Package: ArgosQC (via r-universe)

May 30, 2026

Type Package

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

Version 0.9-16

Date 2026-05-15

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 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 (>= 1.2-15), 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

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 2026-05-15 15:48:23 UTC

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

RemoteRef HEAD

RemoteSha 6eebe98d3a3edc749d6a518adce414ad36a0b260

Contents

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

Description

produces a map of all QC'd tracks and generates various diagnostics to assess QC run

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`	the initial aniMotum fit object from QC process
`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`	the ssm-annotated WC/SMRU tables
`meta`	metadata

lines	add track lines to map (default = FALSE)
obs	add observed locations to map (default = FALSE)
mpath	path to write map file
dpath	path to write all other diagnostic files
QCmode	specify whether QC is near real-time (nrt) or delayed-mode (dm), in latter case start end end of dive data are displayed rather than ctd data.
tag_mfr	the tag manufacturer. Currently, only smru or wc are supported.
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
)
```

Arguments

source	the source of the deployment metadata, current options are imos, 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.

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,
  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.
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; truncates start (and end for "nrt") of individual deployments using ctd dates; converts to sf geometry - all in preparation for SSM-filtering. Splits resulting truncated diag files by species.

Usage

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

Arguments

smru	list of SMRU tables
meta	metadata used to truncate start of diag data for each individual
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), in latter case diag is not right-truncated & date of first dive is used for the track start date

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)
```

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`. 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 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.
  - 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.
  - 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.
- 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")

`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 - subsample predicted locations to 6-h interval, 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"
)
```

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 & atn programs have their own defined metadata structures, all other programs are treated as "Generic".
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)

---

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

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; truncates start (and end for "nrt") of individual deployments using ctd dates; converts to sf geometry - all in preparation for SSM-filtering. Splits resulting truncated diag files by species.

Usage

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

Arguments

wc	list of WC datafiles
meta	metadata used to truncate start of diag data for each individual
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 latter case wc is not right-truncated & date of first dive is used for the track start date.

---

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.
- 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.
  - 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 prior to interpolation of QC'd locations to tag data file event times.

---

wc_write_csv *write annotated WC tag datafiles & annotated metadata*

Description

subsample SSM-predicted locations to 6-h intervals, 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 (default = 6 h)

Index