Quantitative Profiling of Bacterial Communities

Ezra Valido

2025-04-30

Sample Computation

This is sample computation for the "Quantitative profiling of bacterial communities via full length 16S rRNA sequencing with internal controls: Optimization and validation across diverse human microbiomes".

Sample data is provided (attached in supplement). Organisms grouped in "Others" were contaminants. These were organisms found only in the positive and negative controls excluding the spike-in organisms.

```
# Load required libraries
library(readxl)
library(dplyr)
```

```
##
## 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
```

```
library(writexl)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.5.1     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
# Upload the data
sample <- read_excel("Sample.xlsx", sheet = "data")

# Calculate spike ratios
allo_copy <- (0.001 / 125.4) * 140000000
imt_copy <- (0.001 / 125.4) * 60000000
#We first take the ratio of the total spike-in DNA to the total DNA in the
#control (The 125.4ng is provided in the data sheet of the spike).
#We then multiply to the total expected 16S copies for each organism in
#the spike-in.


# Proportion of the spike-in in the sequencing results
spike <- read_excel("Sample.xlsx", sheet = "spike")
#Change species names for easier coding
spike[1, "species"] <- "allo"
spike[2, "species"] <- "imt"

# Get allo values (row 1, exclude column 1 "species")
allo_vector <- as.numeric(spike[spike$species == "allo", -1])
names(allo_vector) <- paste0("sk", 1:5)

# Get imt values (row 2, exclude column 1 "species")
imt_vector <- as.numeric(spike[spike$species == "imt", -1])
names(imt_vector) <- paste0("sk", 1:5)

# Calculate abundance
sample.abundance <- sample %>%
  # Apply Allobacillus corrections
  mutate(across(starts_with("sk"), ~(.x * allo_copy) / allo_vector[cur_column()],
                .names = "{.col}_allo")) %>%
  # Apply Imtechella corrections
  mutate(across(starts_with("sk"), ~(.x * imt_copy) / imt_vector[cur_column()],
                .names = "{.col}_imt")) %>%
  ## Calculate average abundance for each sample since there is an estimation for
  #both   organism
  ## We adjust to the ratio of the total DNA input used in PCR amplification to
  #the   total DNA extracted. For low concentration, this could be the ratio of
  #the input   volume (from sample) used in PCR to the volume (elution) of the
  #extracted DNA.This is dependent on the experimental set-up. For this example,
  #9.5uL of eluted DNA was used during PCR from a total volume of 50uL thus 0.19
  #was used.
  ##We adjust as well to the sample volume during DNA extraction. For this
  #example 250uL was the initial sample volume during extraction.
  ##Lastly, we adjust to the number 16S rRNA copies derived from the rrnDB.
  #If this   is skipped, the data needs to be represented as 16S copies/mL.
    mutate(across(
    matches("^sk[1-5]_allo$"),
    ~ ((.x + get(sub("_allo$", "_imt", cur_column()))) / 2) / (0.19 * 0.25* copy),
    .names = "{sub('_allo$', '.adjusted', .col)}"
  )) %>%
  # Remove intermediate columns if needed
  select(-matches("_(allo|imt)$"))
```

```r
#Data cleaning to only include the species and the samples
sample.abundance  <- sample.abundance  %>%
  select(c(species, sk1.adjusted:sk5.adjusted))
```