Package 'filteRjsats'

April 10, 2023

Title Filter Raw JSATS Acoustic Telemetry Files from Lotek, ATS, Teknologic

Version 1.0

Description Filtering of raw acoustic telemetry transmissions from three acoustic telemetry companies (ATS, Lotek, Teknologic). The filtering steps check for false positives caused by reflected transmissions from surfaces and false pings from other noise generating equipment. The filter is unique for each technology type. The package was written in concert with the Interagency Telemetry Advisory Group (iTAG) and makes use of the JSATS California Fish Tracking Database: <https://oceanview.pfeg.noaa.gov/CalFishTrack/>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Imports broman, dplyr, lubridate, stringr, tidyr, rerddap

Suggests knitr, rmarkdown

Depends R (>= 4.1)

LazyData true

LazyDataCompression xz

VignetteBuilder knitr

NeedsCompilation no

Author Taylor Spaulding [aut, cre] (<https://orcid.org/0000-0003-4381-5296>), Arnold Ammann [aut], Travis Hinkelman [ctb]

Maintainer Taylor Spaulding <tspaulding@esassoc.com>

Repository CRAN

Date/Publication 2023-04-10 14:30:02 UTC

R topics documented:

add_fish . . . 3
cft_fish . . . 3
cft_rcvrs . . . 5
compare_detects . . . 6
filtered_detections . . . 7
filter_fish_detects . . . 7
fish . . . 9
fish_detects . . . 10
format_detects . . . 10
format_fish . . . 12
format_receivers . . . 13
get_fish_fields . . . 14
get_rcvr_data . . . 15
get_rcvr_fields . . . 15
get_receiver_type . . . 16
get_reference_tags . . . 16
get_tagged_fish . . . 17
join_rcvr_data . . . 18
prefilter . . . 18
prefiltered_detects . . . 19
raw_ats . . . 20
rcvr_fields . . . 20
read_ats . . . 21
read_jsats . . . 21
read_lotek . . . 22
read_tekno . . . 23
receivers . . . 23
reftags . . . 24
rolling_sd_3 . . . 25
second_filter_2h4h . . . 25
second_filter_4h . . . 26
second_filter_ats . . . 26
second_filter_ats_4h . . . 27
second_filter_lotek . . . 28
second_filter_tekno . . . 28
second_filter_tekno_4h . . . 29
Set_GVs . . . 30

Index **31**

add_fish *Add Fish Data from ERDAPP to a Detection Dataframe*

Description

This function takes a prefiltered detection dataframe from prefilter() and joins it to fish data downloaded from ERDAPP using the get_fish_data() function or formatted using the format_fish() function. Detections are then filtered further based on the date and time of tag release and expected battery life. Detections occurring before release of the tag or after 2x the expected battery life are removed.

Usage

```
add_fish(prefilter_file, fish)
```

Arguments

prefilter_file	a prefiltered detection dataframe from prefilter()
fish	a dataframe of fish data rectrieved from get_fish_data()

Value

A filtered dataframe converting the raw detection data into rows of detections

Examples

```
# Add fish data to a detection dataframe and filter for only detections
# associated with tagged fish

add_fish(prefiltered_detects, cft_fish)
```

cft_fish *Example fish data from CalFishTrack*

Description

A dataframe of acoustically tagged fish downloaded from ERDDAP representing fish released in 2021 and 2022.

Usage

```
cft_fish
```

Format

`cft_fish`:

A dataframe with 17,227 rows and 25 columns

fish_id Identifies the fish that was tagged. It is unique in that no two fish have the same FishID. Format is 2 or more letters that describe the type of the fish (e.g. WR for WinterRun), followed by the year (YYYY), followed by a dash, then a sequential three digit number (e.g. 001 to 999)

study_id Identifies a group of tagged fish that belong to a study. Format is text description of place followed by year (YYYY). A single StudyID can have fish release on multiple days within a year and/or at multiple locations within a year.

fish_type Describes the fish tagged. Generally a part that describes where it came from and a part that refers to the common name.

fish_origin Describes where the fish is from or where it was collected. Example: Hatchery, Natural, Sacramento River.

fish_date_tagged The date and time that indicates the time the tag was activated and implanted into the fish. Tags are usually activated several minutes before implanting into fish in PST.

fish_release_date The date and time of release in PST.

tag_id_hex The hexadecimal form for the tags code. This is not unique in that two fish can have the same Hex Tag Code. This is usually from the same tag code being used in different years. This is why all queries must be based on Fish ID.

tag_id_decimal The decimal form for the tags code. This is not unique in that two fish can have the same Decimal Tag Code. This is usually from the same tag code being used in different years. This is why all queries must be based on Fish ID.

tag_weight weight of tag in air

tag_model The model of the tag.

tag_pulse_rate_interval_nominal The nominal (aka approximate) pulse rate interval. This is how often the tag transmits its code signal.

tag_warranty_life The minimum number of days a tag is expected to transmit its code. Generally tags transmit for at least 1.5x the warranty life.

fish_length_type The way the fish was measured. Fork length = FL, total length = TL, standard length = SL. If unknown = NA.

fish_length Measured fish length in millimeters

fish_weight Measured fish weight in grams (in air)

release_location The name of the place that the fish was released.

release_latitude The latitude of the release location.

release_longitude The longitude of the release location.

release_river_km The river km of the release location. The Golden Gate Bridge = rkm 0.0. Values increase the further upstream.

email the email address of the point of contact that grants approval for using the data from each fish.

`cft_rcvrs` *Example receiver data from CalFishTrack*

Description

A dataframe of acoustic receiver metadata downloaded from ERDDAP representing receivers deployed from 2021-2022.

Usage

```
cft_rcvrs
```

Format

`cft_rcvrs`:

A dataframe of 1,130 rows and 12 columns:

dep_id a unique number that identifies a single deployment

receiver_serial_number the serial number of the acoustic receiver that recorded the detection. If this value is 1, then the record is not a receiver detection but is the release date-time and location. Every tagged fish has at least one record in the table

receiver_general_location A name for the geographic location of one or more receivers.

receiver_region A name for a larger geographic region that defines a subarea of the Central Valley watershed. Some regions are Upper Sac R, Lower Sac R, Feather R, East Delta, West Delta, SF Bay.

receiver_location The name for the location of a single receiver

latitude Latitude for the exact location of a single receiver. If unknown = cell is blank

longitude Longitude for the exact location of a single receiver. If unknown = cell is blank

receiver_river_km River kilometer for the location of a single receiver. Must be unique for each GPSname. The Golden Gate Bridge is rkm 0. Values increase the further upstream

receiver_make Describes the manufacturer and type of JSATS receiver.

receiver_depth the estimated depth of the receiver hydrophone. If unknown = blank

receiver_start The date and time in Pacific Standard Time when the receiver was deployed (put at the site in the water). Format: date time M/DD/YYYY HH:MM

receiver_end The date and time in Pacific Standard Time when the receiver was recovered (removed from the water). Format: date time M/DD/YYYY HH:MM

receiver_last_valid Taken from the detection data file for this deployment, the date and time in Pacific Standard Time of the last valid detection or sensor data entry. This time cannot be greater than the EndTime, but can be less that the EndTime if receiver quit recording data or was pulled out of the water. Format: date time M/DD/YYYY HH:MM

receiver_data_coverage describes how extensive the data is for this receiver deployment. If the receiver failed to work or was lost and not recovered = none. If it worked for part of the deployment = partial. If the receiver worked the entire deployment = full. Can be blank.

receiver_coverage_problem If the receiver did not work until EndTime then an entry here describes the reason why the receiver may have failed.

receiver_agency The agency (abbreviated) managing the receiver deployment.

receiver_beacon_id_hex The hex code of the beacon tag associated with this receiver deployment

receiver_beacon_id_dec The decimal code of the beacon tag associated with this receiver deployment

receiver_beacon_pri The pulse rate interval of the beacon tag in secs, usually 60 or 30 secs

---

`compare_detects` *Compare two Detection Dataframes*

Description

This function compares two detection dataframes from filtered acoustic telemetry detections and identifies the filtered out detections. The first input is the original dataframe, the second is the filtered dataframe. Then it creates a dataframe of detections which are missing from the filtered dataframe.

Usage

```
compare_detects(a1, a2)
```

Arguments

a1	unfiltered detection dataframe
a2	filtered detection dataframe

Value

A dataframe of detections from the original dataframe missing from the filtered dataframe. Spurious detections.

Examples

```
# Compare detections before and after filtering
compare_detects(raw_ats[1:1000,], filter_fish_detects[1:1000,])
```

filtered_detections *Previously filtered detection data*

Description

An example dataset of real acoustic telemetry detections of fish at several receivers within the California Central Valley from 2021. These detections have already been been processed to remove false positives using the various filtering functions in this package and in companion package `filteRjsats`.

Usage

```
filtered_detections
```

Format

`filtered_detections`**:**

A data frame with 100,000 rows and 3 columns:

serial The serial number of the detecting receiver

local_time the local time of the detection (tz = America/Los_Angeles)

tag_id The hexadecimal acoustic tag ID code

Source

Data collected by the California Department of Water Resources 2021

filter_fish_detects *Example fish detection data which has been prefiltered and has fish data*

Description

Example tag detection data representing a single file which has been processed using the `prefilter()` and `add_fish()` functions.

Usage

```
filter_fish_detects
```

Format

`filter_fish_detects`**:**

A dataframe with 5,000 rows and 39 columns

ReceiverSN the serial number of the acoustic receiver which heard the detection

Make The Manufacturer of the acoustic reciever

DateTime_Local The local date and time of the detection tz = "America/Los_Angeles"

Tag_Decimal The decimal value of the tag ID code

Tag_Hex The hexadecimal value of the tag ID code

Tilt The tilt of the acoustic receiver from a vertical axis

Volt The voltage of the on-board battery of the receiver

Temp The water temperature outside of the receiver

SigStr The strength of the acoustic signal in dB

Freq The frequency of the acoustic signal in kHz

Thres The relative amount of external background noise, signal threshold

CheckMBP A calculated field from the first filter checking the time between acoustic transmissions from the same tag was >0.3secs

TagInFile A calculated field from the add_fish filter which queries whether the tag code of the detection is associated with a fish.

fish_id Identifies the fish that was tagged. It is unique in that no two fish have the same FishID. Format is 2 or more letters that describe the type of the fish (e.g. WR for WinterRun), followed by the year (YYYY), followed by a dash, then a sequential three digit number (e.g. 001 to 999)

study_id Identifies a group of tagged fish that belong to a study. Format is text description of place followed by year (YYYY). A single StudyID can have fish release on multiple days within a year and/or at multiple locations within a year.

fish_type Describes the fish tagged. Generally a part that describes where it came from and a part that refers to the common name.

fish_origin Describes where the fish is from or where it was collected. Example: Hatchery, Natural, Sacramento River.

fish_date_tagged The date and time that indicates the time the tag was activated and implanted into the fish. Tags are usually activated several minutes before implanting into fish in PST.

fish_release_date The date and time of release in PST.

tag_id_hex The hexadecimal form for the tags code. This is not unique in that two fish can have the same Hex Tag Code. This is usually from the same tag code being used in different years. This is why all queries must be based on Fish ID.

tag_id_decimal The decimal form for the tags code. This is not unique in that two fish can have the same Decimal Tag Code. This is usually from the same tag code being used in different years. This is why all queries must be based on Fish ID.

tag_weight weight of tag in air

tag_model The model of the tag.

tag_pulse_rate_interval_nominal The nominal (aka approximate) pulse rate interval. This is how often the tag transmits its code signal.

tag_warranty_life The minimum number of days a tag is expected to transmit its code. Generally tags transmit for at least 1.5x the warranty life.

fish_length_type The way the fish was measured. Fork length = FL, total length = TL, standard length = SL. If unknown = NA.

fish_length Measured fish length in millimeters

fish_weight Measured fish weight in grams (in air)

release_location The name of the place that the fish was released.

release_latitude The latitude of the release location.

release_longitude The longitude of the release location.

release_river_km The river km of the release location. The Golden Gate Bridge = rkm 0.0. Values increase the further upstream.

email the email address of the point of contact that grants approval for using the data from each fish.

release_rkm The river km of the release location. The Golden Gate Bridge = rkm 0.0. Values increase the further upstream.

tag_life The minimum number of days a tag is expected to transmit its code. Generally tags transmit for at least 1.5x the warranty life.

length Measured fish length in millimeters

weight Measured fish weight in grams (in air)

CheckDT A calculated field which checks whether the detection occurred after the release of the fish

CheckBattLife A calculated field which checks whether the detection occurred before the tag battery is expected to expire (2x tag life)

`fish` *Fish Data*

Description

An example dataset of real fish tagged with acoustic telemetry tags and released within the California Central Valley in 2021 and 2022.

Usage

```
fish
```

Format

`fish`**:**

A data frame with 17,227 rows and 10 columns:

fish_type Generally a strain, run, and species of fish (e.g. Nimbus Fall Chinook = Fall-run Chinook Salmon from Nimbus Hatchery)

TagCode The hexadecimal code of the implanted acoustic tag

Release_Date The release date and time of the fish

release_location The coded name of the release site

length The length of the fish in millimeters

weight The weight of the fish in grams

tag_weight The weight of the implanted acoustic tag

tag_model The model number of the implanted acoustic tag

PRI The pulse rate interval (time between transmissions) of the implanted tag, as reported by the manufacturer

TagLife The expected number of days the tag should continue to transmit, as reported by the manufacturer

Source

https://oceanview.pfeg.noaa.gov/CalFishTrack/pageRealtime_download.html

---

`fish_detects` *Example fish detection data*

Description

Example tag detection data from CalFishTrack representing fish released in 2021.

Usage

```
fish_detects
```

Format

`fish_detects`**:**

A dataframe with 10,000 rows and 3 columns

ReceiverSN the serial number of the acoustic receiver which heard the detection

DateTime_Local The local date and time of the detection tz = "America/Los_Angeles"

Tag_Code The hexadecimal value of the acoustic tag ID code

`format_detects` *Format Detections for filteRjsats*

Description

This function takes a detection dataframe from a single receiver and reformats specific columns so that they can be read by the filtering functions in filteRjsats package

Usage

```
format_detects(
  data,
  var_Id,
  var_datetime_local,
  var_frequency = NULL,
  var_receiver_serial,
  var_receiver_make = NULL,
  local_time_zone,
  time_format
)
```

Arguments

data	the detection dataframe with columns for individual receivers, tag IDs,datetime, and the expected ping rate.
var_Id	the column name, in quotes, which identifies the individual transmitter/tag/organism identifier.
var_datetime_local	the column name, in quotes, which identifies the date and time of the detection event. This column should already have been converted to POSIXct format and should be converted to the local timezone.
var_frequency	the column name, in quotes, which identifies the maximum temporal frequency at which transmitters in organisms emit a detectable signal, only for use before JSATS filtering.
var_receiver_serial	the column name, in quotes, which identifies the serial number of the detection receiver
var_receiver_make	the column name, in quotes, which identifies the make or brand of the detection receiver. Must be one of "ATS", "Lotek", or "Tekno", only for use before JSATS filtering.
local_time_zone	the local timezone used for analyses. Uses tz database names (e.g. "America/Los_Angeles" for Pacific Time)
time_format	a string value indicating the datetime format of all time fields

Value

A standardized detection dataframe which can be read by filteRjsats

Examples

```
# Rename columns to work with functions
format_detects(data = filtered_detections,
               var_Id = "tag_id",
               var_datetime_local = "local_time",
               var_receiver_serial = "serial",
```

```
local_time_zone = "America/Los_Angeles",
time_format = "%Y-%m-%d %H:%M:%S")
```

`format_fish` *Format Fish Data for add_fish()*

Description

This function takes a dataframe of fish and tag data and renames the columns to those expected by the add_fish() function

Usage

```
format_fish(
  data,
  var_Id,
  var_release,
  var_tag_life,
  var_ping_rate,
  local_time_zone,
  time_format
)
```

Arguments

`data`	a dataframe of fish and tag data
`var_Id`	the column name, in quotes, which identifies the individual transmitter/tag/organism identifier.
`var_release`	the column name, in quotes, which identifies the release date and time in POSIX format and appropriate timezone
`var_tag_life`	the column name, in quotes, which identified the expected tag life in days
`var_ping_rate`	the column name, in quotes which identifies the expected ping rate of the tag/transmitter
`local_time_zone`	the local timezone used for analyses. Uses tz database names (e.g. "America/Los_Angeles" for Pacific Time)
`time_format`	a string value indicating the datetime format of all time fields

Value

A dataframe which contains fields renamed to match those required by add_org() function

Examples

```
# Rename columns to work with functions
format_fish(data = fish,
            var_Id = "TagCode",
            var_release = "Release_Date",
            var_tag_life = "TagLife",
            var_ping_rate = "PRI",
            local_time_zone = "America/Los_Angeles",
            time_format = "%Y-%m-%d %H:%M:%S")
```

`format_receivers` *Format for Receiver data for filteRjsats*

Description

This function takes a dataframe of receiver metadata and reformats specific columns so that they can be read by the filtering functions in filteRjsats package

Usage

```
format_receivers(
  data,
  var_receiver_serial,
  var_receiver_make,
  var_receiver_deploy,
  var_receiver_retrieve,
  local_time_zone,
  time_format
)
```

Arguments

`data`	the detection dataframe with columns for individual receivers, tag IDs,datetime, and the expected ping rate.
`var_receiver_serial`	the column name, in quotes, which identifies the serial number of the detection receiver
`var_receiver_make`	the column name, in quotes, which identifies the make or brand of the detection receiver. Must be one of "ATS", "Lotek", or "Tekno"
`var_receiver_deploy`	the column name, in quotes, which identifies the date and time the receiver was deployed
`var_receiver_retrieve`	the column name, in quotes, which identifies the date and time the receiver was retrieved

local_time_zone
: the local timezone used for analyses. Uses tz database names (e.g. "America/Los_Angeles" for Pacific Time)

time_format
: a string value indicating the datetime format of all time fields

Value

A dataframe which contains fields renamed to match those required by add_receivers() function

Examples

```
# Rename columns to work with functions
format_receivers(data = receivers,
                 var_receiver_serial = "receiver_serial_number",
                 var_receiver_make = "receiver_make",
                 var_receiver_deploy = "receiver_start",
                 var_receiver_retrieve = "receiver_end",
                 local_time_zone = "America/Los_Angeles",
                 time_format = "%m-%d-%Y %H:%M:%S")
```

---

get_fish_fields *Get A List of Fish Related Fields from ERDAPP*

Description

This function searches the California Fish Tracking ERDAPP Database to create a list of all potential fields related to fish metrics. Used to identify important fish data fields to add to detection data in the add_fish function.

Usage

```
get_fish_fields()
```

Value

A vector of potential fish fields which the user may review to retrieve specific field indices

Examples

```
fish_fields <- get_fish_fields
```

get_rcvr_data *Get A Dataframe of All Receiver Data from CalFishTrack*

Description

This function searches the California Fish Tracking ERDAPP Database to create a dataframe ofacoustic receiver metadata. Desired fields can be set to select only certain fields, object rcvr_fields is the default. This data is used to add receiver metadata to detection data in the add_fish function.

Usage

```
get_rcvr_data(fields = rcvr_fields)
```

Arguments

fields	a vector of important field names to be imported from the ERDDAP dataset

Value

A dataframe of receiver metadata which can be joined to detection data

Examples

```
# Retrieve the default set of fields from Cal Fish Track needed for package
tout <- getOption("timeout")
options(timeout = 4)
try(get_rcvr_data())
options(timeout = tout)
```

get_rcvr_fields *Get A List of Receiver Related Fields from ERDAPP*

Description

This function searches the California Fish Tracking ERDAPP Database to create a list of all potential fields related to acoustic receiver metadata. Used to identify important metadata fields to include when adding receiver data in the join_rcvr_data function.

Usage

```
get_rcvr_fields()
```

Value

A vector of potential receiver metadata fields which the user may review

Examples

```
# View a list of available receiver fields
get_rcvr_fields()
```

`get_receiver_type` *Get Receiver Type*

Description

This function takes a list of file paths to raw acoustic receiver detection files supplied by the user and identifies the type of acoustic receiver technology (Lotek, ATS, Teknologic) used to generate the file.

Usage

```
get_receiver_type(file_list)
```

Arguments

`file_list`	A vector of raw acoustic detection file paths

Value

A dataframe containing the file path, file name, and technology type

Examples

```
# get a list of files
files <- system.file("extdata", package = "filteRjsats")|> list.files()

# get the receiver type
get_receiver_type(files)
```

`get_reference_tags` *Get A List of Reference (Beacon) Tags from ERDAPP*

Description

This function searches the California Fish Tracking ERDAPP Database to create a list of beacon tag hexadecimal IDs. Used in the prefilter to separate beacon tags from tagged fish.

Usage

```
get_reference_tags()
```

Value

A vector of beacon tags hexadecimal IDs

Examples

```
# Download reference tags from CalFishTrack
tout <- getOption("timeout")
options(timeout = 4)
try(ref_tags <- get_reference_tags())
options(timeout = tout)
```

`get_tagged_fish` *Get A Dataframe of All Tagged fish from CalFishTrack*

Description

This function searches the California Fish Tracking ERDAPP Database to create a dataframe of tagged fish data. Desired fields can be set to select only certain fields. This data is used to add fish data to detection data in the add_fish function.

Usage

```
get_tagged_fish(important_fields = NULL)
```

Arguments

`important_fields`
: a vector of important field indexes to be imported from the ERDDAP dataset

Value

A dataframe of fish data which can be joined to detection data

Examples

```
# Retrieve only a few important fields (fish type, tag code, release date)
tout <- getOption("timeout")
options(timeout = 4)
fields <- c(7,8,16)
try(cal_fish_lite <- get_tagged_fish(important_fields = fields))
options(timeout = tout)
```

join_rcvr_data *Add Receiver Data to a Completely Filtered Detection File*

Description

This function takes any acoustic receiver detection dataframe which contains receiver serial numbers in a ReceiverSN field, such as that produced by second_filter_*(), and joins it with receiver metadata generated by the get_rcvr_data() or format_receiver() functions.

Usage

```
join_rcvr_data(final_file, rcvr_data)
```

Arguments

final_file	a dataframe of detections retrieved from second_filter
rcvr_data	a dataframe of receiver metadata retrieved from get_rcvr_data

Value

A dataframe to which receiver metadata has been added.

Examples

```
# Join receiver metadata to detection data
join_rcvr_data(fish_detects, cft_rcvrs)
```

prefilter *Apply the "prefilter" to a Detection Dataframe*

Description

This function takes a detection dataframe output from read_jsats or from format_detects and filters out multipath signals (time between detections < 0.3 seconds) and spurious signals which do not occur within a specified time frame of the last detection. For beacon tags, this time frame is 192 seconds, and for fish, this time is 120 seconds. Following this, the dataframes are standardized so that all detection dataframes from any technology type are identical and superfluous fields are removed.

Usage

```
prefilter(jsats_file, reference_tags)
```

Arguments

jsats_file	A dataframe which is the output from read_jstats() or format_detects()
reference_tags	A vector of potential reference (beacon) tag IDs

Value

A standardized detection dataframe with multipath detects removed

Examples

```
# Filter a raw detection dataset
prefilter(raw_ats, reftags)
```

---

prefiltered_detects *Example Prefiltered Detection Dataframe*

Description

A dataframe of detections which has gone through the `prefilter()` function

Usage

```
prefiltered_detects
```

Format

`prefiltered_detects`**:**

A dataframe with 209,149 rows and 12 columns

ReceiverSN the serial number of the acoustic receiver which heard the detection

Make The Manufacturer of the acoustic reciever

DateTime_Local The local date and time of the detection tz = "America/Los_Angeles"

Tag_Decimal The decimal value of the tag ID code

Tag_Hex The hexadecimal value of the tag ID code

Tilt The tilt of the acoustic receiver from a vertical axis

Volt The voltage of the on-board battery of the receiver

Temp The water temperature outside of the receiver

SigStr The strength of the acoustic signal in dB

Freq The frequency of the acoustic signal in kHz

Thres The relative amount of external background noise, signal threshold

CheckMBP A calculated field from the first filter checking the time between acoustic transmissions from the same tag was >0.3secs

raw_ats *Example raw data from ats receiver after being read in by read_jsats*

Description

A dataframe of ATS acoustic receiver data processed by `read_jsats()`

Usage

```
raw_ats
```

Format

`raw_ats`**:**
a dataframe with 261,648 rows and 11 columns:

ReceiverSN the serial number of the acoustic receiver which heard the detection
Make The Manufacturer of the acoustic reciever
DateTime_Local The local date and time of the detection tz = "America/Los_Angeles"
Tag_Decimal The decimal value of the tag ID code
Tag_Hex The hexadecimal value of the tag ID code
Tilt The tilt of the acoustic receiver from a vertical axis
Volt The voltage of the on-board battery of the receiver
Temp The water temperature outside of the receiver
SigStr The strength of the acoustic signal in dB
Freq The frequency of the acoustic signal in kHz
Thres The relative amount of external background noise, signal threshold

rcvr_fields *A List of Important Receiver Related Fields from ERDAPP*

Description

A vector of acoustic receiver metadata fields which are needed, at a minimum, to run the join_rcvr_data function

Usage

```
rcvr_fields
```

Format

A vector of potential receiver metadata fields which the user may review

read_ats *Read an ATS Acoustic Receiver File*

Description

This function takes a raw acoustic detection file generated by an ATS JSATS receiver and processes it into a dataframe which can be used by the filtering functions in this package.This is called within read_jsats().

Usage

```
read_ats(path, file, timezone = "America/Los_Angeles")
```

Arguments

path	the path to the folder containing the desired file
file	the path of the desired file
timezone	the Olsen Named time zone, default is "America/Los_Angeles"

Value

A dataframe converting the raw detection data into rows of detections

Examples

```
# Read in an ATS file
path = system.file("extdata/", package = "filteRjsats")
read_ats(path = path, file = "17111___210531_115356.csv",
timezone = "America/Los_Angeles")
# Warnings are expected due to the formatting of ATS files
```

read_jsats *Read an Acoustic Receiver File*

Description

This function takes a raw acoustic detection file generated by a Lotek, Teknologic, or ATS JSATS receiver and determines which reader function to use to process it into a dataframe which can be used by the filtering functions in this package.

Usage

```
read_jsats(path, file, timezone = "America/Los_Angeles")
```

Arguments

path	the path to the folder containing the desired file
file	the path of the desired file
timezone	the Olsen Named time zone, default is "America/Los_Angeles"

Value

A dataframe converting the raw detection data into rows of detections

Examples

```
# Read in any ATS, LOTEK, TEKNO file

# list of files
path = system.file("extdata", package = "filteRjsats")
files <- list.files(path)

# Read the files
raw_data <- list()
for(i in 1:length(files)){
raw_data[[i]] <- read_jsats(path = path, file = files[i],
timezone = "America/Los_Angeles")
}
# Warnings are expected due to the formatting of ATS files
```

---

read_lotek *Read a Lotek Acoustic Receiver File*

Description

This function takes a raw acoustic detection file generated by a Lotek JSATS receiver and processes it into a dataframe which can be used by the filtering functions in this package.This is called within read_jsats().

Usage

```
read_lotek(path, file, timezone = "America/Los_Angeles")
```

Arguments

path	the path to the folder containing the desired file
file	the path of the desired file
timezone	the Olsen Named time zone, default is "America/Los_Angeles"

Value

A dataframe converting the raw detection data into rows of detections

Examples

```
# see read_ats or read_tekno for example usage
```

`read_tekno` *Read a Teknologic Acoustic Receiver File*

Description

This function takes a raw acoustic detection file generated by a Teknologic JSATS receiver and processes it into a dataframe which can be used by the filtering functions in this package. This is called within read_jsats().

Usage

```
read_tekno(path, file, timezone = "America/Los_Angeles")
```

Arguments

`path`	the path to the folder containing the desired file
`file`	the path of the desired file
`timezone`	the Olsen Named time zone, default is "America/Los_Angeles"

Value

A dataframe converting the raw detection data into rows of detections

Examples

```
# Read in a Teknologic file
path = system.file("extdata/", package = "filteRjsats")
read_tekno(path = path, file = "2015-6007211361217.SUM",
timezone = "America/Los_Angeles")
```

`receivers` *Receiver Data*

Description

An example dataset of real acoustic telemetry receivers within the California Central Valley in 2021.
These receivers are only those which match the serial numbers in companion dataset `filtered_detections`.
This data is formatted to match the California Fish Tracking receiver metadata found here: https://oceanview.pfeg.noaa.gov/Ca

Usage

```
receivers
```

Format

`receivers`:

A data frame with 52 rows and 10 columns:

dep_id A unique id is created for each receiver deployment

receiver_make The brand of the acoustic receiver

receiver_serial_number The serial number of the acoustic receiver

latitude The decimal degree latitude (WGS1984) of the acoustic receiver at deployment

longitude The decimal degree longitude (WGS1984) of the acoustic receiver at deployment

receiver_location The site name of an individual receiver, often more than one `receiver_location` is found at a `receiver_general_location`

receiver_general_location The more general geographic name of the location of the receiver

receiver_river_km The number of river kilometers the receiver is from the Golden Gate Bridge

receiver_start The start time of the reciever (generally when it was deployed)

receiver_end The end time of the receiver (generally when it was retrieved)

Source

https://oceanview.pfeg.noaa.gov/CalFishTrack/pageRealtime_download.html

---

reftags *Example reference tags*

Description

A vector of example reference tag codes

Usage

```
reftags
```

Format

A vector of example reference tag codes

rolling_sd_3 *Calculate Standard Deviation Over a Rolling Window of 3*

Description

This function is called within the third filter, and used to calculate a rolling standard deviation with a window including the 2 leading values.

Usage

```
rolling_sd_3(x)
```

Arguments

x	an indexed position of a value in a vector for which the rolling standard deviation is needed

Value

the standard deviation of x, and the two leading values

Examples

```
# Calculate rolling Standard Deviation
set.seed(1234)
x <- rnorm(n = 100, mean = 10, sd = 1)
rolling_sd_3(x)
```

second_filter_2h4h *Two or Four Hit Filter for Any Detection File*

Description

This function takes any acoustic receiver detection dataframe generated from the add_fish() function and filters it a second time to remove any remaining multipath detections. Then depending on the type of receiver file applies an algorithm to assess and filter all remaining detections based on 2 or 4 hits.

Usage

```
second_filter_2h4h(fish_file)
```

Arguments

fish_file	a dataframe of detections retrieved from add_fish()

Value

A dataframe which has been filtered to remove false positives

Examples

```
# Apply a 2 or 4 hit filter based on the technology
second_filter_2h4h(filter_fish_detects)
```

---

second_filter_4h *Four Hit Filter for Any Detection File*

Description

This function takes any acoustic receiver detection dataframe generated from the add_fish() function and filters it a second time to remove any remaining multipath detections. Then applies an algorithm to assess and filter all remaining detections based on a four hit filter.

Usage

```
second_filter_4h(fish_file)
```

Arguments

fish_file	a dataframe of detections retrieved from add_fish()

Value

A dataframe which has been filtered to remove false positives

Examples

```
# Apply a 4 hit filter based on the technology
second_filter_4h(filter_fish_detects)
```

---

second_filter_ats *Two Hit Filter for ATS Detections*

Description

This function takes an ATS detection dataframe generated from the add_fish() function and filters it a second time to remove any remaining multipath detections, and then check the remaining detections by comparing the time between each detection to ensure it is less 4x the stated pulse rate interval. It additionally checks that all detections have a frequency between 416.3 and 418.75 kHz and that the frequency of all detections are within 0.505kHz of each other. Called by second_filter_2h4h().

Usage

```
second_filter_ats(fish_file)
```

Arguments

fish_file	a dataframe of detections retrieved from add_fish()

Value

A dataframe which has been filtered to remove false positives

Examples

```
# Apply the ATS filter to a prefiltered dataset with fish and tag
# attributes
second_filter_ats(filter_fish_detects)
```

---

second_filter_ats_4h *Four Hit Filter for ATS Detections*

Description

This function takes an ATS detection dataframe generated from the add_fish() function and filters it a second time to remove any remaining multipath detections, and then check the remaining detections by comparing the time between detections, for a rolling window of 4 detections to ensure it is less 16.6x the stated pulse rate interval. It additionally checks that all detections within the window occur within 20% of the pulse rate interval of the other detections and that the standard deviation of pulse rate intervals is less than 0.025. It additionally checks that all detections have a frequency between 416.3 and 418.75 kHz and that the frequency of all detections are within 0.505kHz of each other. Called by second_filter().

Usage

```
second_filter_ats_4h(fish_file)
```

Arguments

fish_file	a dataframe of detections retrieved from add_fish()

Value

A dataframe which has been filtered to remove false positives

Examples

```
# Apply a four-hit ATS filter to a prefiltered dataset with fish and tag
# attributes
second_filter_ats_4h(filter_fish_detects)
# No detections are valid
```

second_filter_lotek *Four Hit Filter for Lotek Detections*

Description

This function takes a Lotek detection dataframe generated from the add_fish() function and filters it a second time to remove any remaining multipath detections, and then check the remaining detections by comparing the time between detections, for a rolling window of 4 detections to ensure it is less 16.6x the stated pulse rate interval. It additionally checks that all detections within the window occur within 20% of the pulse rate interval of the other detections and that the standard deviation of pulse rate intervals is less than 0.025. Called by second_filter().

Usage

```
second_filter_lotek(fish_file)
```

Arguments

fish_file	a dataframe of detections retrieved from add_fish()

Value

A dataframe which has been filtered to remove false positives

Examples

```
# Apply the Lotek filter to a prefiltered dataset with fish and tag
# attributes
second_filter_lotek(filter_fish_detects)
```

second_filter_tekno *Two Hit Filter for Teknologic Detections*

Description

This function takes a Teknologic detection dataframe generated from the add_fish() function and filters it a second time to remove any remaining multipath detections, and then check the remaining detections by comparing the time between each detection to ensure it is less 4x the stated pulse rate interval. It additionally checks that all detections have a frequency between 390 and 445 kHz and that the frequency of all detections are within 55kHz of each other. Called by second_filter_2h4h().

Usage

```
second_filter_tekno(fish_file)
```

Arguments

fish_file	a dataframe of detections retrieved from add_fish()

Value

A dataframe which has been filtered to remove false positives

Examples

```
# Apply the Teknologics filter to a prefiltered dataset with fish and tag
# attributes
second_filter_tekno(filter_fish_detects)
```

---

second_filter_tekno_4h

Four Hit Filter for Tekno Detections

Description

This function takes an Tekno detection dataframe generated from the add_fish() function and filters it a second time to remove any remaining multipath detections, and then check the remaining detections by comparing the time between detections, for a rolling window of 4 detections to ensure it is less 16.6x the stated pulse rate interval. It additionally checks that all detections have a frequency between 390 and 445 kHz and that the frequency of all detections are within 55kHz of each other. Called by second_filter_4h().

Usage

```
second_filter_tekno_4h(fish_file)
```

Arguments

fish_file	a dataframe of detections retrieved from add_fish()

Value

A dataframe which has been filtered to remove false positives

Examples

```
# Apply a four-hit Teknologics filter to a prefiltered dataset with fish and tag
# attributes
second_filter_tekno_4h(filter_fish_detects)
# No detections are valid
```

Set_GVs *Add in Global Variables*

Description

Sets all global variables to remove warnings in package build

Index

∗ **datasets**
    cft_fish, 3
    cft_rcvrs, 5
    filter_fish_detects, 7
    filtered_detections, 7
    fish, 9
    fish_detects, 10
    prefiltered_detects, 19
    raw_ats, 20
    rcvr_fields, 20
    receivers, 23
    reftags, 24

add_fish, 3

cft_fish, 3
cft_rcvrs, 5
compare_detects, 6

filter_fish_detects, 7
filtered_detections, 7
fish, 9
fish_detects, 10
format_detects, 10
format_fish, 12
format_receivers, 13

get_fish_fields, 14
get_rcvr_data, 15
get_rcvr_fields, 15
get_receiver_type, 16
get_reference_tags, 16
get_tagged_fish, 17

join_rcvr_data, 18

prefilter, 18
prefiltered_detects, 19

raw_ats, 20
rcvr_fields, 20
read_ats, 21
read_jsats, 21
read_lotek, 22
read_tekno, 23
receivers, 23
reftags, 24
rolling_sd_3, 25

second_filter_2h4h, 25
second_filter_4h, 26
second_filter_ats, 26
second_filter_ats_4h, 27
second_filter_lotek, 28
second_filter_tekno, 28
second_filter_tekno_4h, 29
Set_GVs, 30