Package: otndo (via r-universe)

September 9, 2024

Title Understand your OTN data

Version 0.2.3

Description This package provides functions to summarize the various type of OTN-style data.

License CC BY 4.0

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

URL https://mhpob.github.io/otndo/, https://otndo.obrien.page/

BugReports https://github.com/mhpob/otndo/issues

Depends R (>= 4.1.0)

Imports cli, data.table, ggplot2, reactable, sf, mapview, quarto, readxl, rmarkdown, utils

Suggests curl, knitr, testthat (>= 3.0.0), writexl

Config/testthat/edition 3

VignetteBuilder knitr

Repository https://ocean-tracking-network.r-universe.dev

RemoteUrl https://github.com/mhpob/otndo

RemoteRef HEAD

RemoteSha aee9614f37999cdcd87b9a9259bf3f97d7be02d4

Contents

`deployment_gantt` *Create a Gantt-like chart of receiver deployments and recoveries*

Description

Create a Gantt-like chart of receiver deployments and recoveries

Usage

```
deployment_gantt(deployment)
```

Arguments

`deployment`	Cleaned deployment metadata sheet(s). Assumes it was cleaned with the internal `otndo:::clean_otn_deployment` function, read in, and converted to a data.table.

Examples

```
## Not run:
# Download a deployment metadata file
td <- file.path(tempdir(), "matos_test_files")
dir.create(td)

download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/pbsm/",
    "data-and-metadata/2018/pbsm-instrument-deployment-short-form-2018.xls/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm-instrument-deployment-short-form-2018.xls"),
  mode = "wb"
)

# Use internal function to clean
deployment_filepath <- otndo:::write_to_tempdir(
  type = "deployment",
  files = file.path(td, "pbsm-instrument-deployment-short-form-2018.xls"),
  temp_dir = td
)
```

```
# Make the Gantt chart
deployment_gantt(
  data.table::fread(deployment_filepath)
)

## End(Not run)
```

`make_receiver_push_summary` *Create summary reports of receiver project data from the OTN data push*

Description

Create summary reports of receiver project data from the OTN data push

Usage

```
make_receiver_push_summary(
  qualified = NULL,
  unqualified = NULL,
  update_push_log = FALSE,
  deployment = NULL,
  out_dir = getwd(),
  since = NULL,
  rmd = FALSE
)
```

Arguments

`qualified`, `unqualified`	Default is NULL: a character vector of file paths of your qualified/unqualified detections. These can be CSVs or zipped folders.
`update_push_log`	Do you wish to use an updated push log? Default is FALSE, but switch to TRUE if you haven't updated this package since the push occurred.
`deployment`	File path of user-supplied master OTN receiver deployment metadata.
`out_dir`	Defaults to working directory. In which directory would you like to save the report?
`since`	Date in YYYY-MM-DD format. Summarizes what's new since the provided date.
`rmd`	Logical. Compile via RMarkdown rather than Quarto?

Push log

To keep track of when ACT data pushes occur, a log is kept on the package's GitHub page. This is automatically downloaded every time you download or update the package, but you can avoid re-downloading the package by changing `update_push_log` to `TRUE`.

You can get similar behavior by providing a date to the `since` argument.

Output

This function creates an HTML report that can be viewed in your web browser.

Examples

```
## Not run:
td <- file.path(tempdir(), "matos_test_files")
dir.create(td)

download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/pbsm/",
    "data-and-metadata/archived-records/2018/",
    "pbsm-instrument-deployment-short-form-2018.xls/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm-instrument-deployment-short-form-2018.xls"),
  mode = "wb"
)


download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/pbsm/",
    "detection-extracts/pbsm_qualified_detections_2018.zip/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm_qualified_detections_2018.zip"),
  mode = "wb"
)
unzip(
  file.path(td, "pbsm_qualified_detections_2018.zip"),
  exdir = td
)

download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/pbsm/",
    "detection-extracts/pbsm_unqualified_detections_2018.zip/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm_unqualified_detections_2018.zip"),
  mode = "wb"
)
```

```
unzip(
  file.path(td, "pbsm_unqualified_detections_2018.zip"),
  exdir = td
)

qualified_files <- file.path(td, "pbsm_qualified_detections_2018.csv")
unqualified_files <- file.path(td, "pbsm_unqualified_detections_2018.csv")
deployment_files <- file.path(td, "pbsm-instrument-deployment-short-form-2018.xls")


make_receiver_push_summary(
  qualified = qualified_files,
  unqualified = unqualified_files,
  deployment = deployment_files,
  since = "2018-11-01"
)

## End(Not run)
```

`make_tag_push_summary` *Create summary reports of receiver project data from the OTN data push*

Description

Create summary reports of receiver project data from the OTN data push

Usage

```
make_tag_push_summary(
  matched = NULL,
  update_push_log = FALSE,
  since = NULL,
  sensor_decoding = NULL,
  out_dir = getwd(),
  rmd = FALSE
)
```

Arguments

`matched`	This argument also accepts a character vector of file paths of your matched detections. These can be CSVs or zipped folders.
`update_push_log`	Do you wish to use an updated push log? Default is FALSE, but switch to TRUE if you haven't updated this package since the push occurred.
`since`	date in YYYY-MM-DD format. Provides a summary of detections that were matched/edited since that date.

sensor_decoding
: Not yet implemented. Will be a place to provide information to decode and summarize sensor data,

out_dir
: Defaults to working directory. In which directory would you like to save the report?

rmd
: Logical. Compile via RMarkdown rather than Quarto?

Examples

```
## Not run:
# The code below downloads some matched detections from OTN, then calls the function.
td <- file.path(tempdir(), "matos_test_files")
dir.create(td)

download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/",
    "pbsm/detection-extracts/pbsm_matched_detections_2018.zip/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm_matched_detections_2018.zip"),
  mode = "wb"
)
unzip(file.path(td, "pbsm_matched_detections_2018.zip"),
  exdir = td
)

# Provide the detection file(s) to the \code{matched} argument, with an
# optional date to the \code{since} argument to summarize what was new since
# that date.
make_tag_push_summary(
  matched = file.path(
    td,
    "pbsm_matched_detections_2018.csv"
  ),
  since = "2018-11-01"
)

## End(Not run)
```

---

matched_abacus *Create an abacus plot of matched detections*

Description

Create an abacus plot of matched detections

Usage

```
matched_abacus(temp_dist, release)
```

Arguments

temp_dist	Data from the output of temporal_distribution()
release	Data frame of release times/locations; a subset of the matched detections data

Examples

```
## Not run:
# Get a detection file
download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/",
    "pbsm/detection-extracts/pbsm_matched_detections_2018.zip/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm_matched_detections_2018.zip"),
  mode = "wb"
)
unzip(file.path(td, "pbsm_matched_detections_2018.zip"),
  exdir = td
)

matched_dets <- data.table::fread(
  file.path(td, "pbsm_matched_detections_2018.csv")
)


# Run temporal_distribution
temporal <- temporal_distribution(matched_dets, "tag")

# Run matched_abacus
matched_abacus(temporal$data, matched_dets[receiver == "release"])

## End(Not run)
```

---

match_map *Plot the geographic extent of OTN projects*

Description

Plot the geographic extent of OTN projects

Usage

```
match_map(otn_tables)
```

Arguments

otn_tables	A list containing OTN's otn_resources_metadata_points GeoServer layer. Usually created using otn_query.

Examples

```
match_map(
  otn_query("MDWEA")
)
```

`match_table` *Create a reactable table of matched detections*

Description

Create a reactable table of matched detections

Usage

```
match_table(extract, type = c("tag", "receiver"))
```

Arguments

`extract`	matched (transmitter) or qualified (receiver) OTN detections
`type`	Tag or receiver data? Takes values of "tag" and "receiver"; defaults to "tag".

Examples

```
## Not run:
# Receiver
download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/pbsm/",
    "detection-extracts/pbsm_qualified_detections_2018.zip/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm_qualified_detections_2018.zip"),
  mode = "wb"
)
unzip(
  file.path(td, "pbsm_qualified_detections_2018.zip"),
  exdir = td
)

qualified_dets <- data.table::fread(
  file.path(td, "pbsm_qualified_detections_2018.csv")
)

match_table(
  extract = qualified_dets,
  type = "receiver"
)
```

```
# Transmitters
download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/",
    "pbsm/detection-extracts/pbsm_matched_detections_2018.zip/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm_matched_detections_2018.zip"),
  mode = "wb"
)
unzip(file.path(td, "pbsm_matched_detections_2018.zip"),
  exdir = td
)

matched_dets <- data.table::fread(
  file.path(td, "pbsm_matched_detections_2018.csv")
)

match_table(
  extract = matched_dets,
  type = "tag"
)

## End(Not run)
```

`otn_query` *Query the OTN Geoserver*

Description

Query the OTN Geoserver

Usage

```
otn_query(projects)
```

Arguments

`projects`	Character vector of OTN project codes for which you'd like project metadata. Prepended networks can be provided, but are not necessary.

Value

list of the "otn_resources_metadata_points" and "project_metadata" for the given projects

Examples

```
otn_query(c("EST", "FACT.SCDNRDFP", "ACT.MDBSB", "MDBSB"))
```

prep_match_table *Prepare the detection match summary data*

Description

Prepare the detection match summary data

Usage

```
prep_match_table(extract, type = c("tag", "receiver"))
```

Arguments

extract	matched (transmitter) or qualified (receiver) OTN detections
type	Tag or receiver data? Takes values of "tag" and "receiver"; defaults to "tag".

prep_station_spatial *Summarize OTN extract data by station and convert to a spatial object*

Description

Summarize OTN extract data by station and convert to a spatial object

Usage

```
prep_station_spatial(extract, type = c("tag", "receiver"))
```

Arguments

extract	OTN extract data
type	type of extract data: "tag" or "receiver"

Examples

```
## Not run:
# Get an extract file
download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/",
    "pbsm/detection-extracts/pbsm_matched_detections_2018.zip/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm_matched_detections_2018.zip"),
  mode = "wb"
)
unzip(file.path(td, "pbsm_matched_detections_2018.zip"),
```

```
  exdir = td
)

matched_dets <- data.table::fread(
  file.path(td, "pbsm_matched_detections_2018.csv")
)

# Convert to spatial
prep_station_spatial(matched_dets, "tag")

## End(Not run)
```

`project_contacts` *Extract and combine the contacts for matched projects*

Description

Extract and combine the contacts for matched projects

Usage

```
project_contacts(extract, type = c("receiver", "tag"))
```

Arguments

`extract`	data.frame of transmitter/receiver detections matched by OTN: matched detections for tags and qualified detections for receivers
`type`	Type of extract data: "tag" or "receiver"

Value

a data.table containing project names, principal investigators (PI), points of contact (POC), and their respective emails. Multiple emails are separated by commas.

Examples

```
## Not run:
# Set up example data
td <- file.path(tempdir(), "otndo_example")
dir.create(td)

download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/",
    "pbsm/detection-extracts/pbsm_matched_detections_2018.zip/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm_matched_detections_2018.zip"),
```

```
  mode = "wb"
)
unzip(file.path(td, "pbsm_matched_detections_2018.zip"),
  exdir = td
)

matched <- read.csv(file.path(
  td,
  "pbsm_matched_detections_2018.csv"
))

# Actually run the function
project_contacts(matched, type = "tag")

# Clean up
unlink(td, recursive = TRUE)

## End(Not run)
```

`remaining_transmitters` *Estimate transmitters remaining in the system*

Description

This function estimates the transmitters remaining in the system by finding the last date of detection for each transmitter and summing all available transmitters in a given day. This is a very coarse measure and likely to be very inaccurate with sparse data or short time scales.

Usage

```
remaining_transmitters(matched, push_log, release = NULL)
```

Arguments

`matched`	matched OTN transmitter detections
`push_log`	data.frame containing the date of the most-recent data push. This requirement is very likely to change in the future.
`release`	Optional. Data frame of release times/locations; a subset of the matched detections data

Examples

```
## Not run:
#' # Set up example data
td <- file.path(tempdir(), "otndo_example")
dir.create(td)
```

```
# For tag data
download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/",
    "pbsm/detection-extracts/pbsm_matched_detections_2018.zip/@download/file"
  ),
  destfile = file.path(td, "pbsm_matched_detections_2018.zip")
)
unzip(file.path(td, "pbsm_matched_detections_2018.zip"),
  exdir = td
)

matched <- read.csv(file.path(
  td,
  "pbsm_matched_detections_2018.csv"
))

# Run remaining_transmitters()
remaining_transmitters(matched_dets, data.frame(date = as.Date("2020-01-01")))

## End(Not run)
```

`station_table` *Create the station summary table*

Description

Create the station summary table

Usage

```
station_table(extract, type = c("tag", "receiver"))
```

Arguments

`extract`	OTN detections. "Matched" detections for tag data and "qualified" detections for receiver data
`type`	type of data to be summarized.

Value

For tag data, a data.table with the PI, project, station, number of detections, and number of individuals heard. For receiver data, a data.table with the station, number of detections, and number of individuals heard (assuming that the PI and POC is you).

Examples

```
## Not run:
# Set up example data
td <- file.path(tempdir(), "otndo_example")
dir.create(td)

# For tag data
download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/",
    "pbsm/detection-extracts/pbsm_matched_detections_2018.zip/@download/file"
  ),
  destfile = file.path(td, "pbsm_matched_detections_2018.zip")
)
unzip(file.path(td, "pbsm_matched_detections_2018.zip"),
  exdir = td
)

matched <- read.csv(file.path(
  td,
  "pbsm_matched_detections_2018.csv"
))

# Actually run the function
prep_station_table(matched, type = "tag")



# For receiver data
download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/pbsm/",
    "detection-extracts/pbsm_qualified_detections_2018.zip/",
    "@download/file"
  ),
  destfile = file.path(td, "pbsm_qualified_detections_2018.zip"),
  mode = "wb"
)
unzip(
  file.path(td, "pbsm_qualified_detections_2018.zip"),
  exdir = td
)

qualified <- read.csv(file.path(td, "pbsm_qualified_detections_2018.csv"))

# Actually run the function
station_table(qualified, type = "receiver")

# Clean up
unlink(td, recursive = TRUE)

## End(Not run)
```

`temporal_distribution` *Create an abacus plot of detections by project*

Description

Create an abacus plot of detections by project

Usage

```
temporal_distribution(extract, type = c("tag", "receiver"))
```

Arguments

`extract`	OTN data extract file
`type`	Transmitter (tag) or receiver detections?

Examples

```
## Not run:
# Set up example data
td <- file.path(tempdir(), "otndo_example")
dir.create(td)

# For tag data
download.file(
  paste0(
    "https://members.oceantrack.org/data/repository/",
    "pbsm/detection-extracts/pbsm_matched_detections_2018.zip/@download/file"
  ),
  destfile = file.path(td, "pbsm_matched_detections_2018.zip")
)
unzip(file.path(td, "pbsm_matched_detections_2018.zip"),
  exdir = td
)

matched <- read.csv(file.path(
  td,
  "pbsm_matched_detections_2018.csv"
))

temporal_distribution(matched, "tag")

## End(Not run)
```

Index