Reading in Serological Data

2025-05-22

Introduction

The purpose of this document is to read raw data from an imager or scanner into R and to understand the basic structure of these data. Upon completing this lab you should be able to:

- Read in your raw data into R
- Visualize the basic structures
- Identify the antigens being measured
- Identify control samples including standard curves and blank wells.

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.

```
#setwd("~seroanalytics_workshop/Data/)
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message=FALSE)
source("/Users/sberube1/Library/CloudStorage/OneDrive-UniversityofFlorida/Desktop/Research/Bead_serolog
```

```
##
## 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

##
## 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
```

```
## Package 'mclust' version 6.1.1
## Type 'citation("mclust")' for citing this R package in publications.
```

Overall structure

In general, data from serological studies are broken down into plates. Plates represents a unit of usually 96 wells (can be up to 384) ordered into 8 rows labeled A through H and 12 columns labeled 1 through 12. Each well represents a single sample either from the study or a control sample, which can include blank wells, positive and negative controls, and serial dilutions which are typically referred to as standard curves. A machine that measures fluorescence resulting from target antibodies in a sample binding a specific antigen will output a single CSV file for each plate. This CSV usually contains:

- The date the plate was run
- The machine number of the imager or scanner
- The sample volume
- The start and end time of the plate reading
- The last time calibrations, verification, and fluidics tests were run on that particular machine
- The median fluorescence intensity or MFI for each antigen and each sample (including controls) measured in the assay - Note that samples are identified both by their location on the plate which contains row and column number (e.g. A1, B3, F9) etc., and the sample name.
- The number of beads the machine was able to capture per antigen region in each sample (we usually impose a minimum number of beads required to perform further analysis e.g. 30 or 50).
- The average net MFI which is the MFI minus the average background reading (the average of the MFI for all background wells for each antigen).
- The dilution factor of the samples
- Any warnings and errors.

Reading in and Visualizing Files

We will start by reading in 1 plate of our example dataset and looking at the first 6 rows.

```
raw_dat <- read.csv("/Users/sberube1/Library/CloudStorage/OneDrive-UniversityofFlorida/Desktop/Resear
head(raw_dat)
```

```
##   Program             xPONENT       X MAGPIX X.1 X.2 X.3 X.4 X.5 X.6 X.7 X.8
## 1   Build           4.3.309.1
## 2    Date               9/1/23 9:42 AM
## 3
## 4      SN        MAGPX19057723
## 5   Batch 20230901_training_P1
## 6 Version                    1
##   X.9 X.10 X.11 X.12 X.13
## 1
## 2
## 3
## 4
## 5
## 6
```

To view information about the machine, calibration and dates we can use the following code.

```
View(raw_dat[c(1:36),])
```

Next we can visualize the MFI (median fluoresence intensity) values for antigen in each sample. These are extremely important since they typically form the basic unit of all further pre-processing and subsequent analyses.

```
View(raw_dat[c(38:137),])
```

Notice, there are separate rows for each well of the plate, which corresponds to unique samples. There are multiple different types of samples including cohort samples, controls, standard curves, and blanks. We can use the following code to visualize the list of samples in our first plate:

```
raw_dat[c(42:137),2]
```

```
##  [1] "Unknown1"              "Unknown2"              "Unknown3"
##  [4] "Unknown4"              "Unknown5"              "Unknown6"
##  [7] "Unknown7"              "Unknown8"              "Unknown9"
## [10] "Unknown10"             "Unknown11"             "Unknown12"
## [13] "Unknown13"             "Unknown14"             "Unknown15"
## [16] "Unknown16"             "Unknown17"             "Unknown18"
## [19] "Unknown19"             "Unknown20"             "Unknown21"
## [22] "Unknown22"             "Unknown23"             "Unknown24"
## [25] "Unknown25"             "Unknown26"             "Unknown27"
## [28] "Unknown28"             "Unknown29"             "Unknown30"
## [31] "Unknown31"             "Unknown32"             "Unknown33"
## [34] "Unknown34"             "Unknown35"             "Unknown36"
## [37] "Unknown37"             "Unknown38"             "Unknown39"
## [40] "Unknown40"             "Unknown41"             "Unknown42"
## [43] "Unknown43"             "Unknown44"             "Unknown45"
## [46] "Unknown46"             "Unknown47"             "Unknown48"
## [49] "Unknown49"             "Unknown50"             "Unknown51"
## [52] "Unknown52"             "Unknown53"             "Unknown54"
## [55] "Unknown55"             "Unknown56"             "Unknown57"
## [58] "Unknown58"             "Unknown59"             "Unknown60"
## [61] "Unknown61"             "Unknown62"             "Unknown63"
## [64] "Unknown64"             "P1"                    "P2"
## [67] "P3"                    "P4"                    "P5"
## [70] "P6"                    "P7"                    "P8"
## [73] "Std Curve wruv 1:80"   "Std Curve wruv 1:40"   "Std Curve wruv 1:20"
## [76] "Std Curve wruv 1:10"   "Std Curve wruv 1:5"    "Std Curve wruv 1:2.5"
## [79] "Std Curve wruv 1:1.25" "Std Curve wruv 1:0.625" "Std Curve wmev 1:3000"
## [82] "Std Curve wmev 1:1500" "Std Curve wmev 1:750"  "Std Curve wmev 1:375"
## [85] "Std Curve wmev 1:187.5" "Std Curve wmev 1:93.8" "Std Curve wmev 1:46.9"
## [88] "Std Curve wmev 1:23.4" "Std Curve wmev 1:11.7" "Std Curve wmev 1:5.9"
## [91] "POS1"                  "POS2"                  "NEG1"
## [94] "NEG2"                  "BLANK1"                "BLANK2"
```

Here, our positive controls (POS1 and POS2) are pooled positive controls (expected to be positive to all pathogen antigens). The negative controls (NEG1 and NEG2) are from people from a non-endemic area, and are expected to be negative for all antigens.Blanks (BLANK1 and BLANK2) don't include any serum/DBS, and can show how much fluorescence reagents might emit in the absence of any meaningful binding between antibodies and antigens.

We can also use the following code to visualize the names of antigens our dataset:

```
unname(c(raw_dat[c(41),c(3:15)]))
```

```
## [[1]]
## [1] "SNAP"
##
## [[2]]
```

```
## [1] "WNV"
##
## [[3]]
## [1] "YF"
##
## [[4]]
## [1] "JE3"
##
## [[5]]
## [1] "ZIKA"
##
## [[6]]
## [1] "DENV"
##
## [[7]]
## [1] "CHIKV"
##
## [[8]]
## [1] "GLURPR2"
##
## [[9]]
## [1] "CSP"
##
## [[10]]
## [1] "PfAMA1"
##
## [[11]]
## [1] "PfMSP119"
##
## [[12]]
## [1] "WRUV"
##
## [[13]]
## [1] "WMEV"
```

```
#how many antigens are there in this dataset?
```

We can have responses to a control bead that can give information about non-specific binding levels. In this dataset, SNAP serves as this control.

Next, we can visualize Net MFI which is the MFI for each antigen - the average MFI of all blank wells (see the sample list above to determine how many blank wells are in this dataset).

```
View(raw_dat[c(139:236),])
```

Next, we can visualize the bead count or Count which is number of beads captured by the imager or scanner for each antigen in each sample. It is common to set a minimum threshold of bead count (eg. 30 or 50) for any downstream analysis since low bead count can produce innacurate measurements and can be indiciative of issues with the assay. We can use this data to check that a minimum number of beads were measured for each antigen (eg 30 or 50 beads), and exclude results if too few beads were measured.

```
View(raw_dat[c(238:335),])
```

Now we can examine the dilution factor. This likely will be the same for all of the samples. This is important to note for inter assay replicability.

```r
View(raw_dat[c(444:542),])
```

Exercise 1: Repeat the above steps for sample plates 2, 3, and 4. These are located in the Data folder (which you should already have as your working directory). In your investigation identify any differences in any key variables across plates including, the names of antigens (even differences in capitalization or spacing are important), the names of control samples, sample dilutions, or dates the plates were run on. Note there may be no differences, but you should verify.

Demographic and Additional data

In addition to the raw sample data, you will likely have additional demographic or epidemiological data about the samples. In this training dataset, we have sex and age. This could also include variables relating to the study design e.g. in a cluster-based survey this information may include cluster id and cluster specific survey weights.We also need to have an identification variable to be able to match the output of the serological assay with other information about the samples.

First, we will read in the demographic data associated with the example dataset.

```r
demographics_dat <- read.csv("/Users/sberube1/Library/CloudStorage/OneDrive-UniversityofFlorida/Deskt
head(demographics_dat)
```

```
##   Luminex_id    Luminex_id.1 id age sex
## 1   Unknown1 Plate1_Unknown1  1   0   2
## 2   Unknown2 Plate1_Unknown2  2   0   2
## 3   Unknown3 Plate1_Unknown3  3   0   2
## 4   Unknown4 Plate1_Unknown4  4   0   2
## 5   Unknown5 Plate1_Unknown5  5   0   1
## 6   Unknown6 Plate1_Unknown6  6   0   2
```

Other additional data can be useful are: * Specimen information (DBS, plasma) * Manual flag for sample quality

Tidy data

The data you previously visualized was in a wide format. This format describes data that can have multiple unique measurements in a single row, in this case multiple different antigen measurements in a single row. By using the read and tidy function below, our goal will be to transform this data from wide format to long format. Long format describes data where each measurement is contained in a different row. We will also want to add certain information such as standard curve information which contains the sample name, location, and dilution factor of the standard curve.

The read and tidy takes the following arguments as input: * file name (csv file containing the plate output) * plate number (numerical id for a particular plate) * number of wells (usually 96) * antigen names (these need to be written exactly as they appear in the CSV file) * control samples (the sample name of the control samples) * background samples (the sample name of the control samples) * standard curve values (a data frame that desribes the sample name, dilution, plate location, and replicate number (in the event there is more than 1 standard curve on the plate)). * bead threshold, this is the minimum number of beads per antigen in each sample required for downstream analysis

```r
agx_names <- c("SNAP",    "WNV",  "YF",   "JE3",  "ZIKA", "DENV", "CHIKV",    "GLURPR2",  "CSP",  "Pf

ctrl_samples <- c("POS1", "POS2", "NEG1", "NEG2")
bg_samples <- c("BLANK1", "BLANK2")
standard_curve_df= data.frame(Sample= paste0("P", 1:8),
                              Dilution= c(1/100, 1/200, 1/400, 1/800, 1/1600, 1/3200, 1/6400, 1/128
```

```
                              Location= c("65(1,A9)", "66(1,B9)", "67(1,C9)", "68(1,D9)", "69(1,E9)", "
                              Replicate= rep(1,8))



plate_1_tidy<- read_and_tidy(file_name = "/Users/sberube1/Library/CloudStorage/OneDrive-UniversityofF
                              plate_number=1,
                              num_wells=96,
                              antigen_names=agx_names,
                              control_samples=ctrl_samples,
                              background_samples=bg_samples,
                              standard_curve_values=standard_curve_df,
                              bead_threshold=30)
```

The output of the read and tidy function is a long dataframe with the measurements of one antigen in one sample per row. It includes the following variables:

- Location: physical location on the plate (A1,B1,C1),
- Sample: sample name
- Antigen: antigen name
- MFI: mean fluorescence intensity
- BeadCount: bead count
- Plate: identifier for each plate (usually a number)
- Sample_Type: one of TestSample (cohort sample), Ctrl (control), StdCurve (standard curve), BG (blank well).
- Low_Beads: indicator of whether the bead count is low according to the threshold, 0 for no (there are sufficient beads), 1 for yes (there are insufficient beads).

```
View(plate_1_tidy)
```

Exercise 2: repeat this process for other example plates (2,3,and 4)

Lab questions:

1. For the following pieces of information, check where you have each piece of information (ie which dataset), or whether there is information you're missing. You may have information that is missing, or information that just wasn't collected or may not be applicable. If you have additional information (variables like latitude/longitude, species (ie if non-human) etc.) if available as well.

Sample meta-data Unique sample ID Sample date Specimen information (DBS, plasma) Manual flag for sample quality Demographics: age, sex, location, vacc status Survey indicators: sample weight Manual flag for whether the sample is included in the data analysis

Experiment meta-data Plate ID Date run Sample ID Well ID MFI (by antigen) Net MFI (by antigen) Bead count (by antigen)

Control meta-data Unique control ID Positive or negative control Control for which antigen Source/description of control (e.g., "US non-traveler who's never had Pf malaria") Dilution, if part of a standard curve (e.g., "Positive Pool 1:50") Short description of what the control is (e.g., "NIBSC XX international standard")

2. Is there any data that you're missing that may limit the analysis you're able to do?

3. Make a list of the antigens you're using in your analysis.

4. Adjust the code below to read your data sets into R and look at the first 6 rows of the dataset.

5. Take a look at your raw datasets. Is the format similar or different to the raw data set we went through earlier? In what ways is it similar and different?

```r
#Use the View function to look through the datasets you just loaded
```

5. Do you have standard curves in your data? Are they are on the same or different plates than your samples? List the antigens the you have standard curves for. How many points are on these curves, and what dilutions are they?

6. Apply the read and tidy function to your data sets.

a. Adjust the following code

```r
#run the read_and_tidy function on your data. Adjust the code from the above run_read_and_tidy_plate1
```

b. Look at this data frame to ensure the function worked as you expected.

```r
head(plate_1_tidy)
```

```
##   Location   Sample Antigen  MFI BeadCount Plate Sample_Type Low_Beads
## 1  1(1,A1) Unknown1    SNAP   80       173     1  TestSample         0
## 2  2(1,B1) Unknown2    SNAP   74       122     1  TestSample         0
## 3  3(1,C1) Unknown3    SNAP  214       161     1  TestSample         0
## 4  4(1,D1) Unknown4    SNAP 1495       119     1  TestSample         0
## 5  5(1,E1) Unknown5    SNAP  226       132     1  TestSample         0
## 6  6(1,F1) Unknown6    SNAP  189       120     1  TestSample         0
```

8. Redo steps 7 for any other raw plates you have.