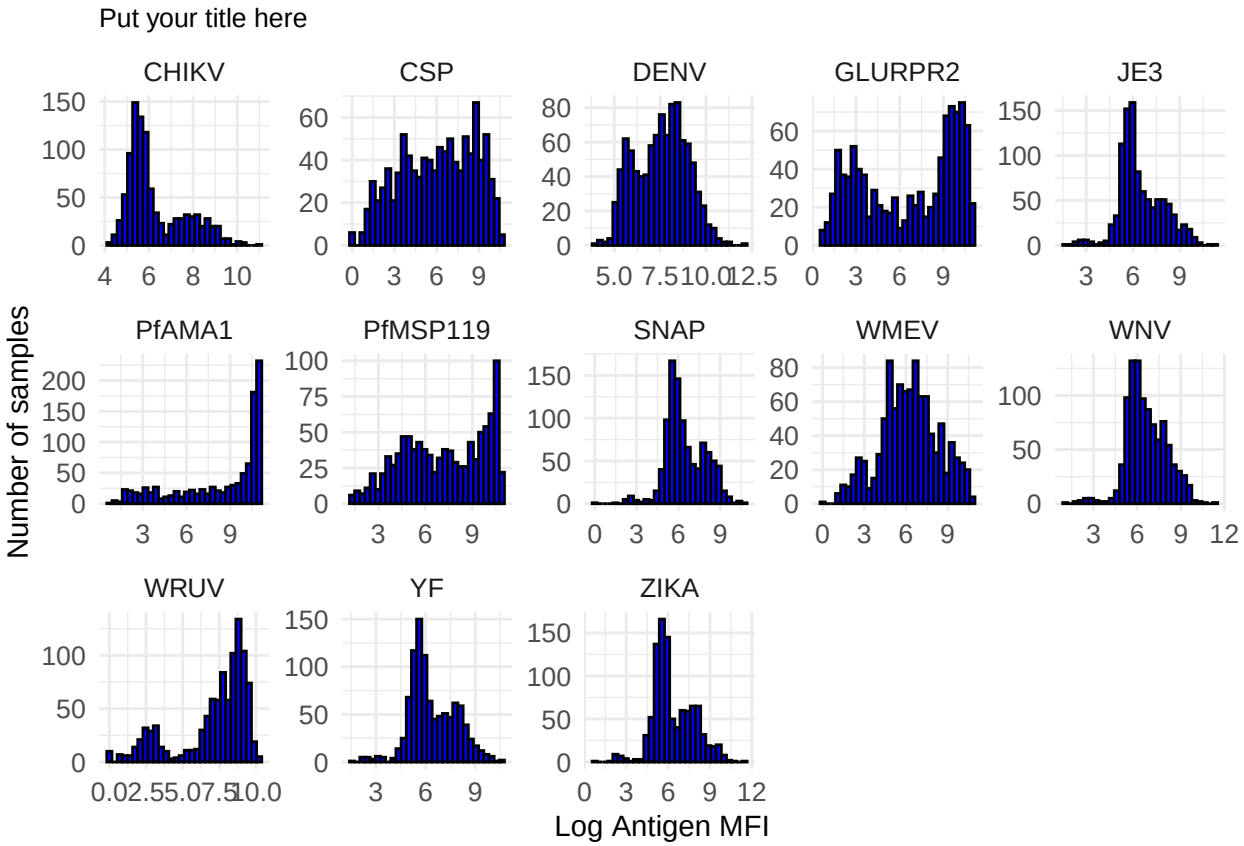
a. Consider how many bins to use (edit bins = 30 to see what data looks like with different numbers of bins).
b. Describe the distribution (untransformed and log scale).
c. Are there any outliers or anything unusual about your data?

```
#natural scale
faceted_natural_scale <- ggplot(sample_long, aes(x = mfi)) +
  geom_histogram(bins = 30, color = "black", fill = "blue") +
  facet_wrap(~ antigen, scales = "free", ncol = 5) +  # <- Set 5 columns per row
  labs(
    title = "Put your title here",
    x = "Antigen MFI",
    y = "Number of samples"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),  # Smaller font for facet labels
    axis.text = element_text(size = 10),  # Smaller font for axis text
    plot.title = element_text(size = 10)  # Smaller font for the plot title
  )
faceted_natural_scale
```




```
#log scale
faceted_log_scale <- ggplot(sample_long, aes(x = log(mfi))) +
  geom_histogram(bins = 30, color = "black", fill = "blue") +
```

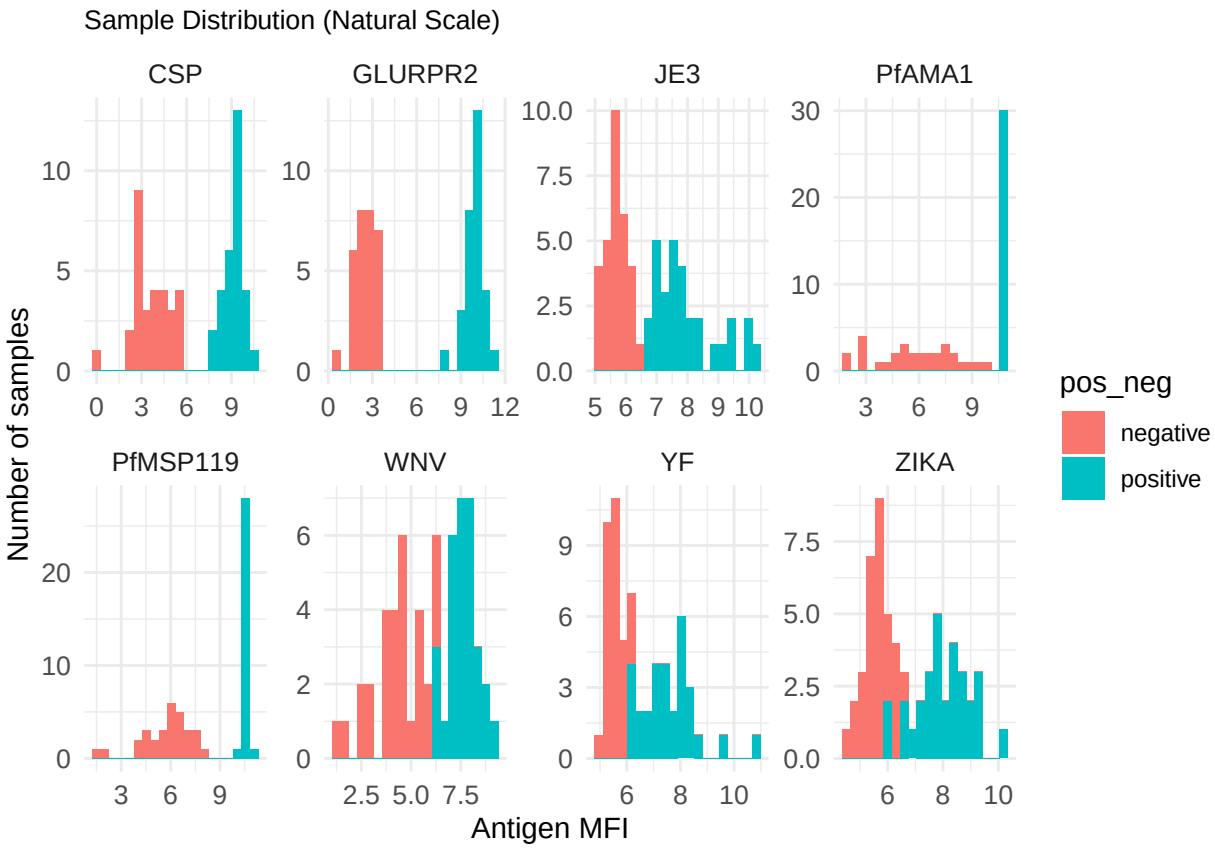
```
  facet_wrap(~ antigen, scales = "free", ncol = 5) +  # <- Set 5 columns per row
  labs(
    title = "Put your title here",
    x = "Log Antigen MFI",
    y = "Number of samples"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),  # Smaller font for facet labels
    axis.text = element_text(size = 10),  # Smaller font for axis text
    plot.title = element_text(size = 10)  # Smaller font for the plot title
  )
faceted_log_scale
```



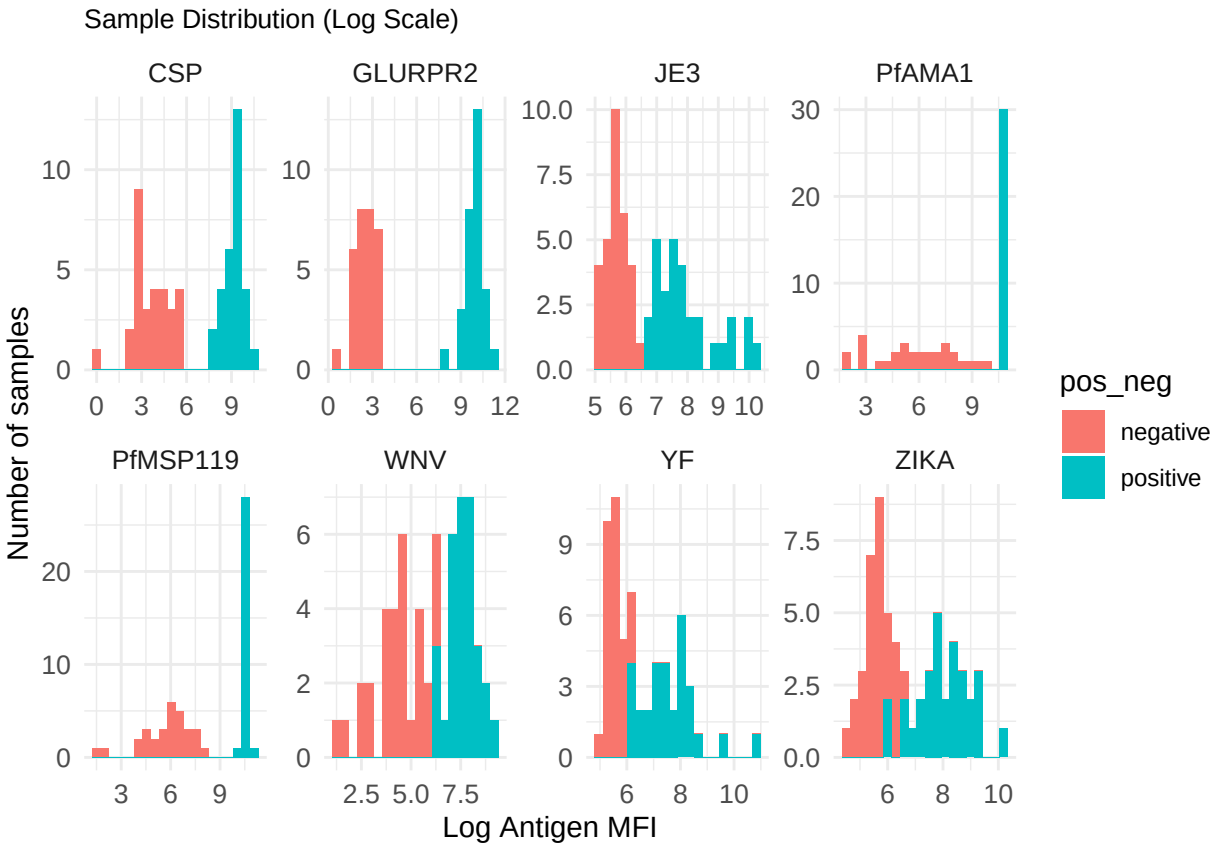
2. Make a histogram of control MFI values. Color the histogram by positive and negative controls. Is there overlap between your positive and negative controls?

```
  neg_controls_natural_scale <- ggplot(control_data, aes(x = log(mfi), fill = pos_neg)) +
    geom_histogram(bins = 20) +
    facet_wrap(~ antigen, scales = "free", ncol = 4) +
    labs(
      title = "Sample Distribution (Natural Scale)",
      x = "Antigen MFI",
      y = "Number of samples"
    ) +
```

```
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),
    axis.text = element_text(size = 10),
    plot.title = element_text(size = 10)
  )
neg_controls_natural_scale
```




```
neg_controls_log_scale <- ggplot(control_data, aes(x = log(mfi), fill = pos_neg)) +
  geom_histogram(bins = 20) +
  facet_wrap(~ antigen, scales = "free", ncol = 4) +  # <- Set 5 columns per row
  labs(
    title = "Sample Distribution (Log Scale)",
    x = "Log Antigen MFI",
    y = "Number of samples"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 10),  # Smaller font for facet labels
    axis.text = element_text(size = 10),  # Smaller font for axis text
    plot.title = element_text(size = 10)  # Smaller font for the plot title
  )
neg_controls_log_scale
```



Establishing and applying a cutoff

3. For an antigen with an established standard cutoff (eg. based on a correlate of protection) like Measles (wmev antigen) the cutoff value is XX IU/mL. Therefore, those samples above XX are positive and protected from future measles infection, and those samples below XX are negative and not protected from future measles infection.

a. For this antigen, plot the standard curve of IU/mL dilution by MFI and add a vertical line showing the cutoff. Add a rug plot to show where the observed values fall on the y-axis (MFI values).

```
## Need to input
```

b. Apply cutoff. How many people are seropositive according to this cutoff, and what proportion of people are seropositive?

```
# #Applying cutoff:
#   #ifelse function, if first statement is true, then outcome is set to 1, and if first statement is f
#   #since the first statement is a vector, then outcome will be a vector of 1's and 0's.
# seropositivity <- ifelse(sample_data$WMEV > cutoff, 1, 0) #1 indicates seropositive, and 0 indicates
#
# #number of people seropositve and seronegative
# cat("Table of number seronegative and seronegative", "\n")
# table(seropositivity,useNA="always")
```

```
#
# #percent of people seropositive and negative
# cat("Table of percent seronegative and seronegative", "\n")
# round(prop.table(table(seropositivity,useNA="always")),3)*100
```

c. Calculate the confidence interval for this seroprevalence

```
# # Set your parameters
# x <- 712 #number seropositive. Get the number seropositive in your sample from the seropositive tab
# n <- nrow(sample_data) #total number of samples in your data, note in these data we saw above there
# conf <- 0.95 #confidence interval. 95% is a standard CI but you can adjust this if you want
#
# # Exact interval
# ci <- binom.exact(x, n, conf.level = conf) #epitools function
# #
# cat("CI lower", round(ci$lower,4)*100, "%, CI upper", round(ci$upper,4)*100, "%")
```

d. For this specific antigen, how would you interpret this seropositivity and confidence interval?

e. What do you think about using this cutoff method for this antigen? What are the assumptions that went into this cutoff method?