ImpactTB/BAA: Standard Operating Procedures for Data Analysis

Colorado State University Coding Team

2022-06-09

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
4. Arizona?

Chapter 3

Initial mouse characteristics

Here is a review of existing methods.

Chapter 4

Mouse weights

Mice are weighed in grams weekly and 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.

```
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

```
weight_data <- read_xlsx("DATA/body_weights.xlsx")
```

4.2 Clean data

```
weight_data %>%
  select(sex, mouse_number, weight, cage_number, group)
```

```
## # A tibble: 0 x 5
## # ... with 5 variables: sex <lgl>, mouse_number <lgl>, weight <lgl>,
## #   cage_number <lgl>, group <lgl>
```

4.3 Summary statistics

4.4 Graph

Chapter 5

Colony forming units to determine bacterial counts

5.1 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

5.2 Read in data

```r
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)
```

5.3 Exploratory analysis and quality checks

5.4 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:

5.5 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))
```

5.6 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_pla
  select(organ, count_date, who_plated, who_counted, groups,  mouse, dilution,  CFUs,
  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>
```

5.7 Create initial report information for these data

5.8 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
```

5.9 Save processed data to database

5.10 Example one

5.11 Example two

Chapter 6

ELISA Words

We have finished a nice book.