

4. Data Workflow

Cob Staines

2024-11-07

Table of contents

Motivation	1
R Data Workflow	1
Setup	1
Data discovery and pulling	2
Python Data Workflow	6
Setup	6
Data discovery and pulling	7

Motivation

- Familiarize ourselves with more data manipulation tools
- Run through a complete workflow including: database connection, data discovery, data pulling, and data manipulation

R Data Workflow

Let’s run through a realistic data scenario from the beginning.

Setup

These setup steps will all be familiar to you by now.

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

# establish database connection
dbcon = hopToDB("ribbitr")
```

Connecting to database... Success!

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

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

Data discovery and pulling

Suppose we are interested in capture and Bd swab data. Specifically, we want to compare Bd qPCR results for juvenile-to-adult individuals, captured in 2015 or later, across species and sites.

Looking at the table metadata, the two observation tables with most of the data of interest are called “capture” and “bd_qpcr_results”.

Pull capture table

```
# capture table, select, filter
db_capture = tbl(dbcon, Id("survey_data", "capture")) %>%
  select(species_capture,
         body_temp_c,
         life_stage,
         sex,
         capture_animal_state,
         bd_swab_id,
         survey_id)
```

Join with support tables, filter by date

```
# create chain object
chain_capture = tbl_chain("capture", mdc)

# join recursively, filter by date
db_capture_chain = tbl_join(dbcon, chain_capture, tbl = db_capture) %>%
  filter(date >= "2015-01-01")
```

```
Joining with bd_swab_lookup ... done.
Joining with survey ... done.
Joining with visit ... done.
Joining with site ... done.
Joining with region ... done.
Joining with country ... done.
```

Join with Bd table

```
db_bd_results = tbl(dbcon, Id("survey_data", "bd_qpcr_results")) %>%
  select(bd_swab_id,
         detected,
         average_target_quant)

db_capture_bd = db_capture_chain %>%
  inner_join(db_bd_results, by="bd_swab_id")
```

Explore # of samples by life stage, then filter

```
db_capture_bd %>%
  select(life_stage) %>%
  group_by(life_stage) %>%
  summarise(row_count = n()) %>%
  arrange(desc(row_count)) %>%
  collect()
```

```
# A tibble: 13 x 2
  life_stage      row_count
  <chr>          <int64>
1 adult          22746
2 <NA>           4515
3 juvenile       3937
4 tadpole        1666
5 subadult        984
6 aquatic_larvae  573
```

7	metamorph	428
8	larva	417
9	unknown	367
10	terrestrial_development	330
11	larvae	218
12	eggmass	7
13	Unknown	2

```
db_capture_bd_life = db_capture_bd %>%
  filter(life_stage %in% c("juvenile",
                          "subadult",
                          "adult"),
         !is.na(life_stage))
```

Explore # of samples by species, then filter

```
(spp_summary = db_capture_bd_life %>%
  select(species_capture) %>%
  group_by(species_capture) %>%
  summarise(sample_count = n()) %>%
  arrange(desc(sample_count)) %>%
  collect())
```

```
# A tibble: 131 x 2
  species_capture      sample_count
  <chr>              <int64>
1 rana_muscosa        13328
2 rana_clamitans      2366
3 rana_catesbeiana    1601
4 pseudacris_crucifer 1118
5 lithobates_sphenocephalus 965
6 rana_pipiens        551
7 notophthalmus_viridescens 535
8 lithobates_chiricahuensis 453
9 colostethus_panamensis 429
10 hyla_versicolor    417
# i 121 more rows
```

```
(spp_list = spp_summary %>%
  filter(sample_count >= 100) %>%
  pull(species_capture))
```

```

[1] "rana_muscosa"           "rana_clamitans"
[3] "rana_catesbeiana"      "pseudacris_crucifer"
[5] "lithobates_sphenocephalus" "rana_pipiens"
[7] "notophthalmus_viridescens" "lithobates_chiricahuensis"
[9] "colostethus_panamensis"   "hyla_versicolor"
[11] "hyla_chrysoscelis"        "anaxyrus_americanus"
[13] "silverstoneia_flotator"   "lithobates_warszewitschii"
[15] "anaxyrus_fowleri"         "acris_blanchardi"
[17] "rhaebo_haematiticus"      "sachatamia_albomaculata"
[19] "ambystoma_opacum"         "hyla_cinerea"
[21] "lithobates_sylvaticus"    "pseudacris_feriarum"
[23] "smilisca_sila"            "plethodon_glutinosus"
[25] "ambystoma_maculatum"      "plethodon_cinereus"
[27] "unknown_species"         "desmognathus_fuscus"
[29] "lithobates_blairi"

```

```

db_capture_bd_life_spp = db_capture_bd_life %>%
  filter(species_capture %in% spp_list,
         !is.na(species_capture))

```

collect data

```

data_capture_bd_query = db_capture_bd_life_spp %>%
  collect()

colnames(data_capture_bd_query)

```

```

[1] "species_capture"      "body_temp_c"      "life_stage"
[4] "sex"                  "capture_animal_state" "bd_swab_id"
[7] "survey_id"            "detection_type"    "visit_id"
[10] "date"                 "survey_time"       "site_id"
[13] "site"                 "region_id"         "region"
[16] "country_id"           "country_name"      "detected"
[19] "average_target_quant"

```

```
head(data_capture_bd_query)
```

```

# A tibble: 6 x 19
  species_capture body_temp_c life_stage sex capture_animal_state bd_swab_id
  <chr>           <dbl> <chr>   <chr> <chr> <chr>
1 rhaebo_haematiti~ NA    juvenile unkn~ alive 150608_04
2 rhaebo_haematiti~ NA    juvenile unkn~ alive 150608_05

```

```

3 colostethus_pana~      NA    adult      unkn~ alive      150610_01
4 rhaebo_haematiti~     26.7 adult      unkn~ alive      150612_01
5 rhaebo_haematiti~     25.7 juvenile unkn~ alive      150612_02
6 rhaebo_haematiti~     26.6 adult      unkn~ alive      150612_03
# i 13 more variables: survey_id <chr>, detection_type <chr>, visit_id <chr>,
#   date <date>, survey_time <chr>, site_id <chr>, site <chr>, region_id <chr>,
#   region <chr>, country_id <chr>, country_name <chr>, detected <dbl>,
#   average_target_quant <dbl>

```

These data are ready to be analyzed and visualized!

Python Data Workflow

Let's run through a realistic data scenario from the beginning.

Setup

These setup steps will all be familiar to you by now.

```

# minimal packages for RIBBiTR DB Workflow
import ibis
from ibis import _
import pandas as pd
import dbconfig
import db_access as db

# establish database connection
dbcon = ibis.postgres.connect(**dbconfig.ribbitr)

# load table metadata
mdt = dbcon.table(database = "public", name = "all_tables").to_pandas()

# load column metadata
mdc = (
    dbcon.table(database="public", name="all_columns")
    .filter(_.table_schema == 'survey_data')
    .to_pandas()
)

```

Data discovery and pulling

Suppose we are interested in capture and Bd swab data. Specifically, we want to compare Bd qPCR results for juvenile-to-adult individuals, captured in 2015 or later, across species and sites.

Looking at the table metadata, the two observation tables with most of the data of interest are called “capture” and “bd_qpcr_results”.

Pull capture table

```
# capture table, select, filter
db_capture = (
  dbcon.table(database="survey_data", name="capture")
  .select([
    'species_capture',
    'body_temp_c',
    'life_stage',
    'sex',
    'capture_animal_state',
    'bd_swab_id',
    'survey_id'
  ])
)
```

Join with support tables, filter to date

```
# create chain object
chain_capture = db.tbl_chain("capture", mdc)

# join recursively, filter by date
db_capture_chain = (
  db.tbl_join(dbcon, chain_capture, tbl=db_capture)
  .filter(_.date >= '2015-01-01')
)
```

```
Joining with bd_swab_lookup ... done.
Joining with survey ... done.
Joining with visit ... done.
Joining with site ... done.
Joining with region ... done.
Joining with country ... done.
```

Join with Bd table

```
# bd qpcr results lazy table
db_bd_results = (
    dbcon.table(database="survey_data", name="bd_qpcr_results")
    .select([
        'bd_swab_id',
        'detected',
        'average_target_quant'
    ])
)

# join capture and bd tables
db_capture_bd = (
    db_capture_chain
    .inner_join(db_bd_results, db_capture_chain.bd_swab_id == db_bd_results.bd_swab_id)
)
```

Explore # of samples by life stage, then filter

```
# count by life stage
life_stage_counts = (
    db_capture_bd
    .group_by('life_stage')
    .aggregate(row_count=_.count())
    .order_by(_.row_count.desc())
    .to_pandas()
)

print(life_stage_counts)
```

	life_stage	row_count
0	adult	22746
1	None	4515
2	juvenile	3937
3	tadpole	1666
4	subadult	984
5	aquatic_larvae	573
6	metamorph	428
7	larva	417
8	unknown	367
9	terrestrial_development	330
10	larvae	218
11	eggmass	7


```
# filter to desired life stages
db_capture_bd_life = (
  db_capture_bd
  .filter(_.life_stage.isin(['juvenile','subadult', 'adult']) & _.life_stage.notnull())
)
```

Explore # of samples by species, then filter

```
# count by species
spp_summary = (
  db_capture_bd_life
  .group_by('species_capture')
  .aggregate(sample_count=_.count())
  .order_by(_.sample_count.desc())
  .to_pandas()
)
print(spp_summary)
```

	species_capture	sample_count
0	rana_muscosa	13328
1	rana_clamitans	2366
2	rana_catesbeiana	1601
3	pseudacris_crucifer	1118
4	lithobates_sphenocephalus	965
..
126	silverstoneia_spp	1
127	diasporus_spp	1
128	craugastor_monnichorum	1
129	acris_crepitans	1
130	hyalinobatrachium_talamancae	1

[131 rows x 2 columns]

```
# generate species list
spp_list = spp_summary[spp_summary['sample_count'] >= 100]['species_capture'].tolist()

# filter to species in spp_list
db_capture_bd_life_spp = (
  db_capture_bd_life
  .filter(_.species_capture.isin(spp_list) & _.species_capture.notnull())
)
```

Pull data

```
# pull data
data_capture_bd_query = db_capture_bd_life_spp.to_pandas()

# print column names
data_capture_bd_query.columns
```

```
Index(['species_capture', 'body_temp_c', 'life_stage', 'sex',
       'capture_animal_state', 'bd_swab_id', 'survey_id', 'bd_swab_id_right',
       'visit_id', 'survey_id_right', 'detection_type', 'visit_id_right',
       'survey_time', 'site_id', 'date', 'site_id_right', 'region_id', 'site',
       'country_id', 'region_id_right', 'region', 'country_id_right',
       'country_name', 'detected', 'average_target_quant'],
      dtype='object')
```

```
# preview data
data_capture_bd_query.head()
```

	species_capture	body_temp_c	...	detected	average_target_quant
0	rhaebo_haematiticus	NaN	...	0.0	0.00
1	lithobates_warszewitschii	18.6	...	1.0	1007.69
2	lithobates_warszewitschii	NaN	...	1.0	257.43
3	lithobates_warszewitschii	21.8	...	1.0	56.15
4	lithobates_warszewitschii	21.6	...	0.0	0.00

```
[5 rows x 25 columns]
```

These data are ready to be analyzed and visualized!

[Previous Tutorial: Data Pulling](#)