Received: 19 June 2020 | Accepted: 24 September 2020

DOI: 10.1111/2041-210X.13503

APPLICATION

Methods in Ecology and Evolution BRITISH ECOLOGICAL SOCIETY

# actel: Standardised analysis of acoustic telemetry data from animals moving through receiver arrays

Hugo Flávio | Henrik Baktoft

Freshwater Fisheries and Ecology, Technical University of Denmark, Silkeborg, Denmark

**Correspondence**
Hugo Flávio
Email: hdmfla@aqua.dtu.dk

**Handling Editor:** Edward Codling

## Abstract

1. Acoustic telemetry data often contain erroneous detections, which need to be addressed before further analysis. It is important that this process is systematic, standardised and reproducible.
2. The R package actel provides a systematic conditional pipeline to filter and analyse acoustic telemetry data in a reproducible fashion, for animals moving between receiver arrays.
3. actel outputs a standardised report containing the main results and the steps taken by the user, in addition to summary information for each animal and other content such as efficiency estimations for each receiver array.
4. actel improves the reproducibility of the acoustic data filtering and analysis, and simplifies the integration of acoustic data sets originating from different studies (in time and/or in space).

KEYWORDS

acoustic telemetry, array efficiency, data sorting, migration, reproducibility, stationary receivers

## 1 | INTRODUCTION

The use of acoustic telemetry in ecology studies is expanding rapidly, allowing the recording of multiple variables such as movement, habitat use and survival (Heupel et al., 2006; Thorstad et al., 2013). In particular, researchers are increasingly using stationary arrays of acoustic receivers to track the migration routes and/or residency patterns of tagged animals (e.g. fish, mammals, reptiles, crustaceans) as they pass through partially confined water bodies (e.g. rivers, lakes, fjords; Bowlby et al., 2007; Kristensen et al., 2018; Lothian et al., 2018; Urke et al., 2013) and also through more open coastal or oceanic areas (e.g. Hazel et al., 2013; Reubens et al., 2014). In such studies, certain assumptions can be made about the animal behaviour. For example, migrating animals are expected to move in a given direction; animals are not likely to pass through multiple receiver arrays undetected; and animals are not likely to move over certain speeds. As such, detections that deviate significantly from these expected movement patterns may indicate death, tag shedding, predation or the presence of erroneous detections in the data (e.g. due to background noise; Pincock et al., 2010). These detections are no longer representative of the tagged animal, and therefore must be identified and excluded from the dataset before running further analyses, so that the obtained results are not biased.

Additionally, it is important that the process of acoustic data filtration is reproducible, and that clearly defined rules are applied to all animals. With these needs in mind, we developed a pipeline of conditional tools that allow the user to process acoustic data from arrays of receivers in a systematic and reproducible fashion. This improves the consistency of the analysis and is particularly relevant for researchers comparing results of multiple studies (at different geographical sites, at different times or both). These tools have been wrapped in a set of master functions and implemented in the R package actel.

## 2 | OVERVIEW OF THE ACTEL PACKAGE

The actel package was designed to help researchers analyse telemetry data from animals moving through receiver arrays in a quick,

systematic and reproducible way. By processing the data through a standardised conditional pipeline, actel ensures that all detection data are processed according to the same rules. actel will alert the user to the presence of irregularities in the data, allowing swift action against common errors in the input (e.g. wrongful specification of release times). Furthermore, researchers can choose the actel analysis that better suits their study to obtain the most appropriate results (Table 1). An overview of the functions provided by the actel package is presented in Figure 1.

Importantly, actel can cope with multiple animal groups, multiple release sites, tags that emit multiple signals and complex study areas with multiple waterways and/or barriers (explained in more detail below). actel has been equipped with a detailed set of package vignettes (accessible by running browseVignettes('actel'), or directly through this link), that guide the user from the beginning steps of formatting the input data in an actel -friendly way, to the possibilities unlocked by advanced arguments and functions.

## 2.1 | Preparing the input

actel requires that the input data are set up in a specific fashion. Briefly, four files are essential: (1) a table containing the tag data (i.e. the biometrics file), (2) a table containing the location of the receivers and release sites (i.e. the spatial file), (3) a table containing the receivers deployed at each location (i.e. the deployments file) and (4) either a folder containing the detection files or a single file containing all the detections. Additionally, two optional files can be included: a distances matrix between the spatial points of the study area, and a spatial.txt file indicating how the different arrays connect to each other.

**TABLE 1** The three main analyses provided by the actel package. Each function contains additional arguments that allow the user to convey details of the study area into the analysis process

| Function | Description |
|---|---|
| explore() | • Allows the user to obtain a quick overview of their data<br>• Works with the simplest input<br>• Returns limited results |
| migration() | • Designed for studies where animals are expected to move unidirectionally through the study area<br>• The unidirection assumption allows for Cormack–Jolly–Seber (CJS) efficiency analyses<br>• Animals are assigned as successful or unsuccessful migrants automatically |
| residency() | • Designed for studies where animals are expected to move back and forth within the study area<br>• Computes residence times at the different sections<br>• Automatically generated graphics reveal movement patterns over time<br>• Efficiency estimates are provided, but may be less precise as the analysis cannot rely on the assumption of unidirectional movement |

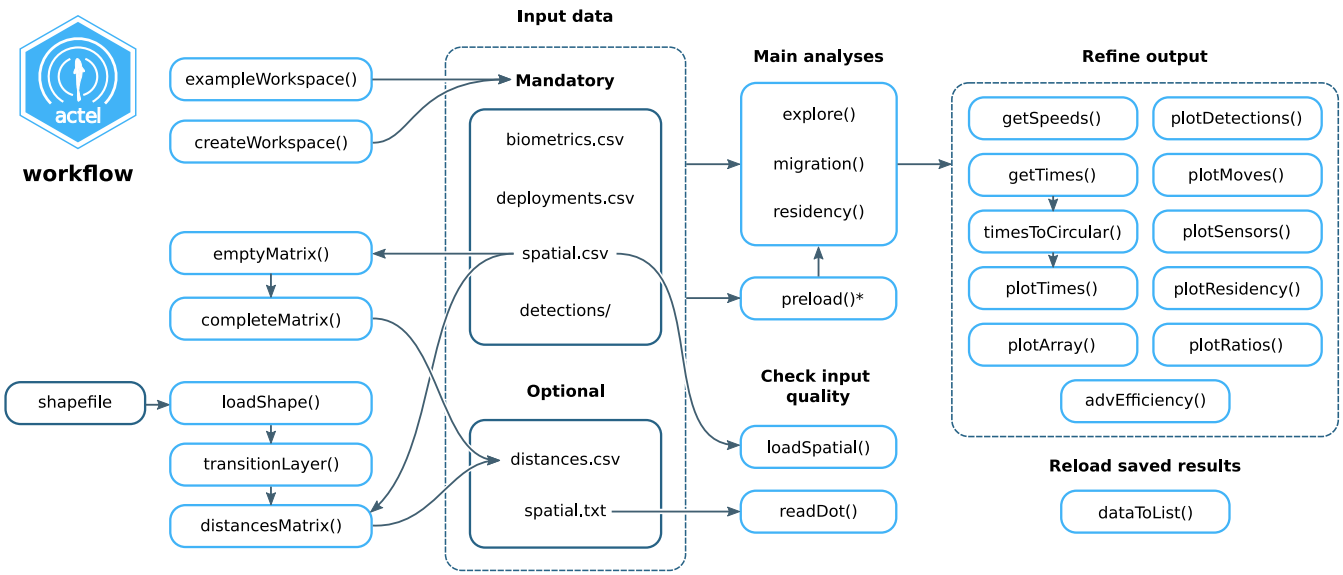


**FIGURE 1** Workflow of actel. * The preload() function can be used to load R objects directly into actel's main functions, without the need to save them as .csv files. The functions loadSpatial() and readDot() allow the user to run quality checks on their spatial data (e.g. column names, column content, input formatting), before attempting to run one of the main analysis. More details on the functions that allow refining the output are provided in a dedicated vignette page (vignette 6.0)

### 2.1.1 | Biometrics

The biometrics file contains information for each tag. This file has two mandatory columns: 'Release.date' and 'Signal'. These define when the animal was released, and the signal(s) emitted by the tag. Note that the 'signal' of a tag is not the same as the tag's serial number. For example, for a Vemco tag 'A69-1303-1234', the signal is '1234'. Tags with multiple signals can then be listed by separating the signals with a '|' (e.g. '1234|1235'). Additional optional columns in this file include 'Release.site', 'Group' and 'Sensor.Unit', which can be used to obtain a more detailed analysis. The release site must match a release site from the spatial file, and the group can be any meaningful group for the study (e.g. 'male' and 'female', 'wild' and 'hatchery', etc.). The user can also include additional columns with data relevant for the study, and these will be carried through the analysis and adjoined to the final results. Additionally, columns whose names contain the keywords 'length', 'mass' or 'weight' will have boxplots automatically drawn per group in the analysis report.

### 2.1.2 | Spatial

The spatial file contains information on the geographical locations where either (a) receivers were deployed, or (b) animals were released. The structure of this input is more rigid than that of the biometrics, with four mandatory columns: 'Station.name', 'Array', 'Section' and 'Type'. There are two types of station: hydrophone stations and release sites. Both stations and release sites can be named freely, but the names must match across the different input files (i.e. biometrics, spatial and deployments). Hydrophone stations are grouped into arrays (functional, isolated units between which the animal can move), and arrays are grouped into sections (study area blocks for which the user would like to compute specific metrics, such as residence time or survival), as exemplified in Figure 2. For the release sites, the 'Array' column should indicate the first array(s) where the released animals are expected to be detected, and the 'Section' column is not used. A vignette page has been dedicated solely to the formatting of the study area (vignette 1.1).

*spatial.txt—Connections between arrays*

If the study area's arrays are not connected linearly (i.e. the study area has multiple water paths), then the user must specify how the arrays connect with each other. The spatial.txt is an auxiliary file that allows the user to transmit this information to the analysis. For example if array A and B are connected (i.e. an animal can move from A to B), then the user must write 'A -- B' in this file. Additionally, the user can specify unidirectional barriers (e.g. 'A -> B', if the animals can move from A to B, but not from B to A). More details on how to configure this file are provided in this section of the manual.

### 2.1.3 | Deployments

The deployments.csv file indicates which receivers were deployed at each station (matching the 'Station.name' in the spatial file), and how long that deployment lasted. This file has four mandatory columns: 'Station.name', 'Receiver', 'Start' and 'Stop', with the last two being the timestamps (in 'yyyy-mm-dd hh:mm:ss' format) of the deployment and recovery of the respective receiver. Multiple receivers can be simultaneously placed at the same station, and the same receiver can be redeployed at different stations over time.

### 2.1.4 | Detections

The offloaded detections can be pasted directly as .csv files into the detections folder. In fact, opening these .csv files and/or editing them is not recommended, as it can introduce unexpected modifications to the file contents (e.g. changing timestamp formats). For example, the user can make use of the arguments *start.time* and *stop.time* to trim the detections that are used for the analysis, thus avoiding editing the detection files manually. actel is able to determine the manufacturer of the hydrophone based on the headings of the .csv file, and from there it is able to import the detections. For experienced data analysts, the possibility is given to use a standard file format, which allows the user to save previously edited detections in a format that actel will comprehend, and is manufacturer-agnostic (details available here).

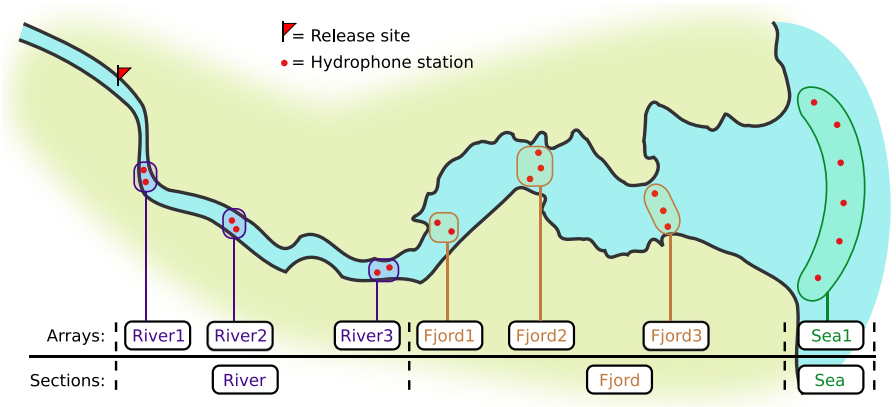


**FIGURE 2** Example of a study area and of the relationship between stations, arrays and sections. Note how the arrays are isolated from each other, creating individual spaces between which the animal can move. An animal is not expected to be within range of two arrays at the same time

### 2.1.5 | Distances matrix

To enable the calculation of movement speeds between arrays, the user must create a distances.csv file, containing the distances between the multiple stations/release sites of the study area. This file must have a very specific formatting and, as such, the steps to obtain it are explained in a dedicated vignette page (vignette 1.2). Currently, actel does not take detection range into consideration when calculating movement speeds. If the arrays are very close to each other (e.g. 2 km apart), this can lead to considerable overestimates of migration speed, if the tag was detected at the very detection range edge of both arrays. This is something the user should keep in mind. The possibility of including receiver detection ranges to the analysis is being planned for a future update.

### 2.1.6 | Creating a template workspace

The user can run the function createWorkspace() to automatically create a directory with template files. The user can then fill these template files with the data relevant for their study. This reduces the risk of errors arising due to formatting mistakes (e.g. wrongly named columns). It is important to note that actel is hardware-agnostic, and no compatibility issues between manufacturers should arise so long as the input data is formatted according to the provided vignettes.

### 2.1.7 | Running the analysis with R objects

Advanced R users may prefer to load their data into R and manipulate it before running the analysis. To avoid having to write all objects into .csv files, the user can call the function preload() to compile a datapack ready to be analysed. This function is explained in detail in vignette 1.3. Note that a compatibility token is generated for each datapack compiled through preload(), and this token will expire once R is restarted.

## 2.2 | Initial data checks

Some preliminary data quality checks are performed by actel before initiating the analyses. Among others, these include checking that (a) the user did not accidentally state that a single receiver was deployed at different places at the same time, (b) there are no detections registered for a tag before the specified release time of their respective animals and (c) there are no duplicated signals in the data (i.e. two tags from different manufacturers and/or different frequencies emitting the same code). If any of these initial tests fails, the user will receive a detailed warning or error message, indicating what needs to be corrected.

## 2.3 | Movement events

actel condensates the detection data into functional events (i.e. groups of detections that indicate movement), which are then used to assess data quality and entry and exit points of each section of the study area. The detections of each individual tag are grouped by receiver array and threshold time (Table 2). This threshold time defaults to 60 min (*max.interval* argument), and determines how much time can pass between detections before these detections are considered separate events. This argument is explained in detail in vignette 2.0. In short, this argument controls how many individual events are generated for a given set of consecutive detections at an array (see the vignette mentioned above for an illustration). Note that the number of sequential events at a given array does not have any impact on the final calculations (so, in essence, *max.interval* = 10 and *max.interval* = 1000 yield the same output in terms of arrival times, residence times, speeds, etc.). However, because a smaller *max.interval* leads to the generation of more events (for sequential detections at the same array), it also allows the user to invalidate smaller groups of detections at a time (because invalidation is performed at the event level). Nevertheless, *max.interval* should always be set to a value that allows some measure of grouping of detections at the same array. For example, *max.interval* should never be lower than the transmission rate of the tags, and we would not recommend setting it to <10 min, as that could generate a massive amount of events, which would be hard to manage during data quality checks.

## 2.4 | Movement checks

actel performs multiple quality checks aimed at revealing potential flaws in the data. For example, by default, tags that only have one movement event with two or less detections are considered invalid

**TABLE 2** Example of a movements table generated based on the detections of a single tag. Timestamps were omitted to reduce the table's size. Time travelling and time in array are represented as hh:mm:ss

| Array | Detections | First station | Last station | First time | Last time | Time travelling | Time in array |
|---|---|---|---|---|---|---|---|
| River1 | 12 | St.1 | St.2 | … | … | 295:44:34 | 0:08:12 |
| River2 | 5 | St.3 | St.3 | … | … | 3:48:39 | 0:04:01 |
| River3 | 1 | St.4 | St.4 | … | … | 17:24:25 | 0:00:00 |
| River4 | 7 | St.5 | St.6 | … | … | 4:05:53 | 0:04:12 |
| River5 | 6 | St.7 | St.7 | … | … | 1:30:15 | 0:03:06 |

**TABLE 3** Movement quality checks performed by actel

| Check | Analysis | Description |
|---|---|---|
| Minimum detections | All | For tags with only one movement event, verifies that the tag was detected a minimum number of times (determined by the argument *minimum.detections*) for the data to be considered reliable |
| Impossible movements | All | Verifies that the tag did not move between arrays that are not connected (e.g. moving over a barrier; more details here) |
| Skipping arrays | All | Verifies how many arrays were skipped between movement events. The behaviour can be controlled using the *jump.warning* and *jump.error* arguments (which default to 2 and 3, respectively; see more details here) |
| Speed | All | Verifies if a tag has moved too quickly between different arrays. Only activated if the user defines at least one of the *speed.warning* or *speed.error* arguments (in m/s, more details here) |
| Inactiveness | All | Verifies if a tag has stayed at the same location for an extended period of time before disappearing. Only activated if the user defines at least one of the *inactive.warning* or *inactive.error* arguments (in days, more details here) |
| Detections outside expected migration path | migration | Verifies if a tag was detected at an array that is not within the expected migration path of the respective animal (more details here) |
| Reversed movements | migration | Verifies if a tag has moved in a reverse direction to what the expected migration path was. Only triggers a warning if the tag moves between sections (i.e. back and forth movements between arrays of the same section are tolerated; more details here) |
| Consecutive section detections | residency | Verifies if a tag has periods of very short section residency (argument *section.minimum* defaults to 2 detections; more details here) |

for further analyses (argument *minimum.detections* defaults to 2). Note that if *max.interval* is set to a value lower than the tag transmission rate, and *minimum.detections* is set to a value higher than one, this fail-safe will never be triggered. A list of all movement quality checks, as well as the arguments that control them, is provided in Table 3.

## 2.5 | Subsequent steps

### 2.5.1 | Fate assignment

actel determines the last location of each animal based on its last valid movement event. In the residency analysis, a barplot is generated showing the number of animals last seen at each study area section. In the migration analysis, actel goes a step further and assigns each animal as a successful or unsuccessful migrant, depending on their last locations. Arrays indicative of successful migration can be personalised through the *success.arrays* argument. The last-seen barplot then shows how many animals disappeared in each section and how many succeeded.

### 2.5.2 | Array efficiency

In the migration analysis, array efficiency is estimated using Cormack–Jolly–Seber (CJS) modelling, as adapted for telemetry arrays by Perry et al. (2012). These efficiency estimates are used to estimate the total number of animals that could have crossed each receiver array and quantify how many recorded fates may differ from the real fates (the method is explained in further detail in vignette 3.3). In the residency analysis, array efficiency is estimated by analysing pairs of movement events at different arrays and determining (a) if the two arrays are neighbours (i.e. the animal can move directly from one to the other) and, if not, (b) which arrays have failed to detect the animal between the two recorded events. Importantly, actel is able to calculate efficiency estimates for non-linear study areas, such as rivers with multiple branches or complex inter-connecting channels. This is made possible by a preliminary assessment of the array disposition (i.e. how are the arrays connected with each other), which is explained in detail in vignette 4.3. Note that, for completely open study areas (e.g. ocean arrays), where an animal can leave an array and arrive at any other array simply by moving around the study area, actel will refrain from calculating efficiency values, simply because all arrays are essentially directly connected to each other.

If the study area has replicated arrays (sensu Perry et al., 2012), the user can specify the receivers composing the replicate array using the *replicates* argument (informative illustrations are available here). This also allows the user to estimate the efficiency of the edge arrays (i.e. arrays placed at the borders of the study area, where an animal may be detected before moving out of the study area). Lastly, any of the above-mentioned efficiency results can be enhanced by running advEfficiency() on the standard efficiency output, which provides the user with a 95% confidence interval over the estimated efficiencies.

### 2.5.3 | Residency times

The residency analysis calculates time spent at the different sections of the study area. By default, this information is compiled at

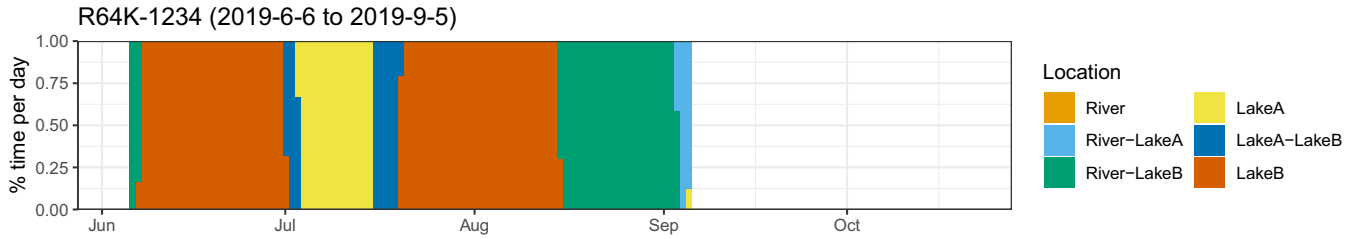


**FIGURE 3** Example of an individual residency plot generated from the output of a residency analysis, using the plotResidency() function. The progression allows the user to quickly identify that the animal moved from lake B to lake A and back, before moving towards the river. The user can also obtain global and group residency plots by using the function plotRatios()

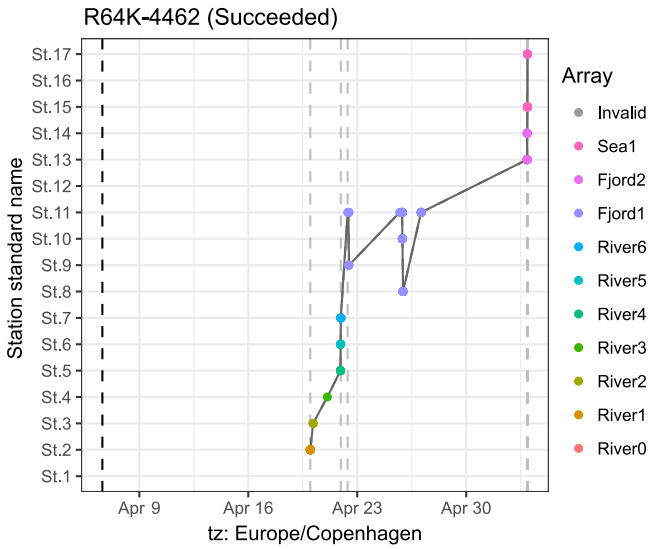


**FIGURE 4** Example of an individual detection plot generated from the output of a migration analysis, using the plotDetections() function. The tag number and respective fate are displayed above the plot. The vertical black dashed line shows the time of release. Vertical grey dashed lines show the assigned moments of entry and exit for each study area section. The user can also compare the movements of multiple animals simultaneously through the function plotMoves()

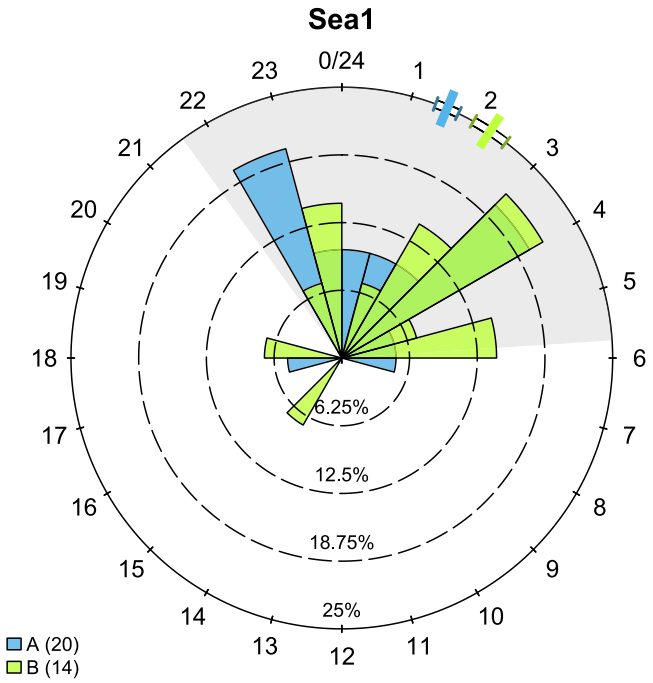


**FIGURE 5** Example of a circular plot generated by the plotTimes() function. This plot shows the arrival times-of-day of the different animals to array 'Sea1', split by the groups defined in the biometrics file. Long coloured dashes at the plot edge indicate the mean value for each group and the respective short dashes show the standard error of the mean. Each groups' bars sum to 100%. The number of animals in each group is presented between brackets in the legend. A shaded area was added using the *night* argument

a daily level, for all the days for which there is valid detection data. However, by setting the argument *timestep* to 'hours' (as opposed to 'days'), the user can obtain hourly residency calculations instead, which provide more detailed results (at the cost of extended computing time). A daily/hourly residency plot is automatically generated for each animal in the report (if *report* = TRUE), and allows the user to quickly identify potential movement patterns over time (e.g. cyclical movements between different study area sections; Figure 3). The processes behind these calculations are explained in further detail in vignette 4.1.

## 2.6 | Analysis output

The three main analyses (explore(), migration() and residency()) return output both in the form of R objects and, if run with the argument *report* = TRUE, in the form of a HTML report. The R output is a list of objects containing information on multiple aspects such as the recorded movements, the array efficiency and the analysis-specific metrics computed. The HTML report details the analysis performed, containing summary information on the study area and input data, the array efficiency, any warning messages displayed or comments left by the user during the analyses, the plots generated for each individual animal (e.g. Figures 3 and 4), among other topics. Circular graphics showing the time of arrival at each array are also produced (Figure 5). The data used in these plots are part of the R output, which allows the users to personalise their plots with the various plot functions provided (e.g. plotMoves() and plotTimes(); see Figure 1 for a full list).

### 2.6.1 | Reproducibility

The HTML report includes a log of the analysis' progress, including the decisions made by the user while running the analysis. This allows the user to re-run the analysis in the same way, and allows other researchers to reproduce the same analysis, if provided with the report. This log is one of the key pieces of actel, as it vastly increases reproducibility of the acoustic data analyses. Note that even if an analysis is run with *report* = FALSE, the user will be given a chance to save the job log as a .txt file.

## 2.7 | Finding and correcting irregularities

Upon analysing the output, the user may decide that actel has missed an irregularity and want to take action manually. In that case, the user has the possibility to list the relevant tags in the *override* argument and re-run the analysis. Doing so will trigger full manual mode for the listed tags. This allows the user to freely decide which movement events are valid or invalid, while bypassing the default checks performed by actel. The individual detection plots for the overridden animals are highlighted in red in the report, and the manual decisions/actions taken are documented in the report to ensure reproducibility.

### 2.7.1 | Example data

To allow users to quickly explore the workflow of actel, the package includes a set of example data that can be deployed and analysed immediately using the function exampleWorkspace(). This function will deploy the example files and present the user with a suggestion of how to run the analysis.

# 3 | CONCLUSIONS

actel facilitates a quick and methodological way of analysing telemetry data for animals moving through receiver arrays. Furthermore, actel allows the user to identify inconsistencies, so that the corresponding input data can be scrutinised and determined as valid or invalid. The flexibility provided by actel ensures that it is applicable to a wide range of scenarios, and can encompass variation in study design and target animal groups. As such, researchers can rely on actel to standardise methods and outputs, which in turn eases the comparison of results from multiple studies. actel has already been successfully used in peer-reviewed works, such as that of Flávio et al. (2019) and Flávio et al. (2020), and also plays a central role in preparing the input data for the recently developed R package RSP (Refined Shortest Paths), as detailed by Niella et al. (2020).

## ACKNOWLEDGEMENTS

We would like to thank the SMOLTRACK group for the data provided for the initial tests of actel and to the researchers who voluntarily helped improve the package by testing it on their data and providing feedback. We would also like to thank Emily Winter for her helpful insights during the development of the residency analysis, and Glen Wightman for providing test data for tags with multiple sensors.

## AUTHORS' CONTRIBUTIONS

H.F. conceived the project and developed the original R functions underlying actel; H.F. and H.B. converted the original functions into an R package; H.F. led the manuscript writing, with the help of H.B.; both H.F. and H.B. approved publication.

## PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/2041-210X.13503.

## DATA AVAILABILITY STATEMENT

The actel package is available on CRAN (https://CRAN.R-project.org/package=actel) and can be installed by running install.packages('actel'). The respective source code is also available at https://github.com/hugomflavio/actel.

## ORCID

*Hugo Flávio* https://orcid.org/0000-0002-5174-1197
*Henrik Baktoft* https://orcid.org/0000-0002-3644-4960

## REFERENCES

Bowlby, H. D., Hanson, J. M., & Hutchings, J. A. (2007). Resident and dispersal behavior among individuals within a population of American lobster *Homarus americanus*. *Marine Ecology Progress Series*, *331*, 207–218.

Flávio, H., Aarestrup, K., Jepsen, N., & Koed, A. (2019). Naturalised Atlantic salmon smolts are more likely to reach the sea than wild smolts in a lowland fjord. *River Research and Applications*, *35*, 216–223.

Flávio, H., Caballero, P., Jepsen, N., & Aarestrup, K. (2020). Atlantic salmon living on the edge: Smolt behaviour and survival during seaward migration in River Minho. *Ecology of Freshwater Fish*. Early View.

Hazel, J., Hamann, M., & Lawler, I. R. (2013). Home range of immature green turtles tracked at an offshore tropical reef using automated passive acoustic technology. *Marine Biology*, *160*, 617–627.

Heupel, M. R., Semmens, J. M., & Hobday, A. J. (2006). Automated acoustic tracking of aquatic animals: Scales, design and deployment of listening station arrays. *Marine and Freshwater Research*, *57*, 1.

Kristensen, M. L., Birnie-Gauvin, K., & Aarestrup, K. (2018). Routes and survival of anadromous brown trout *Salmo trutta* L. post-smolts during early marine migration through a Danish fjord system. *Estuarine, Coastal and Shelf Science*, *209*, 102–109.

Lothian, A. J., Newton, M., Barry, J., Walters, M., Miller, R. C., & Adams, C. E. (2018). Migration pathways, speed and mortality of Atlantic salmon (*Salmo salar*) smolts in a Scottish river and the nearshore coastal marine environment. *Ecology of Freshwater Fish*, *27*, 549–558.

Niella, Y., Flávio, H., Smoothey, A., Aarestrup, K., Taylor, M., Peddemors, V., & Harcourt, R. (2020). Refined shortest paths (RSP): Incorporation of topography in space use estimation from node-based telemetry data. *Methods in Ecology and Evolution*. (in press).

Perry, R. W., Castro-Santos, T., Holbrook, C. M., & Sandford, B. P. (2012). Using mark-recapture models to estimate survival from telemetry

data. In N. Adams, J. Beeman, & J. H. Eiler (Eds.), *Telemetry techniques: A user guide for fisheries research* (pp. 453–475). American Fisheries Society.

Pincock, D., Welch, D., McKinley, S., & Jackson, G. (2010). Acoustic Telemetry for studying migration movements of small fish in rivers and the ocean – Current capabilities and future possibilities. In K. Wolf & J. O'Neal (Eds.), *PNAMP special publication: Tagging, telemetry and marking measures for monitoring fish populations–A compendium of new and recent science for use in informing technique and decision modalities* (pp. 105–117). Pacific Northwest Aquatic Monitoring Partnership – Special Publication.

Reubens, J. T., De Rijcke, M., Degraer, S., & Vincx, M. (2014). Diel variation in feeding and movement patterns of juvenile Atlantic cod at offshore wind farms. *Journal of Sea Research*, *85*, 214–221. https://doi.org/10.1016/j.seares.2013.05.005

Thorstad, E. B., Rikardsen, A. H., Alp, A., & Okland, F. (2013). The use of electronic tags in fish research – An overview of fish telemetry methods. *Turkish Journal of Fisheries and Aquatic Sciences*, *13*, 881–896.

Urke, H. A., Kristensen, T., Ulvund, J. B., & Alfredsen, J. A. (2013). Riverine and fjord migration of wild and hatchery-reared Atlantic salmon smolts. *Fisheries Management and Ecology*, 544–552. https://doi.org/10.1111/fme.12042

**How to cite this article:** Flávio H, Baktoft H. actel: Standardised analysis of acoustic telemetry data from animals moving through receiver arrays. *Methods Ecol Evol*. 2021;12:196–203. https://doi.org/10.1111/2041-210X.13503