Retrieve standard data from Seatrack

Jens Åström

2019-02-23

Contents

Connecting to the database

Users at NINA and the Polarinstitute (using a computer that is within these networks' IP-addresses) can connect to the Seatrack database. It is a PostgreSQL (9.6) database answering to the address `seatrack.nina.no`, on the standard port 5432. Users should use their individual login user names and passwords. Contact Jens Åström (jens.astrom@nina.no) for details about usernames and passwords.

This instruction deals with the preferred way of connecting to the database, using R and the `seatrackR` package. Another option is to connect through a dedicated database management software, such as Pgadmin3 (or 4), HeidiSQL, or similar. Some users may prefer to use the MS Access interface.

To simplify the connection, use the convenience function `connectSeatrack`. This creates a connection named `con` by using the packages `DBI` and `RPostgres`.

```
require(seatrackR)
```

```
## Loading required package: seatrackR
```

```
## Loading required package: DBI
```

```
connectSeatrack(Username = "testreader", Password = "testreader")
```

Custom queries

As of now, 4 functions exist to retreive data from the database through prebuilt queries. Apart from that, users are free to use their own queries through the functions in `DBI` and `dplyr`, using the connection named `con` made by the `connectSeatrack()`-function.

It is perfecty fine to download data through your own custom queries. Creating interesting queries requires some knowledge about the structure of the database however. Pgadmin3(4) would be a useful tool to get further info on that. For now, we show a simple query involving just one table. Here we get the different

locations currently recorded from the Faroe Islands (Coordinates not updated). Note that you have to load the DBI package and use its query functions.

```
require(DBI)

myQuery <- "SELECT * from metadata.location
              WHERE colony_int_name = 'Faroe Islands'"

faroeLocations <- dbGetQuery(con, myQuery)
head(faroeLocations)
```

```
##                                     id  location_name colony_int_name
## 1 b7e16a70-0bf0-11e8-82b0-005056b165f3     Gassadalur   Faroe Islands
## 2 b7e1f8fa-0bf0-11e8-82b0-005056b165f3     Glyvursnes   Faroe Islands
## 3 b7e67e98-0bf0-11e8-82b0-005056b165f3    Havnardalur   Faroe Islands
## 4 b7ee5884-0bf0-11e8-82b0-005056b165f3 Kirkjubøholmur   Faroe Islands
## 5 b7f25f38-0bf0-11e8-82b0-005056b165f3 Lamba grotbrot   Faroe Islands
## 6 b7f52fce-0bf0-11e8-82b0-005056b165f3     Leynavatni   Faroe Islands
##   colony_nat_name    lat    lon geom
## 1         Føroyar 66.585 12.229 <NA>
## 2         Føroyar 66.585 12.229 <NA>
## 3         Føroyar 66.585 12.229 <NA>
## 4         Føroyar 66.585 12.229 <NA>
## 5         Føroyar 66.585 12.229 <NA>
## 6         Føroyar 66.585 12.229 <NA>
```

Position data

The primary data of the positions of the birds is stored in the table `positions.postable`. This includes all entered positions in the database.

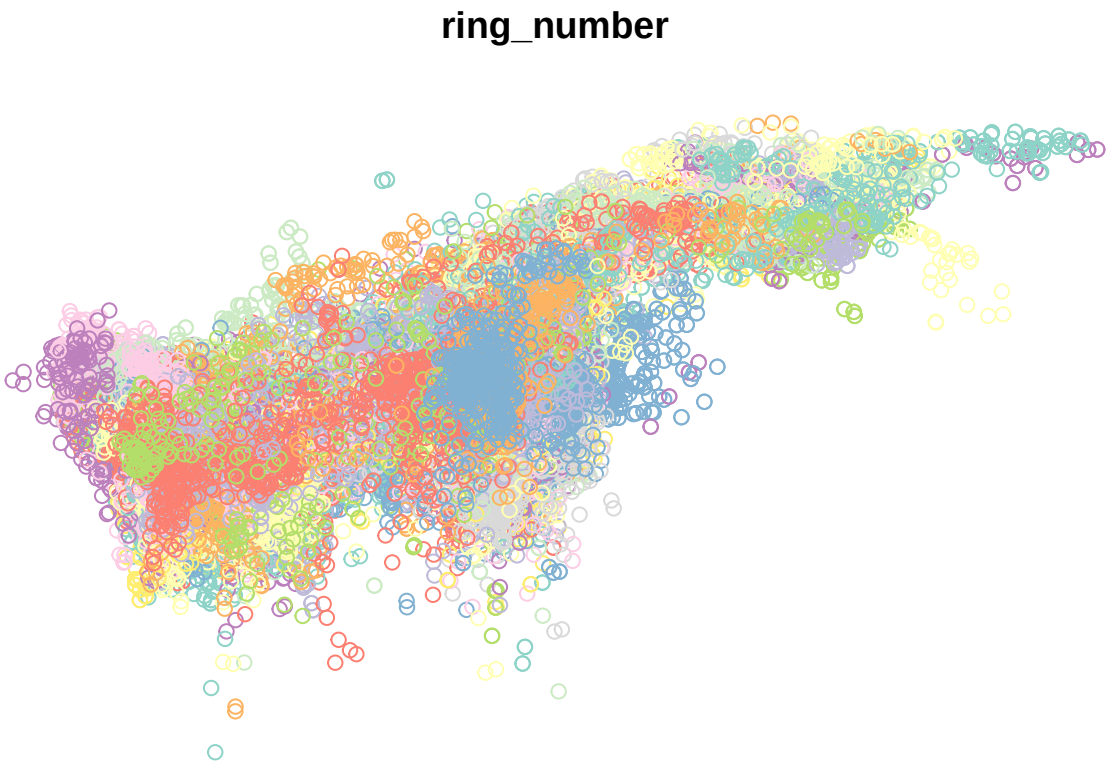
The `getPosdata` function retrieves this table, with options to subselect only specific species, colonies, responsible contact person, specific ring numbers, and years. There is also an option to limit the records to a set number of rows, and to load the position coordinates as a spatial object.

```
eynhallowPositions <- getPosdata(selectColony = "Eynhallow", loadGeometries = T)
eynhallowPositions
```

```
## Simple feature collection with 132590 features and 45 fields
## geometry type:  POINT
## dimension:      XY
## bbox:           xmin: -66.59141 ymin: 29.88674 xmax: 82.16431 ymax: 79.68491
## epsg (SRID):    4326
## proj4string:    +proj=longlat +datum=WGS84 +no_defs
## # A tibble: 132,590 x 46
##    id    date_time           logger logger_id logger_model year_tracked
##    <chr> <dttm>              <chr>  <chr>     <chr>        <chr>
##  1 74a6~ 2015-12-16 11:31:33 M970_~ M970      c250         2015_16
##  2 9a6b~ 2011-01-25 00:05:30 22300~ 22300     mk15         2010_11
##  3 a907~ 2010-04-14 00:21:30 12155~ 12155     mk13         2009_10
##  4 944d~ 2016-06-26 00:20:06 M717_~ M717      c250         2015_16
##  5 9a6b~ 2011-02-01 12:23:00 22300~ 22300     mk15         2010_11
##  6 9a6b~ 2011-02-04 00:37:00 22300~ 22300     mk15         2010_11
##  7 9a69~ 2010-11-02 11:42:30 22300~ 22300     mk15         2010_11
```

```
##  8 99e5~ 2010-08-20 23:54:30 22029~ 22029     mk15         2010_11
##  9 a0f5~ 2011-01-31 11:51:00 22275~ 22275     mk15         2010_11
## 10 b2cf~ 2016-12-06 23:49:00 B349_~ B349      mk3006       2016_17
## # ... with 132,580 more rows, and 40 more variables: session_id <int>,
## #   year_deployed <int>, year_retrieved <int>, ring_number <chr>,
## #   euring_code <chr>, species <chr>, colony <chr>, lon_raw <dbl>,
## #   lat_raw <dbl>, lon_smooth1 <dbl>, lat_smooth1 <dbl>,
## #   lon_smooth2 <dbl>, lat_smooth2 <dbl>, disttocol_s2 <dbl>,
## #   eqfilter1 <int>, eqfilter2 <int>, eqfilter3 <int>,
## #   lat_smooth2_eqfilt3 <dbl>, sex <chr>, morph <chr>, subspecies <chr>,
## #   age <chr>, col_lon <dbl>, col_lat <dbl>, tfirst <dttm>,
## #   tsecond <dttm>, twl_type <int>, conf <int>, sun <dbl>, software <chr>,
## #   light_threshold <int>, analyzer <chr>, data_responsible <chr>,
## #   logger_yeartracked <chr>, posdata_file <chr>, import_date <date>,
## #   data_version <int>, database_version <int>, automated_ver <chr>,
## #   geometry <POINT [°]>
```

```
plot(eynhallowPositions["ring_number"])
```



Position data for export

The data sent to the Polar institute also have the subspecies names added to the records. The export ready positions data can be retrieved most easily through a specific export view. Note that this will download all records, and will take some time. Note that this export does not contain information on the used and deleted uuids.

```
newExport <- dbReadTable(con, Id(schema = "views", table = "export"))
nrow(newExport)
write.csv(newExport, file = "seatrack_export_2018-08-09.csv")
```

If you are interested in knowing separate old, deleted rows, these are found in the table `positions.deleted_uuid`.

```
deletedUuids <- dbReadTable(con, Id(schema = "positions", table = "deleted_uuid"))
nrow(deletedUuids)
write.csv(deletedUuids, file = "deletedUuids_2018-08-09.csv")
```

Other functions for download

There are some more convenience functions for retrieving information from the database as well. Here follows a quich demo.

The `getFileArchiveSummary` function

This function pulls together data from several tables with focus on the file archive. It should contain enough information to know what the individual raw files contain.

```
eynhallowFiles <- getFileArchiveSummary(selectColony = "Eynhallow")
eynhallowFiles
```

```
## # A tibble: 940 x 9
##    file_id session_id colony ring_number euring_code year_tracked
##      <int>      <int> <chr>  <chr>       <chr>       <chr>
##  1  154501     119472 Eynha~ FP96078     GBT         2006_7
##  2  154502     119472 Eynha~ FP96078     GBT         2006_7
##  3  154503     119472 Eynha~ FP96078     GBT         2006_7
##  4  154504     119472 Eynha~ FP96078     GBT         2006_7
##  5  154505     119472 Eynha~ FP96078     GBT         2006_7
##  6  154506     119474 Eynha~ FP96077     GBT         2006_7
##  7  154507     119474 Eynha~ FP96077     GBT         2006_7
##  8  154508     119474 Eynha~ FP96077     GBT         2006_7
##  9  154509     119474 Eynha~ FP96077     GBT         2006_7
## 10  154510     119474 Eynha~ FP96077     GBT         2006_7
## # ... with 930 more rows, and 3 more variables: logger_serial_no <chr>,
## #   logger_model <chr>, filename <chr>
```

The `getIndividInfo` function

This function summarizes all observation data for the individual birds. We can subselect the colony and year interval the bird where tracked.

```
hornoyaIndivids <- getIndividInfo(selectColony = "Hornoya", selectYear = "2014_15")
hornoyaIndivids
```

```
## # A tibble: 217 x 31
##    session_id colony year_tracked ring_number euring_code color_ring
##         <int> <chr>  <chr>        <chr>       <chr>       <chr>
##  1     122262 Horno~ 2014_15      5185355     NOS         W(NL1)
##  2     122262 Horno~ 2014_15      5185355     NOS         W(NL1)
```

```
##  3     122283 Horno~ 2014_15      5185357     NOS         W(NP1)
##  4     122283 Horno~ 2014_15      5185357     NOS         W(NP1)
##  5     122249 Horno~ 2014_15      5185359     NOS         W(NC1)
##  6     122249 Horno~ 2014_15      5185359     NOS         W(NC1)
##  7     122249 Horno~ 2014_15      5185359     NOS         W(NC1)
##  8     122249 Horno~ 2014_15      5185359     NOS         W(NC1)
##  9     122249 Horno~ 2014_15      5185359     NOS         W(NC1)
## 10     122249 Horno~ 2014_15      5185359     NOS         W(NC1)
## # ... with 207 more rows, and 25 more variables: species <chr>,
## #   subspecies <chr>, morph <chr>, status_age <chr>, status_sex <chr>,
## #   status_sexing_method <chr>, status_date <date>, weight <dbl>,
## #   scull <dbl>, tarsus <dbl>, wing <dbl>, breeding_stage <chr>,
## #   eggs <int>, chicks <int>, hatching_success <lgl>,
## #   breeding_success <lgl>, breeding_success_criterion <chr>,
## #   data_responsible <chr>, back_on_nest <lgl>, comment <chr>,
## #   latest_sex <chr>, latest_sexing_method <chr>, latest_age <chr>,
## #   latest_info_date <date>, eventType <chr>
```

!Note the weird duplicate records here! **TO BE FIXED**

```
hornoyaIndivids %>% print(width = Inf)
```

```
## # A tibble: 217 x 31
##    session_id colony  year_tracked ring_number euring_code color_ring
##         <int> <chr>   <chr>        <chr>       <chr>       <chr>
##  1     122262 Hornoya 2014_15      5185355     NOS         W(NL1)
##  2     122262 Hornoya 2014_15      5185355     NOS         W(NL1)
##  3     122283 Hornoya 2014_15      5185357     NOS         W(NP1)
##  4     122283 Hornoya 2014_15      5185357     NOS         W(NP1)
##  5     122249 Hornoya 2014_15      5185359     NOS         W(NC1)
##  6     122249 Hornoya 2014_15      5185359     NOS         W(NC1)
##  7     122249 Hornoya 2014_15      5185359     NOS         W(NC1)
##  8     122249 Hornoya 2014_15      5185359     NOS         W(NC1)
##  9     122249 Hornoya 2014_15      5185359     NOS         W(NC1)
## 10     122249 Hornoya 2014_15      5185359     NOS         W(NC1)
##    species         subspecies morph status_age    status_sex
##    <chr>           <chr>      <chr> <chr>         <chr>
##  1 Atlantic puffin <NA>       <NA>  adult_unknown <NA>
##  2 Atlantic puffin <NA>       <NA>  adult_unknown <NA>
##  3 Atlantic puffin <NA>       <NA>  adult_unknown <NA>
##  4 Atlantic puffin <NA>       <NA>  adult_unknown <NA>
##  5 Atlantic puffin <NA>       <NA>  adult_unknown <NA>
##  6 Atlantic puffin <NA>       <NA>  adult_unknown <NA>
##  7 Atlantic puffin <NA>       <NA>  adult_unknown <NA>
##  8 Atlantic puffin <NA>       <NA>  adult_unknown <NA>
##  9 Atlantic puffin <NA>       <NA>  adult_unknown <NA>
## 10 Atlantic puffin <NA>       <NA>  adult_unknown unknown
##    status_sexing_method status_date weight scull tarsus  wing
##    <chr>                <date>       <dbl> <dbl>  <dbl> <dbl>
##  1 dna                  2014-07-06     510    82   31     176
##  2 none_yet             2015-06-11     500    NA   NA      NA
##  3 dna                  2014-07-06     540    78   29.5   182
##  4 none_yet             2015-07-01     530    NA   NA      NA
##  5 dna                  2014-07-06     520    83   29.5   180
##  6 none_yet             2015-06-11     545    NA   NA      NA
```

```
##  7 none_yet             2015-06-11     545    NA   NA      NA
##  8 none_yet             2016-06-05     535    83   29.6   183
##  9 none_yet             2016-06-05     535    83   29.6   183
## 10 <NA>                 2017-06-08     550    84   30.3   182
##    breeding_stage          eggs chicks hatching_success breeding_success
##    <chr>                  <int>  <int> <lgl>            <lgl>
##  1 incubating                 1     NA NA               NA
##  2 rearing chicks            NA     NA TRUE             NA
##  3 incubating                 1     NA NA               NA
##  4 rearing chicks            NA     NA TRUE             NA
##  5 incubating                 1     NA NA               NA
##  6 rearing chicks            NA     NA TRUE             NA
##  7 rearing chicks            NA     NA TRUE             NA
##  8 breeding/stage_unknown    NA     NA NA               NA
##  9 breeding/stage_unknown    NA     NA NA               NA
## 10 unknown                   NA     NA NA               NA
##    breeding_success_criterion data_responsible     back_on_nest
##    <chr>                      <chr>                <lgl>
##  1 <NA>                       Rob Barret           NA
##  2 none                       Kjell Einar Erikstad NA
##  3 <NA>                       Rob Barret           NA
##  4 none                       Kjell Einar Erikstad NA
##  5 <NA>                       Rob Barret           NA
##  6 none                       Kjell Einar Erikstad NA
##  7 none                       Kjell Einar Erikstad NA
##  8 none                       Kjell Einar Erikstad NA
##  9 none                       Kjell Einar Erikstad NA
## 10 none                       Kjell Einar Erikstad NA
##    comment          latest_sex latest_sexing_method latest_age
##    <chr>            <chr>      <chr>                <chr>
##  1 BLOOD FOR SEXING unknown    <NA>                 adult_unknown
##  2 <NA>             unknown    <NA>                 adult_unknown
##  3 BLOOD FOR SEXING <NA>       dna                  adult_unknown
##  4 <NA>             <NA>       dna                  adult_unknown
##  5 BLOOD FOR SEXING unknown    <NA>                 adult_unknown
##  6 <NA>             unknown    <NA>                 adult_unknown
##  7 <NA>             unknown    <NA>                 adult_unknown
##  8 <NA>             unknown    <NA>                 adult_unknown
##  9 <NA>             unknown    <NA>                 adult_unknown
## 10 <NA>             unknown    <NA>                 adult_unknown
##    latest_info_date eventType
##    <date>           <chr>
##  1 2017-06-12       Deployment
##  2 2017-06-12       Retrieval
##  3 2016-07-05       Deployment
##  4 2016-07-05       Retrieval
##  5 2017-06-08       Deployment
##  6 2017-06-08       Retrieval
##  7 2017-06-08       Retrieval
##  8 2017-06-08       <NA>
##  9 2017-06-08       <NA>
## 10 2017-06-08       <NA>
## # ... with 207 more rows
```

Commonly used info

I have made a couple of views for som common information, that are displayed on the shiny app http://view.nina.no/seatrack/. These can be found like this as well.

```
shorttable <- dbReadTable(con, Id(schema = "views", table = "shorttable"))
shorttable
```

```
##   Antall.arter Antall.kolonier Antall.år Antall.positions Antall.individer
## 1           11              36        13          2509098             2723
##   Antall.tracks..logger_yeartracked.
## 1                               4830
```

```
shorttableeqfilter3 <- dbReadTable(con, Id(schema = "views", table = "shorttableeqfilter3"))
shorttableeqfilter3
```

```
##   Antall.arter Antall.kolonier Antall.år Antall.positions Antall.individer
## 1           11              36        13          1625885             2721
##   Antall.tracks..logger_yeartracked.
## 1                               4826
```