7. Sample Results and Controls

Cob Staines

2026-02-18

Table of contents

This tutorial is available as a .qmd on Github.

Motivation

- Familiarize ourselves with the sample data (what is there, how is it stored and accessed)
- Demonstrate pulling samples along with corresponding negative controls

R

In general, sample results for Bd-qPCR, AMPs, and mucosome assays and their negative controls are stored in similar, parallel structures in the database.

Setup

```
# minimal packages for RIBBiTR DB data discovery
librarian::shelf(tidyverse, dbplyr, RPostgres, DBI, RIBBiTR-BII/ribbitrrr)

# establish database connection
dbcon <- hopToDB("ribbitr")
```

```
Connecting to 'ribbitr'... Success!
```

```
# load table metadata
mdt <- tbl(dbcon, Id("public", "all_tables")) %>%
  filter(table_schema == "survey_data") %>%
  collect()

# load column metadata
mdc <- tbl(dbcon, Id("public", "all_columns")) %>%
  filter(table_schema == "survey_data") %>%
  collect()
```

Data discovery and pulling

Looking at the survey_data schema diagram, we can browse to see which tables and columns we want. We can also consult the table or column metadata. The two observation tables with the primary data of interest are called "capture" and "bd_qpcr_results".

Point to support tables

```
# survey tables
db_capture = tbl(dbcon, Id("survey_data", "capture"))
db_survey = tbl(dbcon, Id("survey_data", "survey"))
db_visit = tbl(dbcon, Id("survey_data", "visit"))
db_site = tbl(dbcon, Id("survey_data", "site"))
db_region = tbl(dbcon, Id("survey_data", "region"))
db_country = tbl(dbcon, Id("survey_data", "country"))

# sample results tables
db_sample = tbl(dbcon, Id("survey_data", "sample"))
db_bdqpcr = tbl(dbcon, Id("survey_data", "bd_qpcr_results"))
db_amp_gia = tbl(dbcon, Id("survey_data", "amp_gia"))
db_amp_intensity = tbl(dbcon, Id("survey_data", "amp_maldi_intensity"))
db_amp_peak = tbl(dbcon, Id("survey_data", "amp_maldi_peak"))
db_amp_total = tbl(dbcon, Id("survey_data", "amp_total"))
db_mucosome_gia = tbl(dbcon, Id("survey_data", "mucosome_gia"))
```

Identify data of interest

Let's begin by taking a look at all the capture data since 2022 for which we have AMP total results. We can preview the columns in there `amp_total` table with `colnames(db_amp_total)`, by looking at the metadata (`mdc`), or consulting the survey_data schema diagram.

```
# un-executed (no "collect()") preliminary query
data_oi = db_amp_total %>%
  inner_join(db_sample, by = "sample_id") %>%
  inner_join(db_capture, by = "capture_id") %>%
  left_join(db_survey, by = "survey_id") %>%
  left_join(db_visit, by = "visit_id") %>%
  left_join(db_site, by = "site_id") %>%
  left_join(db_region, by = "region_id") %>%
  left_join(db_country, by = "country_id") %>%
  filter(date >= "2022-01-01",
         sample_type == "amp")

# investigate what data are there
(stats_data_oi_species = data_oi %>%
  group_by(taxon_capture) %>%
  count() %>%
  collect() %>%
  arrange(desc(n)))
```

```
# A tibble: 11 x 2
# Groups:   taxon_capture [11]
   taxon_capture                   n
   <chr>                       <int64>
 1 hylodes_phyllodes             165
 2 colostethus_panamansis        153
 3 rana_catesbeiana              133
 4 rana_sierrae                  132
 5 ischnocnema_henselii          120
 6 lithobates_warszewitschii      94
 7 pseudacris_crucifer            39
 8 rana_muscosa                   12
 9 rana_clamitans                  7
10 rana_pipiens                    6
11 atelopus_varius                 1
```

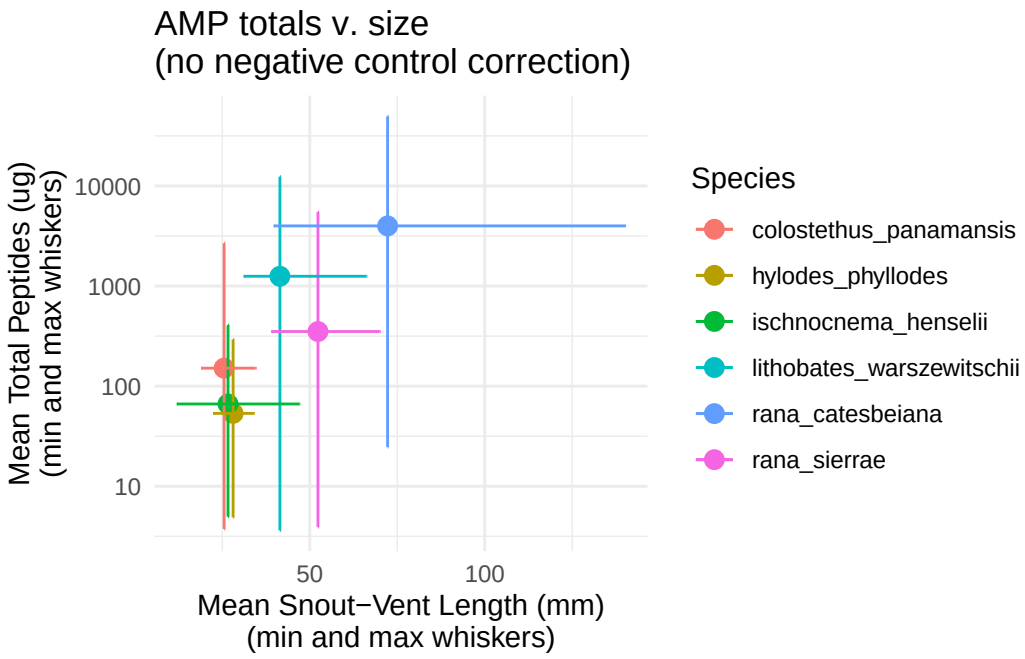
Filter to species and variables of interest

```
species_oi = stats_data_oi_species %>%
  filter(n > 50) %>%
  pull(taxon_capture)

data_amp_total = data_oi %>%
  filter(taxon_capture %in% species_oi) %>%
  select(result_id,
         sample_id,
         sample_name_amp,
         total_peptides_ug,
         notes,
         negative_control_group_id,  # we will touch on this later
         taxon_capture,
         svl_mm,
         date,
         site,
         region,
         country) %>%
  collect()
```

Summarize by species & visualize

```
data_summary_species = data_amp_total %>%
  group_by(taxon_capture) %>%
  summarise(n = n(),
            svl_mm_mean = mean(svl_mm, na.rm = TRUE),
            svl_mm_sd = sd(svl_mm, na.rm = TRUE),
            svl_mm_min = min(svl_mm, na.rm = TRUE),
            svl_mm_max = max(svl_mm, na.rm = TRUE),
            total_peptides_ug_mean = mean(total_peptides_ug),
            total_peptides_ug_sd = sd(total_peptides_ug),
            total_peptides_ug_min = min(total_peptides_ug),
            total_peptides_ug_max = max(total_peptides_ug),
            .groups = "drop")

ggplot(data_summary_species, aes(x = svl_mm_mean, y = total_peptides_ug_mean, color = taxon_
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = total_peptides_ug_min, ymax = total_peptides_ug_max), width = 0,
  geom_errorbar(aes(xmin = svl_mm_min, xmax = svl_mm_max), height = 0, orientation = "y") +
  theme_minimal() +
  labs(x = "Mean Snout-Vent Length (mm)\n(min and max whiskers)", y = "Mean Total Peptides (
  scale_y_log10() +
  ggtitle("AMP totals v. size\n(no negative control correction)")
```

```
`height` was translated to `width`.
```



Great! But what about our negative controls? We should subtract any peptides found in corresponding negative controls from samples to make this more accurate of the frogs sampled.

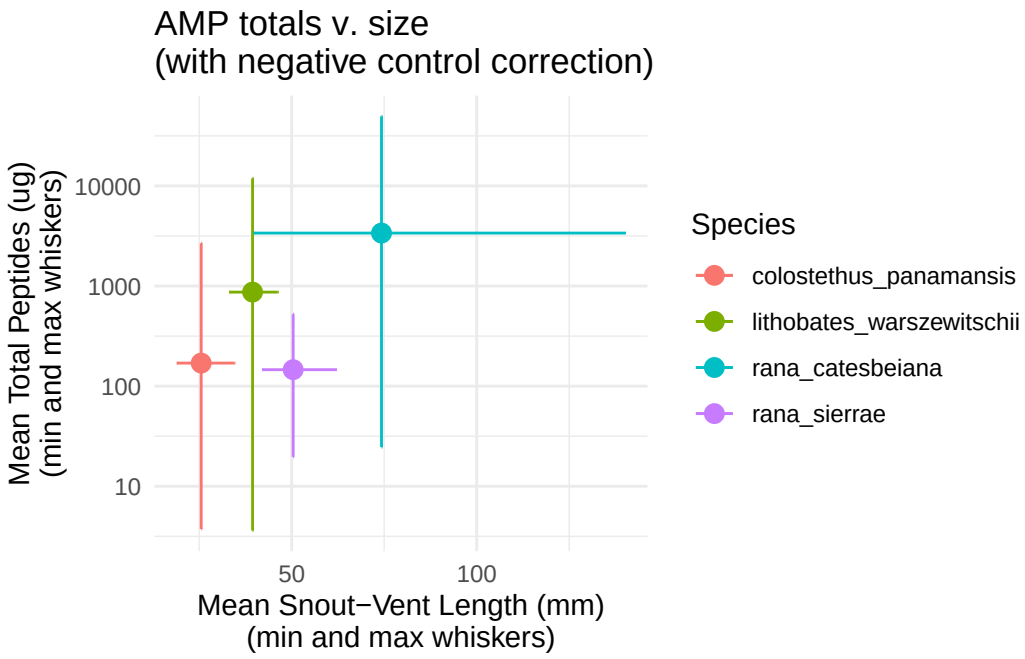
```
# get negative_control_group_ids
nc_groups_count = data_amp_total %>%
  group_by(negative_control_group_id) %>%
  count()

# we see that a substantial number don't have corresponding negative controls in the databas

nc_groups = data_amp_total %>%
  filter(!is.na(negative_control_group_id)) %>%
  pull(negative_control_group_id) %>%
  unique()

nc_data = db_amp_total %>%
  inner_join(db_sample, by = "sample_id") %>%
  filter(negative_control,
         negative_control_group_id %in% nc_groups) %>%
  select(result_id,
         sample_id,
         sample_name_amp,
         total_peptides_ug,
```

```
          notes,
          negative_control,
          negative_control_group_id) %>%
  collect()

# in general there may be multiple negative controls within a negative control group, good p
nc_group_data = nc_data %>%
  group_by (negative_control_group_id) %>%
  summarise(nc_n = n(),
            total_peptides_ug_nc_mean = mean(total_peptides_ug),
            .groups = "drop")
```

join samples with negative controls for correction

```
# calculate corrected
data_amp_total_nc = data_amp_total %>%
  left_join(nc_group_data, by = "negative_control_group_id") %>%
  mutate(total_peptides_ug_corrected = total_peptides_ug - total_peptides_ug_nc_mean)

data_summary_species_nc = data_amp_total_nc %>%
  group_by(taxon_capture) %>%
  filter(!is.na(total_peptides_ug_corrected)) %>%
  summarise(n = n(),
            svl_mm_mean = mean(svl_mm, na.rm = TRUE),
            svl_mm_sd = sd(svl_mm, na.rm = TRUE),
            svl_mm_min = min(svl_mm, na.rm = TRUE),
            svl_mm_max = max(svl_mm, na.rm = TRUE),
            total_peptides_ug_mean = mean(total_peptides_ug),
            total_peptides_ug_sd = sd(total_peptides_ug),
            total_peptides_ug_min = min(total_peptides_ug),
            total_peptides_ug_max = max(total_peptides_ug),
            .groups = "drop")

ggplot(data_summary_species_nc, aes(x = svl_mm_mean, y = total_peptides_ug_mean, color = tax
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = total_peptides_ug_min, ymax = total_peptides_ug_max), width = 0,
  geom_errorbar(aes(xmin = svl_mm_min, xmax = svl_mm_max), height = 0, orientation = "y") +
  theme_minimal() +
  labs(x = "Mean Snout-Vent Length (mm)\n(min and max whiskers)", y = "Mean Total Peptides (
  scale_y_log10() +
  ggtitle("AMP totals v. size\n(with negative control correction)")
```

```
`height` was translated to `width`.
```



<- 6. Microclimate Workflow | 8. Database Refesher ->