# ImpactTB/BAA: Standard Operating Procedures for Data Analysis

Colorado State University Coding Team

2022-07-07

# Contents

# Chapter 1

# Overview

Here, we have built a comprehensive guide to wet lab data collection, sample processing, and computational tool creation for robust and efficient data analysis and dissemination.

# Chapter 2

# Introduction

## 2.1 About the project: Immune Mechanisms of Protection against Mycobacterium tuberculosis (IMPAc-TB)

The objective of the IMPAc-TB program is to get a thorough understanding of the immune responses necessary to avoid initial infection with *Mycobacterium tuberculosis (Mtb)*, formation of latent infection, and progression to active TB illness. To achieve these goals, the National Institute of Allergy and Infectious Diseases awarded substantial funding and established multidisciplinary research teams that will analyze immune responses against *Mtb* in animal models (mice, guinea pigs, and non-human primates) and humans, as well as immune responses elicited by promising vaccine candidates. The contract awards establish and give up to seven years of assistance for IMPAc-TB Centers to explain the immune responses required for *Mtb* infection protection.

The seven centers that are part of the study are (in alphabetical order):

1. Colorado State University
2. Harvard T.H. Chan School of Public Health
3. Seattle Children Hospital

Colorado State University Team and role of each member: Dr. Marcela Henao-Tamayo: Principal Investigator Dr. Brendan Podell: Principal Investigator Dr. Andres Obregon-Henao: Research Scientist-III Dr. Taru S. Dutt: Research Scientist-I

# Chapter 3

# Initial mouse characteristics

Here is a review of existing methods.

# Chapter 4

# Mouse Weights

For example, Baazim et al. (2022) said … (Baazim et al., 2022)

Mice are weighed in grams weekly to monitor clinical status as TB patients frequently display weight loss as clinical symptom associated with disease progression.

Weights are recorded in an excel worksheet.

Column titles are as follows: who_collected date_collected sex dob notch_id mouse_number weight unit cage_number group notes

Groups included are: bcg, saline, bcg+id93, saline+id93, saline+noMtb

The notes column contains information regarding clinical observations.

good reference: https://elifesciences.org/articles/74419#s4

```
library(readxl)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

## 4.1 Read in data

# Chapter 5

# Data is stored in one excel sheet, each week is one sheet named as the date -> return vector for each sheet name

# Chapter 6

# can also use rio to read in the data, more streamlined

## 6.1 Clean data

```
data$before_vaccination %>%
 select("sex", "notch_id",  "weight", "cage_number", "group") #"mouse_number" it couldn't find th
```

```
## [1] sex         notch_id    weight      cage_number group
## <0 rows> (or 0-length row.names)
```

## 6.2 Summary statistics

# Chapter 7

# Body weight over time graph and statistics

# Chapter 8

# Weight loss over time graph and statistics

# Chapter 9

# Weight vs CFU

# Chapter 10

# Weight vs ELISA results

# Chapter 11

# Weight vs lesion burden

# Chapter 12

# Colony forming units to determine bacterial counts

## 12.1 Data description

The data are collected in a spreadsheet with multiple sheets. The first sheet (named "[x]") is used to record some metadata for the experiment, while the following sheets are used to record CFUs counts from the plates used for samples from each organ, with one sheet per organ. For example, if you plated data from both the lung and spleen, there would be three sheets in the file: one with the metadata, one with the plate counts for the lung, and one with the plate counts for the spleen.

The metadata sheet is used to record information about the overall process of plating the data. Values from this sheet will be used in calculating the bacterial load in the original sample based on the CFU counts. This spreadsheet includes the following columns:

- `organ`: Include one row for each organ that was plated in the experiment. You should name the organ all in lowercase (e.g., "lung", "spleen"). You should use the same name to also name the sheet that records data for that organ for example, if you have rows in the metadata sheet for "lung" and "spleen", then you should have two other sheets in the file, one sheet named "lung" and one named "spleen", which you'll use to store the plate counts for each of those organs.
- `prop_resuspended`: In this column, give the proportion of that organ that was plated. For example, if you plated half the lung, then in the "lung" row of this spread sheet, you should put 0.5 in the `prop_resuspended` column.
- `total_resuspended_uL`: This column contains an original volume of tissue

homogenate. For example, raw lung tissue is homogenized in 500 uL of PBS in a tube containing metal beads.

- `og_aliquot_uL`: 100 uL of th total_resuspended slurry would be considered an original aliquot and is used to peform serial dilutions.
- `dilution_factor`: Amount of the original stock solution that is present in the total solution, after dilution(s)
- `plated_uL`: Amount of suspension + diluent plated on section of solid agar

## 12.2 Read in data

```
library(readxl)
library(dplyr)
library(purrr)
library(tidyr)
library(stringr)

#Replace w/ path to CFU sheet
path <- c("DATA/Copy of baa_cfu_sheet.xlsx")

sheet_names <- excel_sheets(path)
sheet_names <- sheet_names[!sheet_names %in% c("metadata")]

merged_data <- list()

for(i in 1:length(sheet_names)){

  data <- read_excel(path, sheet = sheet_names[i]) %>%
    mutate(organ = paste0(sheet_names[i]))

  data <- data %>%
    #mutate(missing_col = NA) %>%
    mutate_if(is.double, as.numeric) %>%
    mutate_if(is.numeric, as.character) %>%
    pivot_longer(starts_with("dil_"), names_to = "dilution",
                 values_to = "CFUs") %>%
    mutate(dilution = str_extract(dilution, "[0-9]+"),
           dilution = as.numeric(dilution))


  merged_data[[i]] <- data


}
```

```
all_data <- bind_rows(merged_data, .id = "column_label") %>%
    select(-column_label)
```

## 12.3 Exploratory analysis and quality checks

## 12.4 Exploratory analysis

**Dimensions of input data:**

Based on the input data, data were collected for the following organ or organs:

The following number of mice were included for each:

The following number of replicates were recorded at each count date for each experimental group:

The following number of dilutions and dilution level were recorded for each organ:

**People who plated and collected the data. Date or dates of counting:**

Based on the input data, the plates included in these data were counted by the following person or persons: Based on the input data, the plates included in these data were counted on the following date or dates:

```
all_data %>%
  select(organ, who_plated, who_counted, count_date) %>%
  distinct()
```

```
## # A tibble: 3 x 4
##   organ  who_plated who_counted count_date
##   <chr>  <chr>      <chr>       <chr>
## 1 lung   BK         BK          "\"February 21 2022\""
## 2 lung   BK         BK          "\"April 18 2022\""
## 3 spleen JR         JR          "\"April 25 2022\""
```

**Distribution of CFUs at each dilution:**

WE NEED TO ADD SAMPLE CFU PLOTS

Here's a plot that shows how many plates were too numerous to count at each dilution level:

Here is a plot that shows how the CFU counts were distributed by dilution level in the data:

## 12.5 Identify a good dilution for each sample

```
# Make all_data into tidy data and filter for CFUs between 10-75

tidy_cfu_data <- all_data %>%
  mutate(dilution = str_extract(dilution, "[0-9]+"),
         dilution = as.numeric(dilution)) %>%
  filter(CFUs >= 10 & CFUs <= 75) %>%
  mutate(CFUs = as.numeric(CFUs))
```

## 12.6 Calculate CFUs from best dilution/Estimate bacterial load for each sample based on good dilution

```
# Calculating CFU/ml for every qualifying replicate between 10-75 CFUs. Column binding
meta <- read_excel(path, sheet = "metadata")

tidy_cfu_meta_joined <- inner_join(tidy_cfu_data, meta) %>%
  group_by(groups) %>%
  mutate(CFUs_per_ml = (CFUs * (dilution_factor^2) * (total_resuspension_mL/volume_plat
  select(organ, count_date, who_plated, who_counted, groups,  mouse, dilution,  CFUs, C
  ungroup()
```

```
## Joining, by = "organ"
```

```
tidy_cfu_meta_joined
```

```
## # A tibble: 146 x 9
##    organ count_date           who_plated who_counted groups mouse dilution  CFUs
##    <chr> <chr>                <chr>      <chr>       <chr>  <chr>    <dbl> <dbl>
##  1 lung  "\"February 21 2022~ BK         BK          group~ A            3    53
##  2 lung  "\"February 21 2022~ BK         BK          group~ A            5     4
##  3 lung  "\"February 21 2022~ BK         BK          group~ A            6     2
##  4 lung  "\"February 21 2022~ BK         BK          group~ B            3   119
##  5 lung  "\"February 21 2022~ BK         BK          group~ B            4    48
##  6 lung  "\"February 21 2022~ BK         BK          group~ B            5    18
##  7 lung  "\"February 21 2022~ BK         BK          group~ C            3   120
##  8 lung  "\"February 21 2022~ BK         BK          group~ C            4    32
##  9 lung  "\"February 21 2022~ BK         BK          group~ D            3    53
## 10 lung  "\"February 21 2022~ BK         BK          group~ D            4    31
## # ... with 136 more rows, and 1 more variable: CFUs_per_ml <dbl>
```

## 12.7 Create initial report information for these data

## 12.8 Sample ANOVA

```
cfu_stats <- tidy_cfu_meta_joined %>%
  group_by(organ) %>%
  nest() %>%
  mutate(aov_result = map(data, ~aov(CFUs_per_ml ~ groups, data = .x)),
         tukey_result = map(aov_result, TukeyHSD),
         tidy_tukey = map(tukey_result, broom::tidy)) %>%
  unnest(tidy_tukey, .drop = TRUE) %>%
  separate(contrast, into = c("contrast1", "contrast2"), sep = "-") %>%
  select(-data, -aov_result, -tukey_result, -term, -null.value)# %>%
```

```
## Warning: The `.drop` argument of `unnest()` is deprecated as of tidyr 1.0.0.
## All list-columns are now preserved.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
  # filter(adj.p.value <= 0.05)

cfu_stats
```

```
## # A tibble: 9 x 7
## # Groups:   organ [2]
##   organ  contrast1 contrast2 estimate conf.low conf.high adj.p.value
##   <chr>  <chr>     <chr>        <dbl>    <dbl>     <dbl>       <dbl>
## 1 lung   group_2   group_1     -15.0     -39.4      9.34       0.377
## 2 lung   group_3   group_1     -13.1     -39.2     13.1        0.562
## 3 lung   group_4   group_1      -2.57    -27.1     22.0        0.993
## 4 lung   group_3   group_2       1.98    -22.7     26.7        0.997
## 5 lung   group_4   group_2      12.5     -10.5     35.5        0.491
## 6 lung   group_4   group_3      10.5     -14.4     35.4        0.689
## 7 spleen group_2   group_1     -21.5     -48.8      5.80       0.146
## 8 spleen group_3   group_1     -17.6     -45.9     10.7        0.294
## 9 spleen group_3   group_2       3.90    -23.4     31.2        0.935
```

## 12.9 Save processed data to database

## 12.10 Example one

## 12.11 Example two

# Chapter 13

# Enzyme-linked immunosorbest assay (ELISA)

ELISA is a standard molecular biology assay for detecting and quantifying a variety of compounds, including peptides, proteins, and antibodies in a sample. The sample could be serum, plasma, or bronchoalveolar lavage fluid (BALF).

#### 13.0.0.1 Importance of ELISA

An antigen-specific reaction in the host results in the production of antibodies, which are proteins found in the blood. In the event of an infectious disease, it aids in the detection of antibodies in the body.ELISA is distinguishable from other antibody-assays in that it produces quantifiable findings and separates non-specific from specific interactions by serial binding to solid surfaces, which is often a polystyrene multiwell plate.

In IMPAc-TB project, it is crucial to evaluate the if the vaccine is eliciting humoral immunity and generating antibodies against vaccine antigen. ELISA will be used to determine the presence of Immunoglobulin (Ig) IgG, IgA, and IgM in the serum different time points post-vaccination.

#### 13.0.0.2 Principle of ELISA

ELISA is based on the principle of antigen-antibody interaction. An antigen must be immobilized on a solid surface and then complexed with an enzyme-linked antibody in an ELISA. The conjugated enzyme's activity is evaluated by incubating it with a substrate to yield a quantifiable result, which enables detection. There are four basic steps of ELISA:

**1. Coating multiwell plate with antigen/antibody**: This step depends on what we want to detect the sample. If we need to evaluate the the presence of antibody, the plate will be coated with the antigen, and vice versa. To coat the plate, a fixed concentration of antigen (protein) is added to a 96 well high-binding plate (charged plate). Plate is incubated over night with the antigen at 4 degree celsius (as proteins are temperature sensitive) so that antigens are completely bound to the well.

**2. Blocking**: It is possible that not each and every site of the well is coated with the targeted antigen, and there could be uncovered areas. It is important to block those empty spaces so that primary antibody (which we will add to the next step) binds to these spaces and give us false positive results. For this, microplate well surface-binding sites are blocked with an unrelated protein or other substance.Most common blocking agents are bovine serum albumin, skim milk, and casein. One of the best blocking agents is to use the serum from the organism in which your secondary (detection antibody) is raised. For example, if the secondary antibody is raised in goat, then we can use goat serum as a blocking agent.

**3. Probing**: Probing is the step where we add sample containing antibodies that we want to detect. This will be the primary antibody. If the antibodies against the antigen (which we have coated) are present in the sample, it will bind to the antigen with high affinity.

**4. Washing**: After the incubation of sample containing primary antibody, the wells are washed so that any unbound antibody is washed away. Washing solution contains phosphate buffer saline + 0.05% tween-20 (a mild detergent). 0.05% tween-20 washes away all the non-specific interactions as those are not strong, but keeps all the specific interaction as those are strong and cannot be detached with mild detergent.

**5. Detection**: To detect the presence of antibody-antigen complex, a secondary antibody labelled with an enzyme (usually horseradish peroxidase) is added to the wells, incubated and washed.

**6. Signal Measurement**: Finally to detect "if" and "how much" of the antibody is present, a chromogenic substrate (like 3,3',5,5'-Tetramethylbenzidine) is added to the wells, which can be cleaved the the enzyme that is tagged to the secondary antibody. The color compund is formed after the addition of the substrate, which is directly proportional to the amount of antibody present in the sample. The plate is read on a plate reader, where color is converted to numbers.

## 13.1 Read in data from excel file

```
elisa_raw_data <- read_excel("DATA/elisa_s1_07-25-20.xlsx", sheet = "S1", col_names = F
```

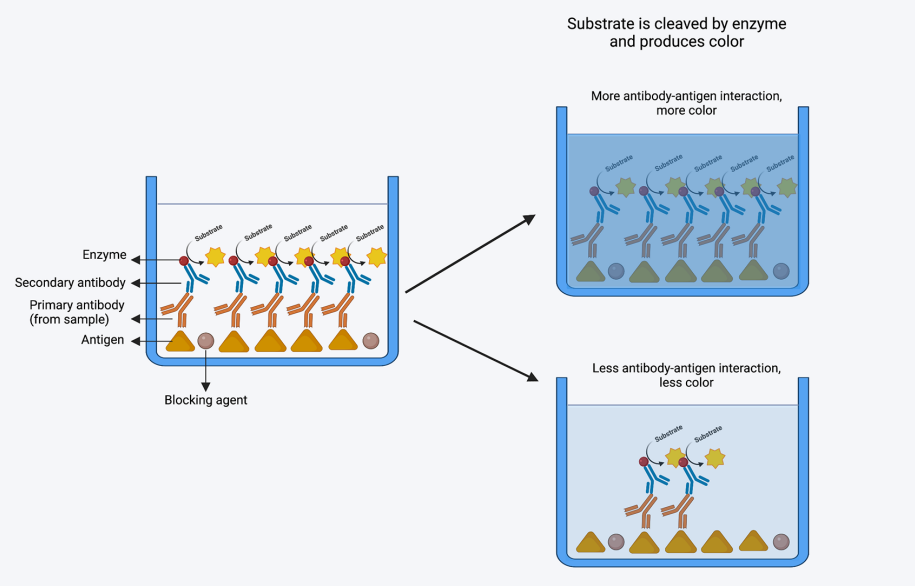


Figure 13.1: A caption

```
## New names:
## * `` -> ...1
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * ...
```

```
head(elisa_raw_data)
```

```
## # A tibble: 6 x 12
##   ...1          ...2  ...3 ...4   ...5  ...6 ...7   ...8  ...9 ...10 ...11 ...12
##   <chr>        <dbl> <dbl> <chr> <dbl> <dbl> <chr> <dbl> <dbl> <chr> <dbl> <dbl>
## 1 5.199999999~ 0.05  0.069 6.3E~ 0.061 0.122 0.16~ 0.145 0.135 6.80~ 0.053 0.05
## 2 7.900000000~ 0.098 0.069 6.80~ 0.115 0.202 5.89~ 0.134 0.069 0.106 0.05  0.075
## 3 8.899999999~ 0.133 0.119 OVRF~ 3.87  2.32  OVRF~ 3.85  2.12  OVRF~ 3.21  1.02
## 4 OVRFLW       3.46  1.16  OVRF~ 3.80  2.36  OVRF~ 3.70  1.49  OVRF~ 3.68  1.63
## 5 3.815999999~ 1.82  0.446 3.89~ 3.42  1.13  OVRF~ 2.33  0.608 OVRF~ 3.41  1.10
## 6 OVRFLW       3.69  1.43  OVRF~ 3.66  1.27  3.839 1.74  0.444 2.49~ 0.637 0.704
```

```
# Convert all columns to numeric

elisa_raw_data_numeric <- elisa_raw_data %>%
  mutate_if(is.character, as.numeric)
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
# pivot longer the data

elisa_raw_data_tidy <- pivot_longer(data = elisa_raw_data_numeric, cols = "...1":"...1

# remove "..." from the first column

elisa_raw_data_tidy$well_id <- str_replace(elisa_raw_data_tidy$well_id, "...", "")

# Add new column to the data_frame

elisa_raw_data_tidy_new <- elisa_raw_data_tidy %>%
  mutate(name = rep(LETTERS[1:8], each = 12))

elisa_raw_data_tidy_new <- elisa_raw_data_tidy_new %>%
  mutate(well_id = paste0(name, well_id)) %>%
  select(-name)

head(elisa_raw_data_tidy_new)
```

```
## # A tibble: 6 x 2
##   well_id od_450nm
##   <chr>      <dbl>
## 1 A1         0.052
## 2 A2         0.05
## 3 A3         0.069
## 4 A4         0.063
## 5 A5         0.061
## 6 A6         0.122
```

## 13.3 Read in second data set

```
elisa_label_data <- read_excel("DATA/elisa_s1_07-25-20.xlsx", sheet = "S1", col_names = FALSE,  r
```

```
## New names:
## * `` -> ...1
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * ...
```

```
head(elisa_label_data)
```

```
## # A tibble: 6 x 12
##   ...1        ...2   ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10 ...11 ...12
##   <chr>       <chr>  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 blank       secon~ naïv~ 1A-1~ 1A-1~ 1A-1~ 1A-2~ 1A-2~ 1A-2~ 1A-3~ 1A-3~ 1A-3~
## 2 1A-4 (1/250 1A-4 ~ 1A-4~ 1B-1~ 1B-1~ 1B-1~ 1B-2~ 1B-2~ 1B-2~ 1B-3~ 1B-3~ 1B-3~
## 3 1B-4 (1/250 1B-4 ~ 1B-4~ 2A-1~ 2A-1~ 2A-1~ 2A-2~ 2A-2~ 2A-2~ 2A-3~ 2A-3~ 2A-3~
## 4 2B-1 (1/250 2B-1 ~ 2B-1~ 2B-2~ 2B-2~ 2B-2~ 2B-3~ 2B-3~ 2B-3~ 2B-4~ 2B-4~ 2B-4~
## 5 3A-1 (1/250 3A-1 ~ 3A-1~ 3A-2~ 3A-2~ 3A-2~ 3A-3~ 3A-3~ 3A-3~ 3A-4~ 3A-4~ 3A-4~
## 6 3B-1 (1/250 3B-1 ~ 3B-1~ 3B-2~ 3B-2~ 3B-2~ 3B-3~ 3B-3~ 3B-3~ 3B-4~ 3B-4~ 3B-4~
```

```
# pivot longer the data

elisa_label_data_tidy <- pivot_longer(data = elisa_label_data, cols = "...1":"...12", names_to =

# remove "..." from the first column

elisa_label_data_tidy$well_id <- str_replace(elisa_label_data_tidy$well_id, "...", "")

# Add new column to the data_frame

elisa_label_data_tidy_new <- elisa_label_data_tidy %>%
  mutate(name = rep(LETTERS[1:8], each = 12))

elisa_label_data_tidy_new <- elisa_label_data_tidy_new %>%
  mutate(well_id = paste0(name, well_id)) %>%
  select(-name)

head(elisa_label_data_tidy_new)
```

```
## # A tibble: 6 x 2
##   well_id information
##   <chr>   <chr>
## 1 A1      blank
```

```
## 2 A2      secondary
## 3 A3      naïve (1/250)
## 4 A4      1A-1 (1/250
## 5 A5      1A-1 (1/1250
## 6 A6      1A-1 (1/6250
```

## 13.4 Join the OD table with the information table

```
elisa_data = elisa_raw_data_tidy_new %>% inner_join(elisa_label_data_tidy_new, by="well

head(elisa_data)
```

```
## # A tibble: 6 x 3
##   well_id od_450nm information
##   <chr>      <dbl> <chr>
## 1 A1         0.052 blank
## 2 A2         0.05  secondary
## 3 A3         0.069 naïve (1/250)
## 4 A4         0.063 1A-1 (1/250
## 5 A5         0.061 1A-1 (1/1250
## 6 A6         0.122 1A-1 (1/6250
```

## 13.5 Separate the information tabke into sample ID and dilution columns

```
tidy_elisa_data <- separate(elisa_data, col = "information", into = c("sample_id", "dil
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 2 rows [1, 2].
```

```
head(tidy_elisa_data)
```

```
## # A tibble: 6 x 4
##   well_id od_450nm sample_id   dilution
##   <chr>      <dbl> <chr>       <chr>
## 1 A1         0.052 "blank"     <NA>
## 2 A2         0.05  "secondary" <NA>
## 3 A3         0.069 "naïve "    1/250)
## 4 A4         0.063 "1A-1 "     1/250
## 5 A5         0.061 "1A-1 "     1/1250
## 6 A6         0.122 "1A-1 "     1/6250
```

```
tidy_elisa_data <- tidy_elisa_data %>%
  mutate(dilution = str_extract(dilution, "(/)[0-9]+"),
```

```
         dilution = str_replace(dilution, "/", ""),
         dilution = as.numeric(dilution))

tidy_elisa_data <- tidy_elisa_data %>%
  select(well_id, sample_id, dilution, od_450nm)

head(tidy_elisa_data)
```

```
## # A tibble: 6 x 4
##   well_id sample_id   dilution od_450nm
##   <chr>   <chr>          <dbl>    <dbl>
## 1 A1      "blank"           NA    0.052
## 2 A2      "secondary"       NA    0.05
## 3 A3      "naïve "         250    0.069
## 4 A4      "1A-1 "          250    0.063
## 5 A5      "1A-1 "         1250    0.061
## 6 A6      "1A-1 "         6250    0.122
```

## 13.6 ELISA data analysis optimization

# Bibliography

Baazim, H., Antonio-Herrera, L., and Bergthaler, A. (2022). The interplay of immunology and cachexia in infection and cancer. *Nature Reviews Immunology*, 22(5):309–321.