Package: ArgosQC (via r-universe)

November 4, 2026

Type Package

Title Quality Control Process for the Integrated Marine Observing System's Argos Location Data

Version 0.10-2

Date 2026-11-03

Author Ian Jonsen [aut, cre, cph]

Maintainer Ian Jonsen <jonsen@stochastic-qc.org>

Description An automated Argos location quality control process for Argos location data from satellite tags. Functions automatically download and collate data from one of several potential remote source: a user-supplied URL, a user-supplied Google Drive link, a user-supplied Dropbox link, the SMRU server, or the Wildlife Computers Portal API. The package matches deployment data with user-supplied deployment metadata; projects location data from lon,lat to a user-supplied projection or a default projection; fits user-specified SSM's in 2 passes to estimate most plausible locations; collates results by species & deployment program; generates diagnostic plots & maps; appends predicted locations at tag-measured event times to the tag manufacturer activity files such as CTD profiles, dive records, haulout records, and the Argos and (when present) GPS location files; saves activity files as .csv in one of several possible schema (IMOS ATF, ATN, User-defined); pushes QC'd files to a user-specified server or saves to a local archive (zipfile).

URL https://github.com/ianjonsen/ArgosQC/, https://ianjonsen.github.io/ArgosQC/

BugReports https://github.com/ianjonsen/ArgosQC/issues/

License CC BY 4.0

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports cli, scales, furrr (>= 0.3.1), future (>= 1.69.0), dplyr (>= 1.1.4), tidyr (>= 1.3.2), lubridate (>= 1.9.4), assertthat, purrr (>= 1.2.1), readr (>= 2.1.6), stringr (>= 1.6.0), sf (>= 1.0-24), kableExtra, ggplot2 (>= 4.0.1), rnaturalearth (>= 1.2.0), snakecase, aniMotum (>= 1.3-0), XML (>= 3.99-0.20), httr2 (>= 1.2.2), openssl (>= 2.3.4), rvest (>= 1.0.5), tibble (>= 3.3.1), traipse (>= 0.3.0), usethis (>= 3.2.1), vctrs (>= 0.7.1), jsonlite (>= 2.0.0), testthat, rnaturalearthdata, grid

Depends R (>= 4.5.0)

SystemRequirements GDAL (>= 3.8.5), GEOS (>= 3.13.0), PROJ (>= 9.5.1), pandoc (>=3.8.3)

Roxygen list(markdown = TRUE)

Remotes jmlondon/pathroutr, ropensci/rnaturalearthhires, ianjonsen/aniMotum

Suggests knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel false

Config/pak/sysreqs libabsl-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev libgdal-dev gdal-bin libgeos-dev git make libharfbuzz-dev libgit2-dev libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev

Repository https://ianjonsen.r-universe.dev

Date/Publication 2026-11-04 00:30:26 UTC

RemoteUrl https://github.com/ianjonsen/ArgosQC

RemoteRef HEAD

RemoteSha dae9dd3ed5fc93f5d363553464d6aca580332ee9

Contents

accept_trim_dates . . . 3
diagnostics . . . 4
download_data . . . 5
get_metadata . . . 7
imos_smru_dm_qc . . . 8
map_QC . . . 11
multi_filter . . . 13
pull_local_data . . . 13
redo_multi_filter . . . 14
smru_append_ssm . . . 15
smru_clean_diag . . . 15
smru_prep_loc . . . 16
smru_pull_tables . . . 16
smru_qc . . . 17

smru_suggest_trim 20
smru_truncate_ssm 23
smru_write_csv 23
ssm_mark_gaps 24
trim_locs 25
trim_zoom 25
wc_append_ssm 27
wc_prep_loc 27
wc_pull_data 28
wc_qc 29
wc_write_csv 31

Index **33**

`accept_trim_dates` *Accept all suggested track end dates for delayed-mode QC*

Description

Accepts the draft truncation file written by `smru_suggest_trim()` as it is: copies `<cid>_trimIDs_draft.csv` to `<cid>_trimIDs.csv` and sets `"trimIDs"` in the config file to that file. With `set_download_false = TRUE` it also sets `"download"` to `false`, so the delayed-mode QC uses the local .mdb files that the suggestions were made from. Use it after reviewing the review PDF, when every suggestion is accepted unchanged. To change any suggestion, edit the draft and save it as `<cid>_trimIDs.csv` instead.

The config file is edited as text: only the `"trimIDs"` value (or a new `"trimIDs"` line after `"dropIDs"`) and, optionally, the `"download"` value change, so its layout is kept. The edited file is checked to parse, with the new values, before anything is written.

The function runs only for delayed-mode QC (`QCmode = "dm"`).

Usage

```
accept_trim_dates(
  wd,
  config,
  draft_file = NULL,
  trim_file = NULL,
  set_download_false = FALSE,
  overwrite = FALSE
)
```

Arguments

`wd`	the working directory, as in `smru_qc()`
`config`	the JSON config file, as in `smru_qc()`
`draft_file`	the draft truncation file. Default: `<cid>_trimIDs_draft.csv` in `wd`
`trim_file`	the accepted truncation file to write. Default: `<cid>_trimIDs.csv` in `wd`

`set_download_false`
: logical; set `"download"` to `false` in the config file (default `FALSE`)

`overwrite`
: logical; replace an existing `trim_file` (default `FALSE`, so a hand-edited file is not lost)

Value

invisibly, the path of the accepted truncation file

`diagnostics` *run QC diagnostics for WC or SMRU tag workflows*

Description

produces a map of all QC'd tracks and latitude and longitude coverage plots to assess the QC run. The map shows each SSM-estimated track in its own hue, with its Argos locations in a light tint of that hue; Argos locations removed by the prefilter are semi-transparent points in the track's dark shade. The coverage plots use the same hues, one panel per deployment, with the SSM-estimated latitude or longitude drawn as a dark line over the Argos locations, and the time outside the QC'd track shaded.

Usage

```
diagnostics(
  fit,
  fit1,
  what = "p",
  cut,
  data,
  ssm,
  meta,
  lines = FALSE,
  obs = FALSE,
  mpath = NULL,
  dpath = NULL,
  QCmode = "nrt",
  tag_mfr = "wc",
  cid = NULL
)
```

Arguments

`fit`
: the final aniMotum fit object from QC process

`fit1`
: ignored (kept for compatibility)

`what`
: the SSM-estimated or rerouted locations to be used

`cut`
: logical; should predicted locations be dropped if keep = FALSE - ie. in a large data gap

data	the standardized WC Locations or SMRU diag file (prior to truncation by metadata CTD start and end dates)
ssm	ignored (kept for compatibility)
meta	ignored (kept for compatibility)
lines	ignored; the map always shows track lines (kept for compatibility)
obs	ignored; the map always shows the Argos locations (kept for compatibility)
mpath	path to write map file
dpath	path to write all other diagnostic files
QCmode	ignored (kept for compatibility)
tag_mfr	ignored (kept for compatibility)
cid	SMRU campaign id (from config file). Ignored if WC data is used.

`download_data` *Download SMRU or Wildlife Computers tag data from their servers*

Description

Satellite tracking data are accessed from the SMRU data server, or accessed from the Wildlife Computers Portal API via the `source` argument. Data files are saved to the `data.dir` specified in the JSON config file. SMRU tag data are currently downloaded as a single `.mdb` (Microsoft Access Database) file. Wildlife Computers tag data are downloaded as a series of `.CSV` files saved in tag-specific directories (uniquely named with WC UUID's). Wildlife Computers data, partial deployment metadata are output as an R object.

Usage

```
download_data(
  dest = NULL,
  source = "smru",
  cid = NULL,
  user = NULL,
  pwd = NULL,
  wc.akey = NULL,
  wc.skey = NULL,
  owner.id = NULL,
  subset.ids = NULL,
  download = TRUE,
  ...
)
```

Arguments

dest	destination path to save download
source	source type of data to be downloaded. Can be one of: • smru - SMRU Data Server; • wc - Wildlife Computers Data Portal API;
cid	SMRU tag deployment campaign id(s) to download, eg. "ct180"
user	SMRU data server username as a quoted string
pwd	SMRU data server password as a quoted string
wc.akey	an Access Key issued by Wildlife Computers for their API
wc.skey	a Secret Key issued by Wildlife Computers for their API
owner.id	the Wildlife Computers uuid associated with the data owner
subset.ids	a single column .CSV file of WC UUID's to be included in the QC, with uuid as the variable name.
download	(logical) indicating if the data is to be downloaded from the tag manufacturer's server. If the source is wc and download = FALSE then only the Wildlife Computers tag deployment metadata is downloaded.
...	additional arguments passed to smru_get_mdb() or wc_get_files()

Examples

```
## Not run:
## SMRU data download
  download_data(
    dest = file.path(wd, config$setup$data.dir),
    source = "smru",
    cid = config$harvest$cid,
    user = config$harvest$smru.usr,
    pwd = config$harvest$smru.pwd,
    timeout = config$harvest$timeout
    )

## Wildlife Computers data download & deployment metadata acquisition
wc.deploy.meta <- download_data(
                    dest = file.path(wd, config$setup$data.dir),
                    source = "wc",
                    unzip = TRUE,
                    wc.akey = config$harvest$wc.akey,
                    wc.skey = config$harvest$wc.skey,
                    subset.ids = config$harvest$tag.list,
                    download = TRUE,
                    owner.id = config$harvest$owner.id
                    )

## End(Not run)
```

`get_metadata` *download deployment metadata*

Description

downloads, restructures & formats metadata, appends dive/CTD start and end datetimes (for QC), & fills in missing required metadata - eg. release_datetime, release_longitude/latitude's with data from the GPS (if present) or Argos location file.

Usage

```
get_metadata(
  source = "smru",
  tag_mfr = "smru",
  tag_data = NULL,
  cid = NULL,
  user = NULL,
  pwd = NULL,
  dropIDs = NULL,
  file = NULL,
  meta.args,
  subset.ids = NULL,
  wc.meta = NULL,
  QCmode = NULL,
  build.meta = FALSE,
  server.check = TRUE,
  path2mdb = NULL,
  p2mdbtools = NULL
)
```

Arguments

`source`	the source of the deployment metadata, current options are `imos`, `otn`, `smru`, or `atn`. If `source = 'imos'` or `source = 'atn'` then metadata are obtained from user-provided .CSV file via the config.json file. If `sourcce = 'smru'` then metadata are built from a combination of SMRU server details & deployment details in the config.json file.
`tag_mfr`	the tag manufacturer, current options are `smru` or `wc` cid SMRU campaign ids
`tag_data`	a list of either `smru` data tables or `wc` files as output by `pull_data`.
`cid`	SMRU campaign id must be provided when the tag_mfr is `smru`
`user`	SMRU data server username as a quoted string - to be used only if metadata are to be built from SMRU server details (`source = 'smru'`).
`pwd`	SMRU data server password as a quoted string - to be used only if metadata are to be built from SMRU server details (`source = 'smru'`).
`dropIDs`	SMRU refs or WC ids to be dropped from QC

file	path to metadata .csv file, if provided then metadata will be read from the provided source
meta.args	optional metadata fields to be passed from config file when downloading tag metadata from SMRU server. Typically used only when no metadata filepath is provided in the config file.
subset.ids	a character vector of comma-separated (no spaces) WC UUID's to be included in the QC. Ignored if tag_mfr != "wc".
wc.meta	an R data.frame of Wildlife Computers tag deployment metadata obtained via download_data(). Typically used only when no metadata filepath is provided in the config file.
QCmode	"nrt" or "dm". For IMOS metadata in delayed-mode QC, the tag identifiers are checked against the SMRU data server (see smru_clean_meta_imos())
build.meta	logical; delayed-mode QC only. Build the metadata of deployments missing from the metadata file from the SMRU data server, the .mdb deployments table, the Argos locations and meta.args
server.check	logical; delayed-mode QC of IMOS metadata only. Check ptt, body, WMO ID and tag type against the SMRU data server (default TRUE). FALSE takes them from the metadata file unchecked; build.meta must then be FALSE. Set by "server.check" in the config setup block
path2mdb, p2mdbtools	the .mdb folder and the mdbtools path (used when build.meta = TRUE)

---

imos_smru_dm_qc *Delayed-mode QC of one IMOS SMRU tag campaign, from config file to AODN upload*

Description

Runs the whole delayed-mode (DM) QC of one IMOS SMRU tag campaign (seals or turtles) and records it:

1. writes, or checks and updates, the campaign config file;
2. suggests track end dates with smru_suggest_trim() and opens the review PDF;
3. asks the operator to accept, edit or reject the suggested end dates;
4. runs smru_qc();
5. zips the output .csv files;
6. uploads the zip file to AODN by sftp, after the operator confirms.

Usage

```
imos_smru_dm_qc(
  cid,
  wd,
  config = NULL,
  meta.file = NULL,
  species = NULL,
  common_name = NULL,
  release_site = NULL,
  state_country = NULL,
  dropIDs = NULL,
  build.meta = NULL,
  model.args = list(),
  smru.usr = NULL,
  smru.pwd = NULL,
  p2mdbtools = "/opt/homebrew/Cellar/mdbtools/1.0.1/bin/",
  download = NULL,
  review = TRUE,
  upload = TRUE,
  aodn.user = "",
  aodn.path = "/data/IMOS/AATAMS/ANIMAL_TRACKING_SATTAG_QC/dm/",
  ...
)
```

Arguments

cid	the SMRU campaign id, e.g. "ct155"
wd	the working directory for the campaign's DM QC
config	the config file name (relative to wd) or path. Default: config_<cid>.json
meta.file	the IMOS metadata .csv file. If NULL, the file named in an existing config file is used. Delayed-mode QC requires a metadata file: the function stops if there is none. It also stops if the file has no deployments for cid, unless build.meta = TRUE. Each deployment's ptt, body, WMO ID and tag type are checked against the SMRU data server, whose values are used where they differ; the QC stops if the server cannot be reached. Set "server.check": false in the config setup block to skip the SMRU server check for one campaign: ptt, body, WMO ID and tag type are then taken from the metadata file unchecked, and build.meta must be false
species	the species scientific name. Required only if it cannot be read from meta.file or an existing config file
common_name	the species common name. Default: from the species lookup table
release_site	the release site. Default: the metadata Location, or the existing config value
state_country	the country or territory of the release site. Default: from the release site lookup table
dropIDs	a .csv file of deployments to exclude from the QC. Default: the existing config value, or an existing <cid>_dropIDs.csv in wd. Deployments are QC'd whether or not they have a WMO ID

`build.meta` logical; QC deployments that are in the tag data but not in the metadata file, with metadata built from the SMRU data server (ptt, body, WMO ID, tag type), the Argos locations (release date and position, after any pre-deployment test transmissions), the .mdb deployments table `LOCATION` (release site, used if within 50 km of the estimated release position; otherwise a known site within 10 km of the daily median position on any of the first 7 days; otherwise the nearest known site within 50 km, or the config `meta` release site within 50 km) and the config `meta` block (species). The built metadata are listed for confirmation and recorded in `qc/diag/<cid>_built_metadata.csv`. `NULL` (default) uses `"build.meta"` in the config setup block (`false` if absent)

`model.args` a named list of `model` settings that replace the species defaults or the existing config values, e.g. `list(dist = 50)`

`smru.usr`, `smru.pwd` the SMRU data server login. Required for a new config file

`p2mdbtools` the path to the mdbtools binaries, used for a new config file

`download` logical; download the campaign .mdb file from the SMRU server before the QC, replacing the local copy. `NULL` (default) uses `"download"` in the config file (`false` for a new config file). The .mdb file is always downloaded if there is no local copy. It is downloaded once per run, so the trim review and the QC use the same data

`review` logical; review the suggested track end dates interactively (`TRUE`), or use the `trimIDs` file set in the config file (`FALSE`)

`upload` logical; offer to upload the zip file to AODN. The upload always needs confirmation in an interactive session

`aodn.user` the AODN sftp username. If blank, it is asked for at upload time

`aodn.path` the AODN sftp directory for DM QC files

`...` further arguments passed to `smru_suggest_trim()`

Details

Config file. If `config` exists, it is read and these settings are applied: `"QCmode": "dm"`; `"diag.dir": "qc/diag"`; for Weddell seals, `"time.step": 6` and `"min.gap": 48`; the `meta` section derived as below; and any `meta.file`, `dropIDs`, `smru.usr`, `smru.pwd`, `download` and `model.args` supplied. Every change is listed and must be confirmed before the file is rewritten. If `config` does not exist, it is built from the species defaults and the arguments.

The `meta` section. `species` and `release_site` are taken, in order of precedence, from the arguments, from the IMOS metadata file (the `Species` and `Location` of the campaign's deployments) and from the existing config file. `common_name` comes from the argument or else from the species lookup table. `state_country` comes from the argument or else from the release site lookup table. With a metadata file, `release_site` lists every site of the campaign, most common first, joined by `"; "`, and `state_country` lists their states or countries the same way. These entries are a record only: the output metadata take the release site, state or country, and species of each deployment from the metadata file. `species` (the most common) still selects the model settings.

Track truncation review (`review = TRUE`). The choices are:

1. accept all suggested end dates (`accept_trim_dates()`);

2. edit <cid>_trimIDs_draft.csv, save it, then continue (accept_trim_dates() checks it and copies it to <cid>_trimIDs.csv);
3. use an existing <cid>_trimIDs.csv unchanged;
4. truncate no tracks;
5. stop.

With review = FALSE, the trimIDs file already set in the config file is used, or no track is truncated if none is set. Use it to repeat a run with the recorded config file and track truncation file.

Records. Each run writes:

1. the config file as used;
2. <cid>_trimIDs.csv (if any track is truncated);
3. qc/diag/<cid>_dm_log.txt, appended on each run, with the ArgosQC and aniMotum versions (and git commits where available), the review decision, the files zipped and the sftp outcome.

Value

invisibly, a list with the config file path, the trimIDs file path (or NULL), the zip file path, whether the upload succeeded, and the value returned by smru_qc()

Examples

```
## Not run:
## first run: review the suggested track end dates interactively
imos_smru_dm_qc(cid = "ct155",
                wd = "/Volumes/work/R/imos/imos_sat/seals/past_qc/dm_qc",
                aodn.user = "user@example.org")

## repeat a run with the recorded config and track truncation files
imos_smru_dm_qc(cid = "ct155",
                wd = "/Volumes/work/R/imos/imos_sat/seals/past_qc/dm_qc",
                review = FALSE)

## End(Not run)
```

---

map_QC *map_QC*

Description

map aniMotum-estimated locations and behavioural indices with coastline and projection options

Usage

```
map_QC(
  x,
  y = NULL,
  what = c("fitted", "predicted", "rerouted"),
  aes = aes_lst(),
  by.id = TRUE,
  by.date = FALSE,
  cut = FALSE,
  crs = NULL,
  ext.rng = c(0.05, 0.05),
  buffer = 10000,
  normalise = TRUE,
  group = FALSE,
  silent = FALSE
)
```

Arguments

x	a aniMotum ssm fit object with class ssm_df or (old) fG_ssm
y	optionally, a aniMotum mpm fit object with class mpm_df or (old) fG_mpm
what	specify which location estimates to map: fitted, predicted or rerouted
aes	a list of map controls and aesthetics (shape, size, col, fill, alpha) for each map feature (estimated locations, confidence ellipses, track lines, observed locations, land masses, water bodies). Constructed by aes_lst() and can be modified for custom maps (see examples)
by.id	when mapping multiple tracks, should locations be coloured by id (logical; default = TRUE if nrow(x) > 1 else FALSE; ignored if behavioural index provided)
by.date	when mapping single tracks, should locations be coloured by date (logical; default = FALSE; ignored if behavioural index provided)
cut	logical; should predicted locations be dropped from mapping if keep = FALSE. default = FALSE.
crs	proj4string for re-projecting locations, if NULL the default projection (Mercator) for the fitting the SSM will be used
ext.rng	proportion (can exceed 1) to extend the plot range in x and y dimensions
buffer	distance (in km) to buffer locations for subsetting land polygons (default = 10000). If map extents are expanded by many factors then the buffer distance may need to be increased, otherwise this should not be used. Ignored if map_type != "default".
normalise	logical; if output includes a move persistence estimate, should g (the move persistence index) be normalised to have minimum = 0 and maximum = 1 (default = TRUE).
group	logical; should g be normalised among individuals as a group, a 'relative g', or separately to highlight regions of lowest and highest move persistence along a track (default = FALSE).

silent	logical; generate maps silently (default = FALSE).

Value

a map as a ggplot2 object

`multi_filter` *multi-filter*

Description

apply SSM filter to diag data across multiple processors

Usage

```
multi_filter(x, vmax = 4, model = "rw", ts = 6, verbose = FALSE)
```

Arguments

x	sf-projected diag file of locations to be filtered
vmax	for prefilter
model	aniMotum model ("rw" orr "crw)
ts	foieGrsa time.step
verbose	turn on/off furrr::future_map progress indicator

`pull_local_data` *Pull tables from local tag datafiles*

Description

reads SMRU or WC tag datafiles & combines in a unified list

Usage

```
pull_local_data(path2data, cid = NULL, tag_mfr)
```

Arguments

path2data	path to local datafile(s)
cid	SMRU campaign id. Ignored if tag_mfr = "wc".
tag_mfr	either "smru" or "wc"

`redo_multi_filter` *redo failed multi-filter cases*

Description

re-apply SSM filter to diag data for id's that failed to converge. parallelized

Usage

```
redo_multi_filter(
  fit,
  diag_sf,
  model = "crw",
  ts = 3,
  vmax = 2,
  ang = c(15, 25),
  distlim = c(1500, 5000),
  min.dt = 180,
  map = NULL,
  reroute = TRUE,
  dist = 500,
  barrier = NULL,
  onland = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

`fit`	aniMotum fit object from first round of filtering
`diag_sf`	`sf`-projected diag to be used
`model`	model argument ("rw" or "crw) for `aniMotum::fit_ssm`
`ts`	time.step argument for `aniMotum::fit_ssm`
`vmax`	threshold travel speed (m/s) to apply during track pre-filtering
`ang`	sdafilter argument
`distlim`	sdafilter argument
`min.dt`	min.dt argument for `aniMotum::fit_ssm`
`map`	params to fix
`reroute`	(logical) should SSM-predicted locations be re-routed off of land (default is TRUE)
`dist`	the distance (in km) to buffer around predicted locations. This buffer allows a larger portion of coastline to be selected for rerouting any locations that are on land. More coastline polygon data can help rerouting, but too much will make computation very slow.

barrier	add a custom POLYGON/MULTIPOLYGON shapefile to use as a land barrier. Default (NULL) reverts to the aniMotum default land barrier.
onland	on-land intervals (data frame with id, start, end), from haul-out and dive records; SSM-estimated locations within them are not re-routed off land (see aniMotum::route_path())
verbose	turn on/off furrr::future_map progress indicator
...	additional arguments to aniMotum::route_path

---

smru_append_ssm *append SMRU tables with SSM-estimated locations*

Description

append SMRU tables so each event has SSM-derived lon, lat, x, y, x.se, y.se.

Usage

```
smru_append_ssm(smru, fit, what = "p", meta, cut = FALSE, dropIDs = NULL)
```

Arguments

smru	SMRU table file - output of pull_smru_tables
fit	final foieGras fit object
what	choose which locations to use for annotating SMRU tables (default = "predicted")
meta	metadata used to truncate start of diag data for each individual
cut	drop predicted locations if keep = FALSE, ie. locations in a large data gap
dropIDs	SMRU refs to be dropped

---

smru_clean_diag *clean diag files*

Description

restructures diag files, formats dates & lc's in preparation for SSM-filtering

Usage

```
smru_clean_diag(smru, dropIDs = NULL)
```

Arguments

smru	list of SMRU tables
dropIDs	SMRU refs to be dropped (eg. tags were turned on but not deployed)

smru_prep_loc *clean, truncate & convert diag files to sf-tibbles*

Description

restructures diag files, formats dates & lc's; keeps the deployments in the metadata and starts each track at its deployment start (metadata column `deploy_start`, set by `smru_deployment_start()`); converts to sf geometry - all in preparation for SSM-filtering. Tracks are not truncated at the first or last CTD profile or dive record.

Usage

```
smru_prep_loc(smru, meta, dropIDs = NULL, crs = NULL, QCmode = NULL)
```

Arguments

`smru`	list of SMRU tables
`meta`	metadata; its `deploy_start` column (if present) sets the start of each track
`dropIDs`	SMRU refs to be dropped (eg. tags were turned on but not deployed)
`crs`	a proj4string to re-project diag locations from longlat. Default is NULL which results in one of 4 possible projections applied automatically, based on the centroid of the tracks. See `overview` vignette for details.
`QCmode`	specify whether QC is near real-time (nrt) or delayed-mode (dm)

smru_pull_tables *Pull tables from SMRU .mdb files*

Description

extracts specified tables from SMRU .mdb files, using Hmisc::mdb.get

Usage

```
smru_pull_tables(
  cids,
  path2mdb,
  tables = c("diag", "gps", "haulout", "ctd", "dive", "cruise", "summary"),
  p2mdbtools = NULL,
  verbose = FALSE
)
```

Arguments

cids	SMRU campaign ids
path2mdb	path to SMRU .mdb file(s)
tables	specify which tables to extract, default is to extract all tables
p2mdbtools	path to mdbtools binaries. Specifying the path can avoid an error when calling from within RStudio, eg. on MacBook Pro M1 Pro with homebrew-installed mdbtools @ /opt/homebrew/Cellar/mdbtools/1.0.0/bin/
verbose	turn on/off progress indicator

---

smru_qc *SMRU SRDL QC workflow*

Description

Wrapper function that executes the complete SMRU QC workflow from data download to SSM-appended tag data files output as CSV files. All settings are specified in a JSON config file, including program - currently, IMOS, ATN or OTN. The program field determines the specific ArgosQC workflow functions called within the wrapper fn.

Usage

```
smru_qc(wd, config, download = NULL)
```

Arguments

wd	the path to the working directory that contains: 1) the data directory where tag data files are stored (if harvest$download = FALSE) or downloaded to (if harvest$download = TRUE); 2) the metadata directory where all metadata files are stored; and 3) the destination directory for QC outputs.
config	a hierarchical JSON configuration file containing the following blocks, each with a set of block-specific parameters: • setup config block specifies paths to required data, metadata & output directories: – program the national (or other) program of which the data is a part. Current options are: imos, atn, or otn. – data.dir the name of the data directory. Must reside within the wd. – meta.file the metadata filename. Must reside within the wd. Required for delayed-mode QC (QCmode = "dm"). For near real-time QC only, it can be NULL, in which case the meta config block (see below) must be present & tag-specific metadata are scraped from the SMRU data server. In delayed-mode QC with an IMOS metadata file, each deployment's ptt, body, WMO ID and tag type are checked against the SMRU data server, whose values are used where they differ; the QC stops if the server cannot be reached.

  - `build.meta` optional; delayed-mode QC only (default `false`). If `true`, deployments in the tag data but not in the metadata file are QC'd, with metadata built from the SMRU data server, the .mdb deployments table, the Argos locations and the `meta` config block. If `false`, they are excluded.
  - `maps.dir` the directory path to write diagnostic maps of QC'd tracks.
  - `diag.dir` the directory path to write diagnostic time-series plots of QC'd lon & lat.
  - `output.dir` the directory path to write QC output CSV files. Must reside within the `wd`.
  - `return.R` logical; should the function return a list of QC-generated objects to the R works pace. This results in a single large object containing the following elements:
    * `cid` the SMRU campaign ID
    * `dropIDs` the SMRU Reference ID's droppped from the QC process
    * `smru` the SMRU tag data tables extracted from the downloaded .mdb file
    * `meta` the working metadata
    * `locs_sf` the projected location data to be passed as input to the SSM
    * `fit1` the initial SSM output fit object
    * `fit2` the final SSM output fit object including re-routed locations if specified.
    * `smru_ssm` the SSM-annotated SMRU tag data tables. This output object can be useful for troubleshooting undesirable results during supervised or delayed-mode QC workflows.
- `harvest` config block specifies data harvesting parameters:
  - `download` a logical indicating whether tag data are to be downloaded from the SMRU data server or read from the local `data.dir`.
  - `cid` SMRU campaign ID.
  - `smru.usr` SMRU data server username as a string.
  - `smru.pwd` SMRU data server password as a string.
  - `timeout` extends the download timeout period a specified number of seconds for slower internet connections.
  - `dropIDs` the SMRU ref ID's that are to be ignored during the QC process. SMRU ref ID's must be supplied as a .CSV file `dropIDs.csv` with a single variable named `ref`. Can be NULL.
  - `trimIDs` (optional; delayed-mode QC only) a .CSV file of per-deployment track truncation dates, with a `ref` column and an `end_date` and/or `start_date` column (UTC). See `trim_locs()`; draft files can be made with `smru_suggest_trim()`. Can be NULL or absent. The QC stops if `trimIDs` is set and `QCmode` is not `dm`.
  - `p2mdbtools` (optional) provides the path to the mdbtools library if it is installed in a non-standard location (e.g., on Macs when installed via Homebrew).
- `model` config block specifies model- and data-specific parameters:

  - model the aniMotum SSM model to be used for the location QC - typically either rw or crw.
  - vmax for SSM fitting; max travel rate (m/s) to identify implausible locations
  - time.step the prediction interval (in decimal hours) to be used by the SSM
  - proj the proj4string to be used for the location data & for the SSM-estimated locations. Can be NULL, which will result in one of 5 projections being used, depending on whether the centroid of the observed latitudes lies in N or S polar regions, temperate or equatorial regions, or if tracks straddle (or lie close to) -180,180 longitude.
  - reroute a logical; whether QC'd tracks should be re-routed off of land (default is FALSE). Note, in some circumstances this can substantially increase processing time. Default land polygon data are sourced from the ropensci/rnaturalearthhires R package.
  - dist the distance in km from outside the convex hull of observed locations from which to select land polygon data for re-routing. Ignored if reroute = FALSE.
  - barrier the file path (must be within the working directory) for a shapefile to use for the land barrier. If NULL (default) then the default rnaturalearth coastline polygon data is used.
  - buffer the distance in km to buffer rerouted locations from the coastline. Ignored if reroute = FALSE.
  - centroids whether centroids are to be included in the visibility graph mesh used by the rerouting algorithm. See ?pathroutr::prt_visgraph for details. Ignored if reroute = FALSE.
  - ho.gap optional; hours (default 6). Haul-out records less than ho.gap h apart, with no dive record between them, form one on-land interval. SSM-estimated locations within on-land intervals, and locations on land at the start or end of a track, are not re-routed off land. Seals, sea lions and fur seals: every haul-out record is used. Turtles: only haul-out records within 21 days of the deployment start and within 5 km of the deployment position (a return to the nesting beach). Other species: none. The species is taken from the config meta block, or from the deployment metadata. Ignored if reroute = FALSE.
  - cut logical; should predicted locations be dropped if they lie within in a large data gap (default is FALSE).
  - min.gap the minimum data gap duration (h) to be used for cutting predicted locations (default is 72 h)
  - QCmode one of either nrt for Near Real-Time QC or dm for Delayed Mode QC.
  - inactive.days optional; near real-time QC only. If no tag has transmitted (Argos location or GPS fix) for more than this many days, the QC stops with an error before any output is written (default is 7 days).
- meta config block specifies species and deployment location information. This config block is only necessary when no metadata file is provided in the setup config block.

- common_name the species common name (e.g., "southern elephant seal")
- species the species scientific name (e.g., "Mirounga leonina")
- release_site the location where tags were deployed (e.g., "Iles Kerguelen")
- state_country the country/territory name (e.g., "French Overseas Territory")

download optional logical; overrides "download" in the config file. NULL (default) uses the config value

---

smru_suggest_trim *Suggest track end dates for delayed-mode QC of SMRU tag data*

Description

Reads a campaign's SMRU tag data and deployment metadata, as smru_qc() does, but fits no SSM. For each deployment it suggests an end_date for truncating the track, writes the suggestions to a draft .csv file for review, and writes a review PDF with one page per deployment: the whole track beside a zoom on the end of the track. After review, the file is supplied to smru_qc() through the trimIDs entry of the config file.

The function runs only for delayed-mode QC (QCmode = "dm"). Near real-time QC is unsupervised, so no tracks are truncated there.

Usage

```
smru_suggest_trim(
  wd,
  config,
  draft_file = NULL,
  plot_file = NULL,
  min_post_days = 2,
  stationary_radius_km = 30,
  min_stationary_days = 2,
  sparse_mean = 10,
  very_sparse_daily = 5,
  final_haulout_days = 2,
  min_final_haulout_hours = 6,
  haulout_reversal_days = 5,
  final_gap_days = 7,
  final_gap_max_days = 3,
  final_outlier_km = 500,
  final_outlier_days = 2,
  keep_days = 0,
  plots = TRUE
)
```

Arguments

wd	the working directory, as in smru_qc()
config	the JSON config file, as in smru_qc(). QCmode must be "dm"
draft_file	path for the draft truncation file. Default: <cid>_trimIDs_draft.csv in wd
plot_file	path for the review PDF. Default: trim_review_<cid>.pdf in the config's diag.dir. Use plots = FALSE to skip the plots
min_post_days	minimum days of locations after the anchor for rules 2 and 3
stationary_radius_km	largest distance (km) of a daily median position from the median position of the final run of days (rule 2)
min_stationary_days	minimum length (days) of the final stationary run (rule 2)
sparse_mean	centred 3-day mean number of locations per day below which a day counts as sparse (rule 3)
very_sparse_daily	a sparse final run must include a day with fewer locations than this (rule 3)
final_haulout_days	a haul-out beginning within this many days of the last location can set a candidate at its start (rule 5)
min_final_haulout_hours	minimum length (hours) of that haul-out (rule 5); shorter dry periods are not treated as a final haul-out
haulout_reversal_days	if the anchor follows the start of the final haul-out by no more than this many days, the haul-out candidate is replaced by the point just after the anchor (rule 5)
final_gap_days	gap between locations, in days, that can end the located track (rule 6)
final_gap_max_days	the longest span of locations after a final gap, in days, for rule 6 to apply
final_outlier_km	distance (km) from the median end-of-track position beyond which a location is a final outlier (rule 7)
final_outlier_days	length (days) of the end-of-track window searched for final outliers (rule 7)
keep_days	days added to each suggested end_date from rules 2 and 3
plots	logical; write the review PDF

Details

The rules, applied to each deployment:

1. The anchor is the last CTD profile, or the last dive for a tag with no CTD profiles. CTD profiles and dives dated after the end of the located track (the last location, or the start of a final gap under rule 6) are ignored: they cannot be located, and may be misdated records. Their number

is reported in n_records_after_track_end. A suggested end_date never removes a CTD profile (except under rule 5): it always keeps at least one location more than one SSM time step (time.step in the config file) after the anchor, so the SSM-predicted track covers every CTD profile. Dives after the last CTD profile are not protected; they are often records from a failing sensor.

2. Stationary end: the final run of days whose daily median positions (all location classes except Z) after the anchor lie within stationary_radius_km of the run's median position, lasting at least min_stationary_days. This is a haul-out, or a likely haul-out with missing haul-out records. Candidate: 00:00 UTC on the run's first day.
3. Sparse end: the final run of days whose centred 3-day mean number of locations is below sparse_mean, provided that run includes a day with fewer than very_sparse_daily locations. This is the location data petering out at the end of the track (battery, antenna or other tag failure). Candidate: 00:00 UTC on the run's first day.
4. Rules 2 and 3 apply only when there are at least min_post_days of locations after the anchor. The suggested end_date is the earlier of their candidates, moved later if needed to satisfy rule 1, then moved later by keep_days. With neither candidate, the track is kept to its end.
5. Final haul-out: a haul-out that begins within the last final_haulout_days of the track and lasts at least min_final_haulout_hours sets a candidate at its start, even if CTD profiles follow it. Haul-out records separated by less than 1 hour, with no dive ending in between, are merged into one haul-out first. If the anchor (the last CTD profile) comes after the start of that haul-out, and no more than haulout_reversal_days after it, the haul-out was not final: the candidate is replaced by the point just after the anchor (rule 1), so the last CTD profile is kept. The suggested end_date is the earliest candidate.
6. Final gap: when the locations after the last gap of more than final_gap_days span no more than final_gap_max_days and include no CTD profile, the located track ends at the start of that gap. The rule is repeated on the remaining track while it applies. CTD profiles and dives inside or after the gap are ignored for the anchor (rule 1). Candidate: just after the last location before the gap.
7. Final outlier: an Argos location of any class, including Z, in the last final_outlier_days of the track that lies more than final_outlier_km from the median position of the other locations (all classes except Z) in that window. A single gross error at the end of a track can pull the SSM track far away, because the speed filter cannot test the last locations well. Candidate: the time of the earliest such location, so it and any later locations are removed. The candidate is used only if it keeps every CTD profile (rule 1); otherwise the outlier is reported in rule for review.

Value

invisibly, a data frame with one row per deployment: ref, start_date (empty), end_date (the suggestion, or empty), rule, anchor, anchor_type, last_dive, last_ctd, last_location, n_records_after_track_end, days_after_anchor, stationary_end, sparse_end, final_haulout_start, final_gap_start, final_outlier, n_locations_removed, n_dives_removed, n_ctd_removed. The same table is written to draft_file. The review data are attached as attribute plot_data, for trim_zoom().

smru_truncate_ssm *right-truncate all SSM-appended SMRU tables using CTD end date for given individuals*

Description

right-truncate all SSM-appended SMRU tables using CTD end date for given individuals, using CTD date-times from metadata

Usage

```
smru_truncate_ssm(smru_ssm, meta, refs)
```

Arguments

smru_ssm	SSM-appended SMRU file to use
meta	metadata used to truncate SSM-appended SMRU tables for each individual
refs	device_id's (SMRU ref's) to apply truncation

smru_write_csv *reconfigure annotated SMRU tables, test for expected data schema for AODN*

Description

reconfigure annotated tables - for near real-time QC (`suffix = "_nrt"`), subsample predicted locations to every ceiling(6 / time.step)-th prediction; for delayed-mode and historical QC, output predicted locations at the SSM time.step. Write to .csv and zip by campaign id

Usage

```
smru_write_csv(
  smru_ssm,
  fit,
  what,
  meta,
  program = "imos",
  proj = NULL,
  test = TRUE,
  path = NULL,
  dropIDs = NULL,
  suffix = "_nrt",
  conform = TRUE
)
```

Arguments

smru_ssm	SSM-appended SMRU table file - output of append_ssm
fit	final aniMotum fit object
what	specify whether predicted or rerouted locations are to be used
meta	metadata
program	Determines structure of output metadata. The imos, otn, & atn programs have their own defined metadata structures, any other program name is treated as "Generic" but may not work properly.
proj	the proj4string specified in the .JSON config file & used to project the location data prior to SSM fitting. It is passed in here to be added to the output metadata .CSV file
test	should variables be tested for standards compliance, default is TRUE. Standards compliance is specific to the program. Currently, only program = imos has defined variable standard against which output compliance is tested.
path	path to write .csv files
dropIDs	individual SMRU ids to be dropped
suffix	suffix to add to .csv files (_nrt, _dm, or _hist)
conform	logical; for program = "imos", conform the diag, ctd, dive, haulout and summary tables to the standard AODN structure before writing: missing variables are added as NA, variables not in the standard structure are dropped, and variables are put in the standard order. Tables from current SMRU tags already have this structure and are unchanged. Default TRUE.

---

ssm_mark_gaps *Identify SSM predicted & rerouted (if present) location estimates in data gaps*

Description

Identify & mark SSM predicted & rerouted (if present) location estimates in track segments with data gaps of a specified minimum duration.

Usage

```
ssm_mark_gaps(ssm, min.gap = 24, mark = TRUE)
```

Arguments

ssm	the SSM fit object from redo_multi_filter()
min.gap	the minimum data gap duration from which SSM estimates are removed (in hours)
mark	logical; should the SSM data be marked (TRUE; default), otherwise the function does no marking and returns the original SSM fit object

trim_locs — *Truncate the location data of individual deployments (delayed-mode QC only)*

Description

Right- and/or left-truncates the prepared location data of individual deployments, using dates supplied in a .csv file. This step runs after `smru_prep_loc()` or `wc_prep_loc()` and before the SSM is fitted. Track truncation is a supervised decision, so the function stops unless the QC mode is delayed-mode (`QCmode = "dm"`). It is never applied in near real-time QC.

Usage

```
trim_locs(locs_sf, file, QCmode)
```

Arguments

`locs_sf`	the prepared location data returned by `smru_prep_loc()` or `wc_prep_loc()`
`file`	path to a .csv file with a `ref` column (the deployment ID: the SMRU ref, or the Wildlife Computers DeploymentID) and an `end_date` and/or `start_date` column. Dates are UTC and may be given as `YYYY-MM-DD` (midnight) or `YYYY-MM-DD HH:MM:SS`. Locations at or after `end_date` and before `start_date` are removed. Rows with no dates, and any other columns (such as those written by `smru_suggest_trim()`), are ignored.
`QCmode`	the QC mode from the config file; must be `"dm"`

Value

`locs_sf` with the location data of the listed deployments truncated

trim_zoom — *Zoom in on part of a deployment's time series to choose track start and end dates*

Description

Draws any time window of one deployment from the `smru_suggest_trim()` review data: a map of the locations in the window, coloured by time, beside their latitude and longitude against time and the counts of locations, dives and CTD profiles. All location classes are shown, including Z. The suggested end date, the anchor and the dates in the draft track truncation file are drawn as vertical lines, so a start or end date can be placed at an exact time, clear of a messy stretch of track. Locations more than `outlier_km` from the running median position (7 locations, centred) are ringed in red: a cluster of rings marks a messy stretch. A table of the last `n` locations in the window is printed, with the distance, time and speed from the previous location and the distance from the running median; outlying locations are marked `*`.

In `imos_smru_dm_qc()` it is offered at the track end date review menu.

Usage

```
trim_zoom(
  trim,
  ref,
  from = NULL,
  to = NULL,
  draft = NULL,
  outlier_km = 100,
  file = NULL,
  n = 30
)
```

Arguments

trim	the data frame returned by smru_suggest_trim()
ref	the deployment reference (- and _ are treated as the same)
from, to	the window, as "YYYY-MM-DD" or "YYYY-MM-DD HH:MM" (UTC). Default: the start and the end of the track
draft	optional path to the draft track truncation file (<cid>_trimIDs_draft.csv); its start_date and end_date for ref are drawn as blue dashed lines
outlier_km	distance (km) from the running median position beyond which a location is marked as outlying
file	optional PDF file to write the plot to. Default: the current graphics device (the RStudio Plots pane, which can be enlarged)
n	the number of locations listed: the last n in the window (0 lists none)

Value

invisibly, a data frame of the locations in the window

Examples

```
## Not run:
trim <- smru_suggest_trim(wd = dm, config = "pending/config_ct23.json")
trim_zoom(trim, "ct23-C12-06", from = "2007-08-01", to = "2007-08-20 12:00",
          draft = file.path(dm, "ct23_trimIDs_draft.csv"))

## End(Not run)
```

wc_append_ssm *append WC tag datafiles with SSM-estimated locations*

Description

append WC tag datafiles so each event has SSM-derived lon, lat, x, y, x.se, y.se.

Usage

```
wc_append_ssm(
  wc,
  fit,
  what = "p",
  meta,
  cut = FALSE,
  dropIDs = NULL,
  crs = "+proj=merc +units=km +ellps=WGS84 +no_defs"
)
```

Arguments

wc	WC tag datafiles - output of pull_wc
fit	final aniMotum fit object(s)
what	choose which locations to use for annotating WC tag datafiles (default = "predicted")
meta	metadata used to truncate start of diag data for each individual
cut	drop predicted locations if keep = FALSE, ie. locations in a large data gap (currently, only used in DM QC mode)
dropIDs	SMRU DeploymentIDs to be dropped
crs	CRS to be applied when interpolating SSM-estimated locations and re-projecting back from Cartesian coords to longlat

wc_prep_loc *clean, truncate & convert WC Locations files to sf-tibbles*

Description

restructures Locations files, formats dates & lc's; keeps the deployments in the metadata and starts each track at its QC start (the deployment start, T0, or a later manual QC start; see wc_deployment_start()); converts to sf geometry - all in preparation for SSM-filtering. Tracks are not truncated at the first or last dive record.

Usage

```
wc_prep_loc(wc, meta, dropIDs, crs = NULL, program = "atn", QCmode = "nrt")
```

Arguments

wc	list of WC datafiles
meta	metadata; its QC_start_date column sets the start of each track
dropIDs	WC DeploymentID's to be dropped (eg. tags were turned on but not deployed)
crs	a proj4string to re-project diag locations from longlat. Default is NULL which results in one of 4 possible projections applied automatically, based on the centroid of the tracks. See overview vignette for details.
program	specify the aniBOS program contributing data (currently: 'atn', 'irap')
QCmode	specify whether QC is near real-time (nrt) or delayed-mode (dm). In delayed-mode QC, tracks end at a manual QC end or, for ATN, the metadata deployment stop, where given.

---

wc_pull_data *Pull data from Wildlife Computers tag data files*

Description

extracts data from X-Locations.csv (Argos), X-FastGPS.csv, ECDHistos.csv, Histos.csv, MixLayer.csv, PDTs.csv, DSA.csv, MinMaxDepth.csv, HaulOut.csv, and SST.csv files. Extracted data are aggregated across individual tags and returned in a single named list with the following data.frames:

- Argos
- FastGPS
- ECDHistos_SCOUT_TEMP_361A
- ECDHistos_SCOUT_DSA
- Histos
- Mixlayer
- PDTs
- DSA
- MinMaxDepth
- Haulout
- SST WC tag data files downloaded via download_data will be stored in separate, tag-specific subdirectories. path2data should point to the outer directory.

Usage

```
wc_pull_data(path2data, subset.ids = NULL)
```

Arguments

path2data	path to all WC tag data files.
subset.ids	a single column .CSV file of WC UUID's to be included in the QC, with uuid as the variable name.

---

wc_qc *WC tag QC workflow*

Description

Wrapper function that executes the complete workflow from data download to SSM-appended tag data files output as CSV files.

Usage

```
wc_qc(wd, config)
```

Arguments

wd	the path to the working directory that contains: 1) the data directory where tag data files are stored (if source = local); 2) the metadata directory where all metadata files are stored; and 3) the destination directory for QC output.
config	a hierarchical JSON configuration file containing the following blocks, each with a set of block-specific parameters:

- setup config block specifies paths to required data, metadata & output directories:
  - program the national (or other) program of which the data is a part. Current options are: atn, or irap.
  - data.dir the name of the data directory. Must reside within the wd.
  - meta.file the metadata filename. Must reside within the wd. Can be NULL, in which case, the meta config block (see below) must be present & tag-specific metadata are scraped from the SMRU data server.
  - maps.dir the directory path to write diagnostic maps of QC'd tracks.
  - diag.dir the directory path to write diagnostic time-series plots of QC'd lon & lat.
  - output.dir the directory path to write QC output CSV files. Must reside within the wd.
  - return.R logical; should the function return a list of QC-generated objects to the R works pace. This results in a single large object containing the following elements:
    - cid the SMRU campaign ID
    - dropIDs the SMRU Reference ID's droppped from the QC process
    - smru the SMRU tag data tables extracted from the downloaded .mdb file

    * meta the working metadata
    * locs_sf the projected location data to be passed as input to the SSM
    * fit1 the initial SSM output fit object
    * fit2 the final SSM output fit object including re-routed locations if specified.
    * smru_ssm the SSM-annotated SMRU tag data tables. This output object can be useful for troubleshooting undesirable results during supervised or delayed-mode QC workflows.
- harvest config block specifies data harvesting parameters:
  - download a logical indicating whether tag data are to be downloaded from the WC data portal (TRUE) or read from the local data.dir (FALSE).
  - collab.id (optional) the WC data owner ID associated with the data to be downloaded. Ignored (if provided) when harvest$download:FALSE.
  - wc.akey (optional) the WC access key for API access to the data portal. Ignored (if provided) when harvest$download:FALSE.
  - wc.skey (optional) the WC secret key for API access to the data portal. Ignored (if provided) when harvest$download:FALSE.
  - dropIDs the WC UUID(s) for specific tag data set(s) that is/are to be ignored during the QC process. UUID's are supplied as a .CSV file dropIDs.csv with a single variable named uuid. Can be NULL.
  - trimIDs (optional; delayed-mode QC only) a .CSV file of per-deployment track truncation dates, with a ref column holding the DeploymentID and an end_date and/or start_date column (UTC). See trim_locs(). Can be NULL or absent. The QC stops if trimIDs is set and QCmode is not dm.
- model config block specifies model- and data-specific parameters:
  - model the aniMotum SSM model to be used for the location QC - typically either rw or crw.
  - vmax for SSM fitting; max travel rate (m/s) to identify implausible locations
  - time.step the prediction interval (in decimal hours) to be used by the SSM
  - proj the proj4string to be used for the location data & for the SSM-estimated locations. Can be NULL, which will result in one of 5 projections being used, depending on whether the centroid of the observed latitudes lies in N or S polar regions, temperate or equatorial regions, or if tracks straddle (or lie close to) -180,180 longitude.
  - reroute a logical; whether QC'd tracks should be re-routed off of land (default is FALSE). Note, in some circumstances this can substantially increase processing time. Default land polygon data are sourced from the ropensci/rnaturalearthhires R package.
  - dist the distance in km from outside the convex hull of observed locations from which to select land polygon data for re-routing. Ignored if reroute = FALSE.

- barrier the file path (must be within the working directory) for a shapefile to use for the land barrier. If NULL (default) then the default rnaturalearth coastline polygon data is used.
- buffer the distance in km to buffer rerouted locations from the coastline. Ignored if reroute = FALSE.
- centroids whether centroids are to be included in the visibility graph mesh used by the rerouting algorithm. See ?pathroutr::prt_visgraph for details. Ignored if reroute = FALSE.
- ho.gap optional; hours (default 6). Haul-out records less than ho.gap h apart, with no dive record between them, form one on-land interval. SSM-estimated locations within on-land intervals, and locations on land at the start or end of a track, are not re-routed off land. Seals, sea lions and fur seals: every haul-out record is used. Turtles: only haul-out records within 21 days of the deployment start and within 5 km of the deployment position (a return to the nesting beach). Other species: none. The species is taken from the config meta block, or from the deployment metadata. Ignored if reroute = FALSE.
- cut logical; should predicted locations be dropped if they lie within in a large data gap (default is FALSE).
- min.gap the minimum data gap duration (h) to be used for cutting predicted locations (default is 72 h)
- QCmode one of either nrt for Near Real-Time QC or dm for Delayed Mode QC.
- pred.int the prediction interval (h) to use for sub-sampling predicted locations in near real-time QC. Ignored in delayed-mode QC, where predicted locations are output at the SSM time.step.

---

wc_write_csv *write annotated WC tag datafiles & annotated metadata*

Description

for near real-time QC (suffix = "_nrt"), subsample SSM-predicted locations to every ceiling(pred.int / time.step)-th prediction; for delayed-mode and historical QC, output predicted locations at the SSM time.step. Write annotated files to .csv

Usage

```
wc_write_csv(
  wc_ssm,
  fit,
  what,
  meta,
  program = "atn",
  path = NULL,
  dropIDs = NULL,
```

```
  suffix = "_nrt",
  pred.int = 6
)
```

Arguments

wc_ssm	SSM-appended WC tag datafiles - output of append_wc_ssm
fit	final SSM fit object
what	specify whether predicted or rerouted locations are to be used
meta	metadata
program	Determines structure of output metadata. Currently, either imos or atn.
path	path to write .csv files
dropIDs	individual WC DeploymentID's to be dropped
suffix	suffix to add to .csv files (_nrt, _dm, or _hist)
pred.int	prediction interval to use for sub-sampling predicted locations in near real-time QC (default = 6 h). Ignored in delayed-mode and historical QC, where predicted locations are output at the SSM time.step

Index

accept_trim_dates, 3
accept_trim_dates(), *10, 11*

diagnostics, 4
download_data, 5

get_metadata, 7

imos_smru_dm_qc, 8
imos_smru_dm_qc(), *25*

map_QC, 11
multi_filter, 13

pull_local_data, 13

redo_multi_filter, 14

smru_append_ssm, 15
smru_clean_diag, 15
smru_prep_loc, 16
smru_pull_tables, 16
smru_qc, 17
smru_qc(), *8*, *11*
smru_suggest_trim, 20
smru_suggest_trim(), *8*, *10*, *25, 26*
smru_truncate_ssm, 23
smru_write_csv, 23
ssm_mark_gaps, 24

trim_locs, 25
trim_zoom, 25
trim_zoom(), *22*

wc_append_ssm, 27
wc_prep_loc, 27
wc_pull_data, 28
wc_qc, 29
wc_write_csv, 31