Received: 2 July 2021 | Accepted: 14 October 2021

DOI: 10.1111/1365-2656.13610

Journal of Animal Ecology | BRITISH ECOLOGICAL SOCIETY

RESEARCH METHODS GUIDE

A guide to pre-processing high-throughput animal tracking data

Pratik Rajan Gupte[1,2] | **Christine E. Beardsworth**[2] | **Orr Spiegel**[3,4] | **Emmanuel Lourie**[4,5] | **Sivan Toledo**[6,4] | **Ran Nathan**[4,5] | **Allert I. Bijleveld**[2]

[1]Groningen Institute for Evolutionary Life Sciences, University of Groningen, Groningen, The Netherlands; [2]Department of Coastal Systems, NIOZ Royal Netherlands Institute for Sea Research, Den Burg, The Netherlands; [3]School of Zoology, Faculty of Life Sciences, Tel Aviv University, Tel Aviv, Israel; [4]Minerva Center for Movement Ecology, The Hebrew University of Jerusalem, Jerusalem, Israel; [5]Movement Ecology Lab, Department of Ecology, Evolution, and Behavior, Alexander Silberman Institute of Life Sciences, The Hebrew University of Jerusalem, Jerusalem, Israel and [6]Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel

Correspondence
Pratik R. Gupte
Email: pratikgupte16@gmail.com; p.r.gupte@rug.nl

Funding information
Minerva Foundation; Israel Science Foundation, Grant/Award Number: ISF ISF-965/15; Dutch Research Council, Grant/Award Number: VI.Veni.192.051

Handling Editor: Mark Hewison

Abstract

1. Modern, high-throughput animal tracking increasingly yields 'big data' at very fine temporal scales. At these scales, location error can exceed the animal's step size, leading to mis-estimation of behaviours inferred from movement. 'Cleaning' the data to reduce location errors is one of the main ways to deal with position uncertainty. Although data cleaning is widely recommended, inclusive, uniform guidance on this crucial step, and on how to organise the cleaning of massive datasets, is relatively scarce.
2. A pipeline for cleaning massive high-throughput datasets must balance ease of use and computationally efficiency, in which location errors are rejected while preserving valid animal movements. Another useful feature of a pre-processing pipeline is efficiently segmenting and clustering location data for statistical methods while also being scalable to large datasets and robust to imperfect sampling. Manual methods being prohibitively time-consuming, and to boost reproducibility, pre-processing pipelines must be automated.
3. We provide guidance on building pipelines for pre-processing high-throughput animal tracking data to prepare it for subsequent analyses. We apply our proposed pipeline to simulated movement data with location errors, and also show how large volumes of cleaned data can be transformed into biologically meaningful 'residence patches', for exploratory inference on animal space use. We use tracking data from the Wadden Sea ATLAS system (WATLAS) to show how pre-processing improves its quality, and to verify the usefulness of the residence patch method. Finally, with tracks from Egyptian fruit bats *Rousettus aegyptiacus*, we demonstrate the pre-processing pipeline and residence patch method in a fully worked out example.
4. To help with fast implementation of standardised methods, we developed the `R` package `atlastools`, which we also introduce here. Our pre-processing pipeline

and `atlastools` can be used with any high-throughput animal movement data in which the high data-volume combined with knowledge of the tracked individuals' movement capacity can be used to reduce location errors. `atlastools` is easy to use for beginners while providing a template for further development. The common use of simple yet robust pre-processing steps promotes standardised methods in the field of movement ecology and leads to better inferences from data.

KEYWORDS

ATLAS tracking, `atlastools`, big data, biotelemetry, data cleaning, high-throughput movement ecology, residence patch, reverse GPS

1 | INTRODUCTION

The movement of an animal is an adaptive, integrated response to multiple drivers, including internal state, life-history traits and capacities, biotic interactions, and other environmental factors (Holyoak et al., 2008; Nathan et al., 2008). The movement ecology framework links the drivers, processes and fitness outcomes of animal movement (Nathan et al., 2008), and remotely tracking individual animals in the wild is the methodological mainstay of movement ecology (Hussey et al., 2015; Kays et al., 2015; Nathan et al., 2008; Wikelski et al., 2007). A key challenge with observed tracks is to extract information on the behavioural, cognitive, social, ecological and evolutionary processes that shape animal movement. Tracking data, which are observations of a continuous process (animal movement) at discrete timesteps, reveal useful information about the movement process when the tracking interval is considerably shorter than the typical duration of a movement mode (Getz & Saltz, 2008; Nathan et al., 2008; Noonan et al., 2019). This can be accomplished by wildlife tracking systems that collect position data from many individuals at high temporal and spatial resolution (i.e. high-throughput tracking) relative to the scale of the movement mode of interest (Getz & Saltz, 2008). High-throughput tracking technologies include GPS tags (Harel et al., 2016; Klarevas-Irby et al., 2021; Papageorgiou et al., 2019; Strandburg-Peshkin et al., 2015), tracking radars (Horvitz et al., 2014), and computer vision methods for tracking entire groups of animals from video recordings (Pérez-Escudero et al., 2014; Rathore et al., 2020). Furthermore, high-throughput wildlife tracking is routinely provided by terrestrial reverse GPS systems such as ATLAS (Advanced Tracking and Localization of Animals in real-life Systems Toledo et al., 2014; Toledo et al., 2016; Toledo et al., 2020; Weiser et al., 2016, see also MacCurdy et al., 2019; MacCurdy et al., 2009), and underwater acoustic reverse GPS tracking of aquatic animals (Aspillaga et al., 2021; Aspillaga, Arlinghaus, Martorell-Barceló, Follana-Berná, et al., 2021; Baktoft et al., 2017, 2019; Jung et al., 2015). Finally, low-resolution tracking over a long duration may also capture important aspects of animal behaviour at certain time-scales (e.g. migration, long-range dispersal; Getz & Saltz, 2008), thereby being 'relatively' high-throughput.

Although high-throughput tracking provides a massive amount of data on the path of a tracked animal, these data present a challenge to ecologists. When tracking animals at a high temporal resolution, the location error of each position may approach or exceed the true movement distance of the animal, compared to low-resolution tracking with the same measurement error. This leads to an overestimation of the true distance moved by an animal between two discrete time-points, leading to unreliable behavioural metrics ultimately derived from movement distance, such as speed and tortuosity (see Calenge et al., 2009; Hurford, 2009; Noonan et al., 2019; Ranacher et al., 2016). Additionally, the location error around a position introduces uncertainty when studying the relationship between animal movements and either fixed landscape features (e.g. roads), or mobile elements (e.g. other tracked individuals), as well as confounding estimates of habitat selection. Users have two main options to improve data quality, (a) making inferences after modelling the system-specific location error using a continuous time movement model (Aspillaga, Arlinghaus, Martorell-Barceló, Follana-Berná, et al., 2021; Fleming et al., 2014, 2020; Johnson et al., 2008; Jonsen et al., 2003, 2005; Patterson et al., 2008) or (b) pre-processing data to clean it of positions with large location errors (Bjørneraas et al., 2010). The first approach may be limited by the animal movement models that can be fitted to the data (Fleming et al., 2014, 2020; Noonan et al., 2019), may result in unreasonable computation times or may be entirely beyond the computational capacity of common hardware, leading users to prefer data cleaning instead. Data cleaning reveals another challenge of high-throughput tracking: the large number of observations make it difficult for researchers to visually examine each animal's track for errors (Toledo et al., 2020; Weiser et al., 2016). With manual identification and removal of errors from individual tracks prohibitively time-consuming, data cleaning can benefit from automation based on a protocol.

Pre-processing of movement data—defined as the set of data management steps executed prior to data analysis—must reliably discard large location errors, also called outliers, from tracks (analogous to reducing false positives) while avoiding the overzealous rejection of valid animal movements (analogous to reducing false negatives). How well researchers balance these imperatives has consequences for downstream analyses (Stine & Hunsaker, 2001). For instance,

small-scale resource selection functions can easily infer spurious preference and avoidance effects when there is uncertainty about an animal's true position (Visscher, 2006). Ecologists recognise that tracking data are imperfect observations of the underlying movement process, yet they implicitly consider cleaned data equivalent to the ground-truth. This assumption is reflected in popular statistical methods in movement ecology such as Hidden Markov Models (HMMs; Langrock et al., 2012), stationary-phase identification methods (Patin et al., 2020) or step-selection functions (SSFs; Avgar et al., 2016; Barnett & Moorcroft, 2008; Signer et al., 2017), which expect minimal location errors relative to real animal movement (i.e. a high signal-to-noise ratio). This makes the reproducible, standardised removal of location errors crucial to any animal tracking study. While gross errors are often removed by positioning-system algorithms in both GPS and reverse GPS setups, 'reasonable' errors often remain to confront end users (Fischler & Bolles, 1981; Ranacher et al., 2016; Weiser et al., 2016). Furthermore, as high-throughput tracking is deployed in more regions and for more species, standardised pre-processing steps should be general enough to tackle animal movement data recovered from a range of environments, so as to enable sound comparisons across species and ecosystems.

Despite the importance and ubiquity of reducing location errors in tracking data, movement ecologists lack formal guidance on this crucial step. Pre-processing protocols are not often reported in the literature, or may not be easily tractable for mainstream computing hardware and software. Some tracking data, such as GPS, are autonomously pre-processed without user access to the raw data (using error estimates and Kalman smooths; Kaplan & Hegarty, 2005, and substantial location errors may yet persist). However, filtering out positions using estimates of location error alone may not be sufficient to exclude outliers which represent unrealistic movement but have low error measures (Ranacher et al., 2016; Weiser et al., 2016). When tracking systems do make their raw data available to researchers, this can enable users to better control the data pre-processing stage, and to substantially improve data quality while ensuring that cleaning does not itself lead to unrealistic movement tracks (e.g. Kalman smooths which distort tracks, Kaplan & Hegarty, 2005). Furthermore, this makes identifying and removing biologically implausible locations from a track an important component of recovering true animal movement (Bjørneraas et al., 2010). Even after removing unrealistic movement, a track may be comprised of positions that are randomly distributed around the true animal location (Noonan et al., 2019). The large data-volumes of high-throughput tracking allow for a neat solution: tracks can be 'median smoothed' to reduce small location errors that have remained undetected (e.g. Bijleveld et al., 2016) Large data-volumes may also need to be thinned, for example, examining environmental covariates as predictors of prolonged residence in an area (see e.g. Aarts et al., 2008; Bijleveld et al., 2016; Bracis et al., 2018; Harel et al., 2016; Oudman et al., 2018) might require thinning of high-resolution movement data to match the lower spatial resolution of environmental measurements. Data thinning and clustering are also required to avoid non-independent observations due to strong spatiotemporal autocorrelation, or to examine the effect of sampling scale on movement metrics and resource selection (Fleming et al., 2014; Noonan et al., 2019).

When dealing with datasets that contain many millions of positions, researchers may run into computational limits when trying to apply pre-processing steps to their full dataset. For instance, the size of working memory (RAM) limits the size of datasets that can be loaded into `R`, the programming and statistical language of choice in movement ecology (Joo, Boone, et al., 2020; Joo, Picardi, et al., 2020; R Core Team, 2020). Data-rich fields such as genomics inspire a possible solution: to break very large data into smaller subsets, and pass these subsets through automated computational 'pipelines' (Peng, 2011; Schadt et al., 2010). Pre-processing pipelines for animal tracking data—the set of steps that users apply to prepare the data for a specific analysis—come with some additional concerns: (a) identifying which pre-processing steps are necessary and (b) ensuring that these steps reproducibly operate on the data as expected, and as efficiently as possible. While exploratory data analysis and visualisation can help determine how to pre-process the data to maximise the signal-to-noise ratio (Slingsby & van Loon, 2016), standardising implementations of pre-processing techniques into robust, version controlled software packages (e.g. in `R`, see Wickham, 2015), can increase the reliability and reproducibility of animal movement ecology (Archmiller et al., 2020; Haddaway & Verhoeven, 2015; Lewis et al., 2018; Powers & Hampton, 2019). Overcoming hard computational constraints on speed and memory usage for very large data will often require a combination of programming strategies, such as using tools optimised for tabular data, or parallelised processing.

Here, we present guidelines for reproducibly pre-processing high-throughput animal tracking data (Figure 1), with a focus on simple, widely generalisable steps that help improve data quality (Figure 2). We take two important considerations into account that (a) the pre-processing steps should be easily understood and reproduced and (b) our implementations must be computationally efficient and reliable. Consequently, formalising tools as functions in an `R` package would improve portability and reproducibility (Marwick et al., 2018; Wickham, 2015). Using simulated movement tracks, we demonstrate simple yet robust implementations of the pre-processing steps we recommend, conveniently wrapped into the `R` package `atlastools` (Gupte, 2020), with a discussion of features that make these steps more reproducible, and more efficient. We also suggest one potential application of high-throughput tracking in studies of animal movement and space use, illustrated by the first-principles-based synthesis of 'residence patches' from clusters of spatiotemporally proximate positions (sensu Barraquand & Benhamou, 2008; Bijleveld et al., 2016; Oudman et al., 2018). In two fully worked out examples using our package on real tracking data, we show how to apply basic spatiotemporal and data quality filters, how to filter out unrealistic movement and how to reduce the effect of location error with a median smooth. In the first example, using calibration data from an ATLAS system, we show how the residence patch segmentation-clustering method can be used to accurately

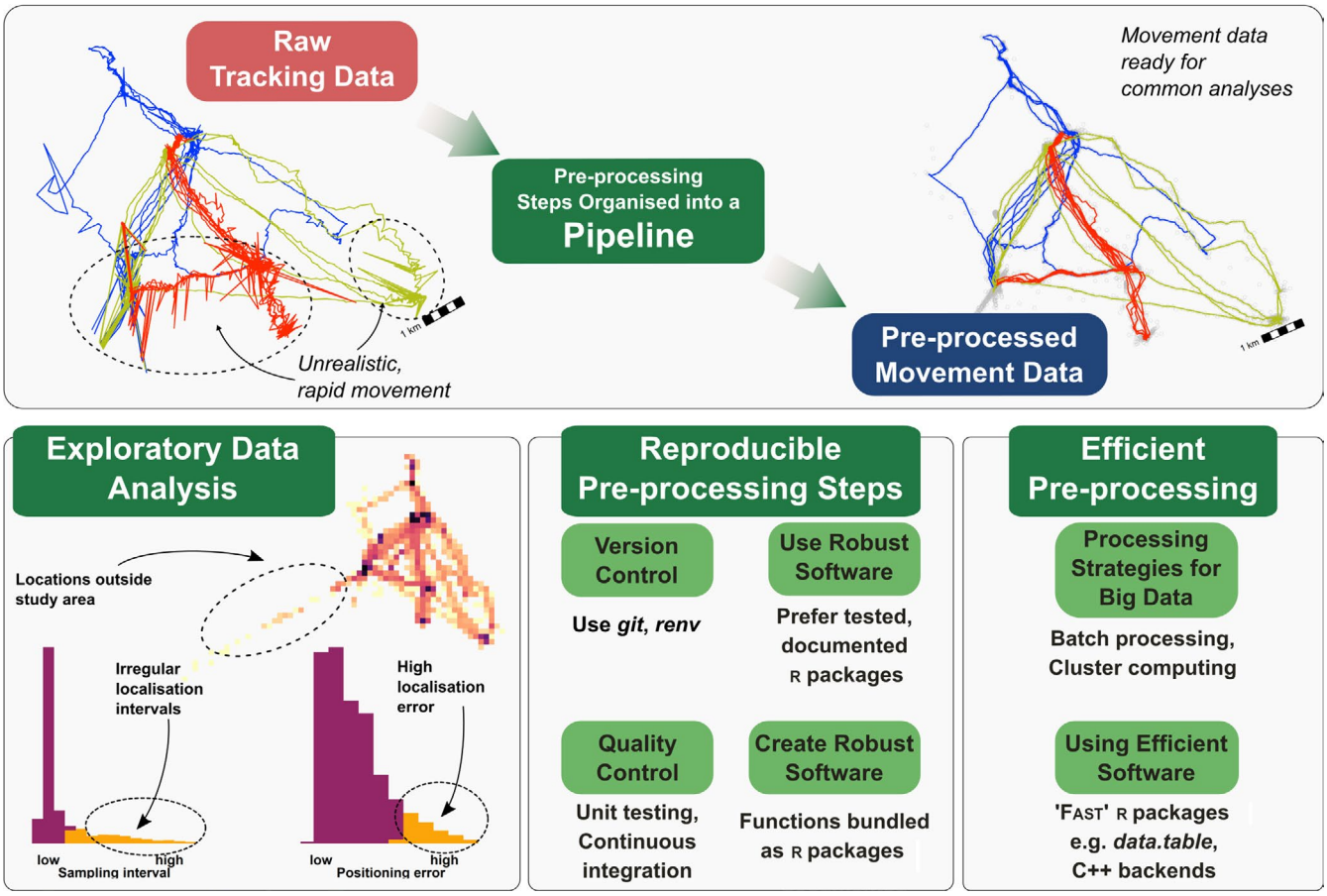

FIGURE 1 Some best practices for pre-processing high-throughput tracking data. Simple pre-processing of animal tracking data can improve the quality of animal tracking data and the inferences that are drawn from it. The organisation of pre-processing workflows into a 'pipeline'—a set of steps that users apply to prepare the data for a specific analysis—can help make research more reproducible and reliable. Exploratory data analysis of representative subsets of the data can help to identify common issues with data quality, and to determine which pre-processing, steps such as filters and smooths, might be necessary (see also Figure 2). Pre-processing steps implemented as programming code can be made reproducible and shareable by following best practices for software development: (1) tracking changes to the steps, and the software used, using version control (e.g. `git`, `renv`), (2) preferring pre-existing tools, such as `R` packages, which are well documented and tested, (3) encapsulating custom-written code as functions, and bundling-related functions into a package, and (4) checking the quality of both custom-written code (e.g. by testing functions) and the overall pipeline (e.g. data visualisation). The efficiency of pre-processing steps can be increased by using strategies for dealing with large datasets, such as batch processing, or using a computing cluster. The use of existing tools optimised for large datasets, or by writing code in a 'fast' language such as `C++`, can also speed up the pre-processing of large datasets (see main text for examples). See the Worked Out Example on Egyptian fruit bats, as well as Supporting Information 1, for more details on implementing pipelines. Figure 2 shows an example of such a pipeline

identify areas of prolonged residence under real field conditions. Finally, in our second example, we use ATLAS data from Egyptian fruit bats *Rousettus aegyptiacus* tracked in the Hula Valley, Israel, to show a fully worked out example of the pre-processing pipeline and the residence patch method. While our approach to high-throughput tracking data, and our package of pre-processing functions was developed with reverse GPS ATLAS systems in mind, both are broadly suitable to a wide range of high-throughput animal tracking data sources, from underwater acoustic reverse GPS (Aspillaga, Arlinghaus, Martorell-Barceló, Barcelo-Serra, et al., 2021; Aspillaga, Arlinghaus, Martorell-Barceló, Follana-Berná, et al., 2021; Baktoft et al., 2017, 2019; Jung et al., 2015), high-resolution GPS (Harel et al., 2016; Klarevas-Irby et al., 2021; Papageorgiou et al., 2019; Strandburg-Peshkin et al., 2015), tracking radars (Horvitz et al., 2014) and visual video tracking (Pérez-Escudero et al., 2014; Rathore et al., 2020).

2 | BEST PRACTICES FOR PRE-PROCESSING WORKFLOWS

2.1 | Exploratory data analysis to identify pre-processing steps

Exploratory data analysis should be the first step towards pre-processing movement data (see Figure 1; Slingsby & van Loon, 2016). Researchers with very large datasets of perhaps millions of rows should ideally select a representative subset of these data for

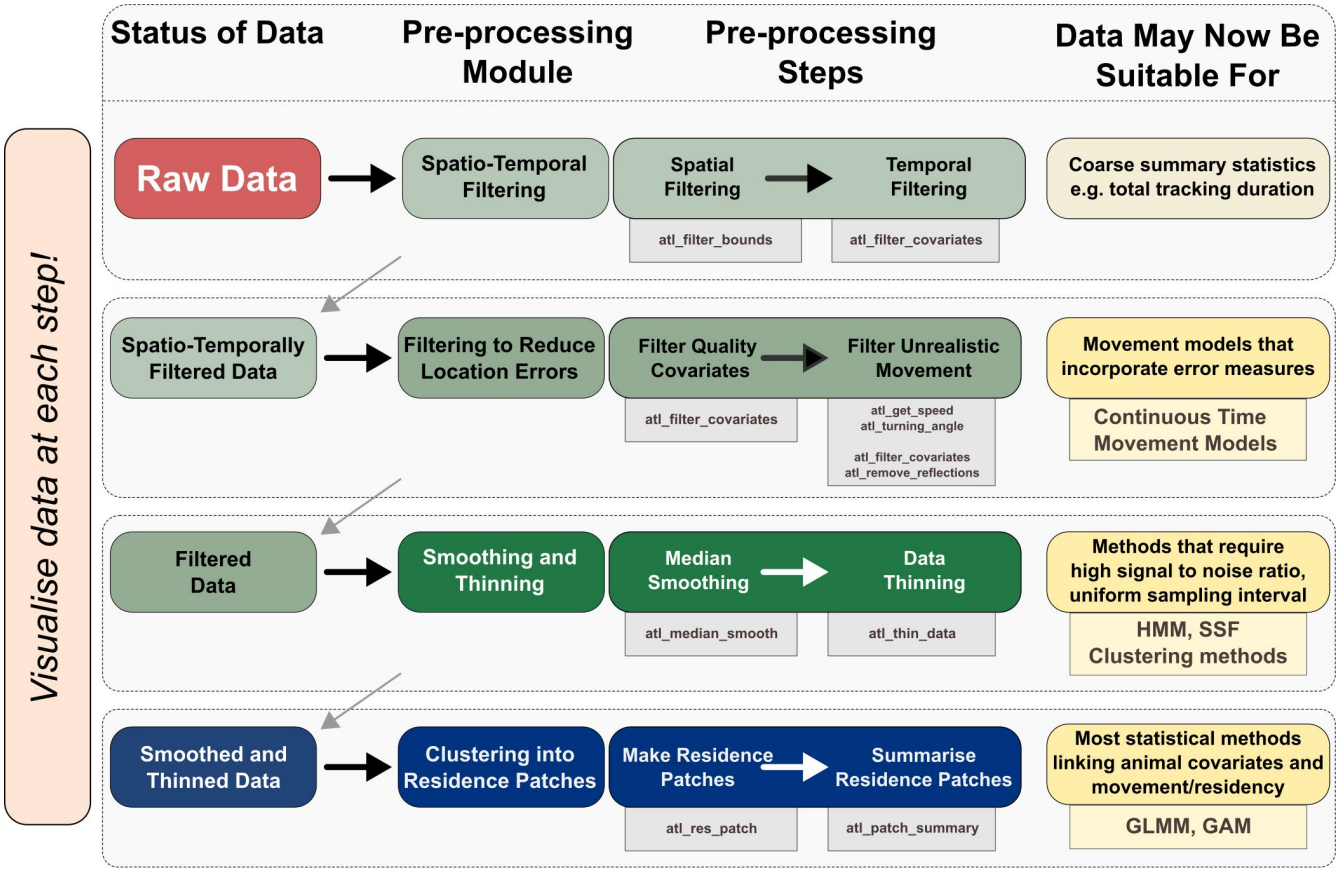

FIGURE 2 An example of a modular pipeline for pre-processing high-throughput tracking data from raw localisations to cleaned data, and optionally into residence patches. Users should apply the appropriate pre-processing modules and the steps therein until the data are suitable for their intended analysis, some of which are suggested here. The `atlastools` function that may be used to implement each pre-processing step is shown in the grey boxes underneath each step. Popular statistical methods are shown underneath possible analyses (yellow boxes). Users are strongly encouraged to visualise their data and scan it for location errors as they work through the pipeline, always asking the question, could the animal plausibly move this way?

exploratory data analysis, including individuals of different species, sexes or seasonal cohorts. Examples of exploratory data analysis include plotting heatmaps of the number of observations per unit area across the study site (Figure 1). Histograms of the location error estimates, plotting the linear approximations of animal paths between observations and histograms of the sampling interval can help determine how data need to be treated so as to minimise location errors and improve computational tractability (Figure 1). While pre-processing steps required for datasets will differ between studies and tracking technologies, we elaborate upon candidate steps and their parameterisation in following sections (see also Figure 2).

2.2 | Improving reliability and reproducibility

Following exploratory data analysis and the parameterisation of data cleaning steps, the specific implementation of these steps should be made reliable and reproducible. Since reproducing pre-processing steps can be challenging when using only written descriptions from published articles, providing the code to implement pre-processing steps reduces ambiguity and increases reproducibility (Haddaway & Verhoeven, 2015). For technically advanced users, the best practices here are (a) to implement pre-processing steps as 'functions', (b) to collect related functions—for example, for similar kinds of data—into a software 'package', (c) to 'test' that the functions handle input as expected and (d) implement 'version control' throughout such that the process of development is documented (Figure 1; Alston & Rick, 2020; Perez-Riverol et al., 2016; Wickham, 2015). As an example, our `atlastools` package incorporates these best practices, and may be used as a reference (Gupte, 2020). We have written each pre-processing step as a separate function, and each of these functions is tested, usually on simulated data, but in some cases also on empirical data (Wickham, 2015, see the directory `tests/` in the associated Zenodo repository). Finally, logging error messages is crucial when passing data through a pipeline, helping determine which data subsets could not be handled as expected, and why. Users who would prefer to rely on pre-existing toolsets and methods can use `R` packages that follow these best practices, such as '`move`' (Kranstauber et al., 2011) and '`sftrack`' (Boone et al., 2020).

2.3 | Improving speed and efficiency

The large size of modern, high-throughput animal tracking data means that the computational challenge can often be *the* main challenge in working with these data. For beginning users, organising their workflows so that they process subsets of the data (such as one individual) at a time can help overcome limitations on working memory. Animal tracking data stored in a relational database (e.g. SQL databases Codd, 1970), for example, can be broken into meaningful subsets based on individual identity and tracking season. These smaller subsets can then be loaded into working memory, pre-processed and saved in a separate location (see Supporting Information 1, Section 2 for a worked out example on an SQL database). Using existing tools optimised for tabular data, such as the `R` package `data.table` (Dowle & Srinivasan, 2020), can also speed up computation; `atlastools` is built using `data.table` for this reason. More advanced users seeking substantial speed gains might wish to look into parallel-processing, and process each subset of the data independently of the full dataset, for example by using a computing cluster (see also Dai, 2021, for an alternative). Finally, another advanced method, used by popular packages such as `move` (Kranstauber et al., 2011) and `recurse` (Bracis et al., 2018), is to write one's own methods in a 'fast' low-level language, such as `C++`, and link these to `R` (Eddelbuettel, 2013, see also `adehabitatLT`, which is written partially in `C`: Calenge, 2006). Beginning practitioners can organise their workflows around these packages to benefit from the features they incorporate.

3 | PRE-PROCESSING STEPS, USAGE AND SIMULATING DATA

3.1 | An overview of pre-processing steps and `atlastools`

In the sections that follow, we lay out pre-processing techniques for raw high-throughput tracking data, and demonstrate working examples of these techniques, which we have collected in the `R` package `atlastools` (see Figure 2). Our package is aimed at getting 'raw data' to the 'analysis' stage identified by Joo, Picardi, et al. (2020) in their review of `R` packages in movement ecology. The package is based on `data.table`, a fast implementation of data frames; thus, it is compatible with a number of data structures from popular packages including `move`, `sftrack` and `ltraj` objects, which can be converted to data frames (Boone et al., 2020; Calenge et al., 2009; Kranstauber et al., 2011). Our package functions are suitable for use with both regularly sampled data, as well as data with missing observations.

We cover, first, the use of simple Spatiotemporal Filters to select positions within a certain time or area. Next, we show how users can Reduce Location Errors by removing unreliable positions based on a system-specific error measure, or by the plausibility of associated movement metrics, such as speed and turning angle (Calenge et al., 2009; Seidel et al., 2018). We then show how users can tackle small-scale location errors by applying a Median Smooth, and users who need uniformly sampled data, can undertake Data Thinning by either aggregation or subsampling. At this stage, the data are ready for a number of popular statistical treatments such as Hidden Markov Model-based classification (Langrock et al., 2012; Michelot et al., 2016). Finally, we show how users wishing simple, efficient segmentation-clustering of points where the animal showed prolonged residence can classify their data into 'residence patches' (Barraquand & Benhamou, 2008; Bijleveld et al., 2016) based on the movement ecology of their study species, after filtering out travelling segments (see System-Specific Pre-Processing Tools).

These pre-processing techniques and package were designed with ATLAS systems in mind, motivated to meet the rapid growth of studies using this high-throughput system worldwide: in Israel (Corl et al., 2020; Toledo et al., 2014, 2016, 2020; Vilk et al., 2021), the UK (Beardsworth, Whiteside, Capstick, et al., 2021; Beardsworth, Whiteside, Laker, et al., 2021) and the Netherlands (Beardsworth, Gobbens, et al., 2021; Bijleveld et al., 2021). However, the principles and functions presented here are ready for use with other massive high-resolution data collected by GPS (e.g. Papageorgiou et al., 2019), reverse GPS (e.g. Aspillaga, Arlinghaus, Martorell-Barceló, Follana-Berná, et al., 2021) or any other high-throughput tracking system. Users may construct a pre-processing pipeline comprising of all the techniques we cover, or implement the modules most suitable for their data. Users are advised to visualise their data throughout their workflow, and especially to perform thorough exploratory data analysis, to check for evident location errors or other issues (Slingsby & van Loon, 2016).

3.2 | Simulating data to demonstrate pre-processing steps

To demonstrate pre-processing steps, we simulated a realistic movement track of 5,000 positions using an unbiased correlated velocity model (UCVM) implemented via the `R` package `smoove` (Gurarie et al., 2017, see Figure 3a). We added four kinds of error to the simulated track: (a) normally distributed small-scale offsets to the X and Y coordinates (small-scale error), (b) normally distributed large-scale offsets to a random subset (0.5%) of the positions (spikes), (c) large-scale displacement of a continuous sequence of 300 of the 5,000 positions (prolonged spikes; indices 500–800) and (d) we removed 10% of the canonical track to simulate missing data (see Figure 3a). To demonstrate the residence patch method, we obtained data, in the form of 1,000 positions, from a mechanistic, individual-based simulation model, in which agents move using simple decision-making rules, and can find high-productivity patches using only ephemeral cues, such as the density of prey items and other competitors (Gupte et al., 2021; Netz & Gupte, 2021). The emergent,

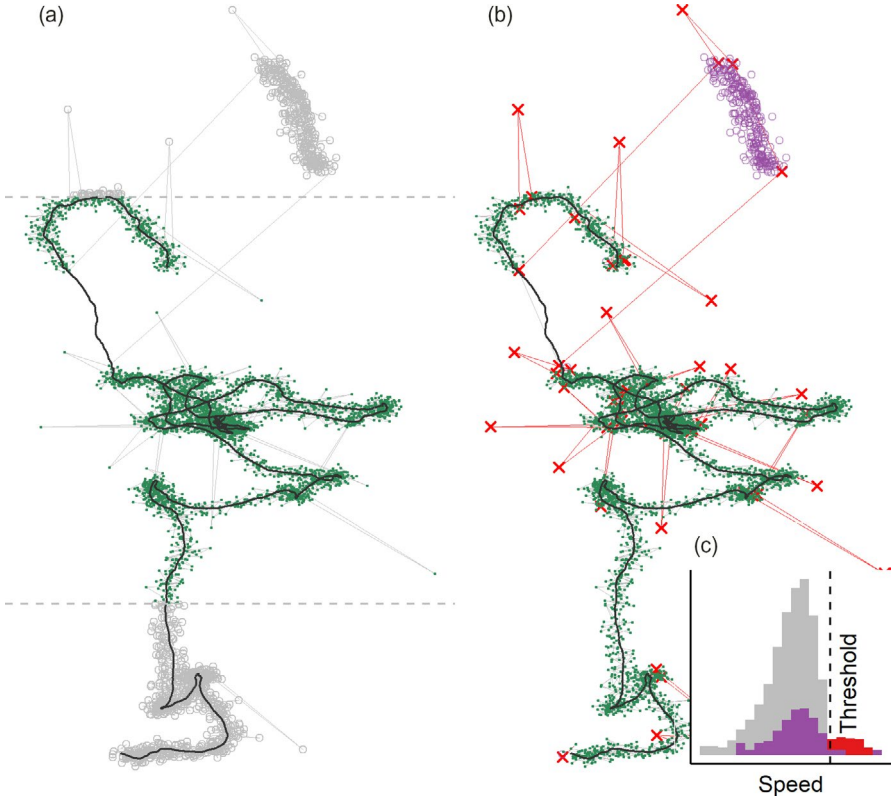

FIGURE 3 Simulated movement data showing four kinds of artificially added errors. (1) Normally distributed small-scale error on each position, (2) large-scale error added to 0.5% of positions, (3) 10% of positions removed to simulate missing data and (4) 300 consecutive positions displaced to simulate a gross distortion affecting a continuous subset of the track. (a) Tracks can be quickly filtered by spatial bounds (dashed grey lines) to exclude broad regions (green = retained; grey = removed). (b) Location error may affect single observations resulting in point outliers or 'spikes' (red crosses and track segments), or continuous subsets of a track, called a 'prolonged spike' (purple circles, top right), and both represent unrealistic movement. (c) Histograms of speed for the track (grey = small-scale errors, red = spikes), and the prolonged spike (purple) show that while spikes could be removed by filtering out positions with both high incoming and outgoing speeds and turning angles, prolonged spikes cannot be removed in this way, and should be resolved by conceptualising algorithms that find the bounds of the distortion instead. Users should frequently check the outputs of such algorithms to avoid rejecting valid data

complex track structure is analogous to the foraging movements of animals, and provides a suitable challenge for the residence patch method and helps to demonstrate its generality.

4 | SPATIOTEMPORAL FILTERING

4.1 | Spatial filtering using bounding boxes and polygons

First, users should exclude positions outside the spatial bounds of a study area by comparing position coordinates with the range of acceptable coordinates (the bounding box) and removing positions outside them (Figure 3a; Listing 1). A bounding box filter does not require a geospatial representation, such as a shapefile, and can help remove unreliable data from a tracking system that is less accurate beyond a certain range (Beardsworth, Gobbens, et al., 2021). In some special cases, users may wish to remove positions *inside* a bounding box, either because movement behaviour within an area is not the focus of a study or because positions recorded within an area are known to be erroneous. An example of the former is studies of transit behaviour between features which can be approximated

```
filtered_data <- atl_filter_bounds(
                          data = data,
                          x = "X", y = "Y",
                          x_range = c(x_min, x_max),
                          y_range = c(y_min, y_max),
                          sf_polygon = your_polygon,
                          remove_inside = FALSE
                        )
```

LISTING 1 The `atl_filter_bounds` function filters on an area defined by coordinate ranges, a polygon, or all three; it can remove positions outside (`remove_inside = FALSE`), or within the area (`remove_inside = TRUE`). The arguments `x` and `y` determine the X and Y coordinate columns, `x_range` and `y_range` are the filter bounds in a coordinate reference system in metres, and the data can be filtered by an `sf-(MULTI)POLYGON`, which can be passed using the `sf_polygon argument`. The output is a `data.table`, which must be saved as an object (`here, filtered_data`.)

by their bounding boxes. Instances of the latter are likely to be system specific, but are known from ATLAS systems. Bounding boxes are typically rectangular, and users seeking to filter for other geometries, such as a circular or irregularly shaped study area, need a geometric intersection between their data and a spatial representation of the area of interest (e.g. shapefile, geopackage or `sf`-object in `R`). The `atlastools` function `atl _ filter _ bounds` implements both bounding box and explicit spatial filters, and accepts X and Y coordinate ranges, an `sf`-polygon or multi-polygon object (Pebesma, 2018), or any combination of the three to filter the data (Listing 1).

4.2 | Temporal and spatiotemporal filters

Tracking data might fail to properly represent an animal's movement at certain times, for instance, data recorded before release, or data from shortly after release when the animal is still influenced by the stress of capture and handling. Periods of poor tracking quality may result from system malfunctions and unusual disturbances, and users may wish to exclude these data as well. Temporal filtering can exclude positions from intervals when data are expected to be unreliable for ecological inference, either due to abnormal movement behaviour or system-specific issues. Temporal filters can be combined with spatial filters to select specific time-location combinations. For example, studies of foraging behaviour of a nocturnal animal would typically exclude tracking data from the animal's daytime roosts (see Worked Out Example). Users should apply filters in sequence rather than all at once and visualise the output after each filtering step ('sanity checks'; see Supporting Information Section 2). The

```
night_data <- atl_filter_covariates(
                    data = dataset,
                    filters = c(
                        "!inrange(hour, 6, 18)",
                        "between(x, x_min, x_max)"
                    )
                )

filtered_data <- atl_filter_covariates(
                        data = data,
                        filters = c(
                            "NBS > 3",
                            "SD < 100",
                            "between(day, 5, 8)"
                        )
                    )
```

LISTING 2 Data can be filtered by a temporal or a spatiotemporal range using `atl _ filter _ covariates`. Filter conditions are passed to the `filters` argument as a character vector. Only rows in the data satisfying *all* the conditions are retained. Here, the first example shows how nighttime data can be retained using a predicate that determines whether the value of 'hour' is between 6 and 18, and also within a range of X coordinates. The second example retains ATLAS locations calculated using >3 base stations (`NBS`), with location error (`SD`) < 100, and data between an arbitrary day 5 and day 8

`atlastools` function `atl _ filter _ covariates` allows convenient filtering of a dataset by any number of logical statements, including querying data within a spatiotemporal range (Listing 2). The function keeps only those data which satisfy each of the filter conditions, and users must ensure that the filtering variables exist in their dataset to avoid errors.

5 | FILTERING TO REDUCE LOCATION ERRORS

5.1 | Filtering on data quality attributes

Tracking data attributes can be good indicators of the reliability of positions calculated by a tracking system (Beardsworth, Gobbens, et al., 2021). GPS systems provide direct measures of location error during localisation (Ranacher et al., 2016, Horizontal Dilution of Precision, HDOP in GPS), while in reverse GPS systems, a measure referred to as Standard Deviation (SD in many datasets), can be calculated from the variance–covariance matrix of each position as: $SD = \sqrt{VarX + VarY + CovXY}$ (see details in MacCurdy et al., 2009, 2019; Ranacher et al., 2016; Weiser et al., 2016). Tracking data can also include indirect indicators of data quality. For instance, GPS systems' location error may be indicated indirectly by the number of satellites involved in the localisation. In reverse GPS systems too, the number of base stations involved in each localisation is an indirect indicator of data quality, and positions localised using more receivers are usually more reliable (the minimum required for an ATLAS localisation is 3; see Beardsworth, Gobbens, et al., 2021; Weiser et al., 2016). Unreliable positions can be removed by filtering on direct or indirect measures of quality using `atl _ filter _ covariates` (Listing 2). While filtering on direct quality attributes and unrealistic movement speeds (see below) will often be sufficient, filtering on indirect quality indicators is a strategy to consider when direct error measures are not available.

5.2 | Filtering unrealistic movement

Filtering on system-generated attributes may not remove all erroneous positions, and the remaining data may still include biologically implausible movement. Users are encouraged to visualise their tracks before and after filtering point locations, and especially to 'join the dots' and connect consecutive positions with lines (Figure 3b). Whether the resulting track looks realistic is ultimately a subjective human judgement, but any decision to filter-out data must remain independent of the hypothesised movement behaviour. This basic principle does not preclude explicitly integrating prior knowledge of the movement ecology of the study species to ask, 'Does the animal move this way?' Segments which appear to represent unrealistic animal movement are often obvious to researchers with extensive experience of the study system (the non-movement approach; see Bjørneraas et al., 2010). Since it is both difficult and prohibitively

time-consuming to exactly reproduce expert judgement when dealing with large volumes of tracking data from multiple individuals, some automation is necessary. Users should first manually examine a representative subset of tracks and attempt to visually identify problems—either with individual positions or with subsets of the track—that persist after filtering on system-generated attributes. Once such problems are identified, users can conceptualise algorithms that can be applied to their data to resolve them.

A common example of a problem with individual positions is that of point outliers or 'spikes' (Bjørneraas et al., 2010), where a single position is displaced far from the track (see Figure 3b). Point outliers are characterised by artificially high speeds between the outlier and the positions before and after (called incoming and outgoing speed, respectively; Bjørneraas et al., 2010), lending a 'spiky' appearance to the track. Removing spikes is simple: remove positions with extreme incoming and outgoing speeds. Users must first define plausible upper limits of the study species' speed (Calenge et al., 2009; Seidel et al., 2018). Here, it is important to remember that speed estimates are scale dependent; high-throughput tracking typically overestimates the speed between positions where the animal is stationary or moving slowly, due to small-scale location errors (Noonan et al., 2019; Ranacher et al., 2016). Even after data with large location errors have been removed, it is advisable to begin with a liberal (high) speed threshold that excludes only the most unlikely speeds. Estimates of maximum speed may not always be readily obtained for all species, and an alternative is to use a data-driven threshold such as the 90th percentile of speeds from the track. Once a speed threshold S has been chosen, positions with incoming *and* outgoing speeds > S may be identified as spikes and removed.

Some species can realistically achieve speeds > S in fast transit segments when assisted by their environment, such as birds with tailwinds, and a simple filter on incoming and outgoing speeds would exclude this valid data. To avoid removing valid, fast transit segments while still excluding spikes, the speed filter can be combined with a filter on the turning angles of each position (see Bjørneraas et al., 2010; Calenge et al., 2009). This combined filter assumes that positions in high-throughput tracking with both high speeds and large turning angles are likely to be due to location errors, since most species are unable to turn sharply at very high speed. Users can then remove those positions whose incoming and outgoing speeds are both > S, and where $\theta > A$ (sharp, high-speed turns), where θ is the turning angle and A is the turning angle threshold. Many other track metrics may be used to identify implausible movement and to filter data (Seidel et al., 2018). At this early stage in pre-processing, track metrics should be considered provisional—it is not until after smoothing and potentially resampling to a regular interval (see below) that calculated track metrics should be used for ecological inference. We show an implementation of spike removal using the `atl_filter_covariates` function (Listing 3).

Sometimes, entire subsets of the track may be affected by the same large-scale location error. For instance, multiple consecutive positions may be roughly translated (geometrically) away from the real track and form 'prolonged spikes' or 'reflections' (see Figure 3b). These cannot be corrected by targeted removal of individual positions, as in Bjørneraas et al.'s approach (2010), since there are no positions with both high incoming and outgoing speeds, as well as sharp turning angles that characterise spikes. Since filtering individual positions will not suffice, algorithms to correct such errors must take a track-level view, and target the displaced sequence overall. Track-subset algorithms are likely to be system specific and may be challenging to conceptualise or implement. In the case of prolonged

```
data$speed_in <- atl_get_speed(
                    data,
                    x = "x", y = "y",
                    time = "time",
                    type = c("in")
                )

data$angle <- atl_turning_angle(
                    data,
                    x = "x", y = "y",
                    time = "time"
                )

filtered_data <- atl_filter_covariates(
                    data = data,
                    filters = c(
                        "(speed_in < S & speed_out < S) | angle < A"
                    )
                )
```

LISTING 3 Filtering a movement track on incoming and outgoing speeds, and on turning angle to remove unrealistic movement. The functions `atl_get_speed` and `atl_turning_angle` are used to get the speeds and turning angles before filtering, and assigned to a column in the data (assignment of `speed_out` is not shown). The filter step only retains positions with speeds below the speed threshold S *or* angles above the turning angle threshold θ, that is, positions where the animal is slow but makes sharp turns, and data where the animal moves quickly in a relatively straight line

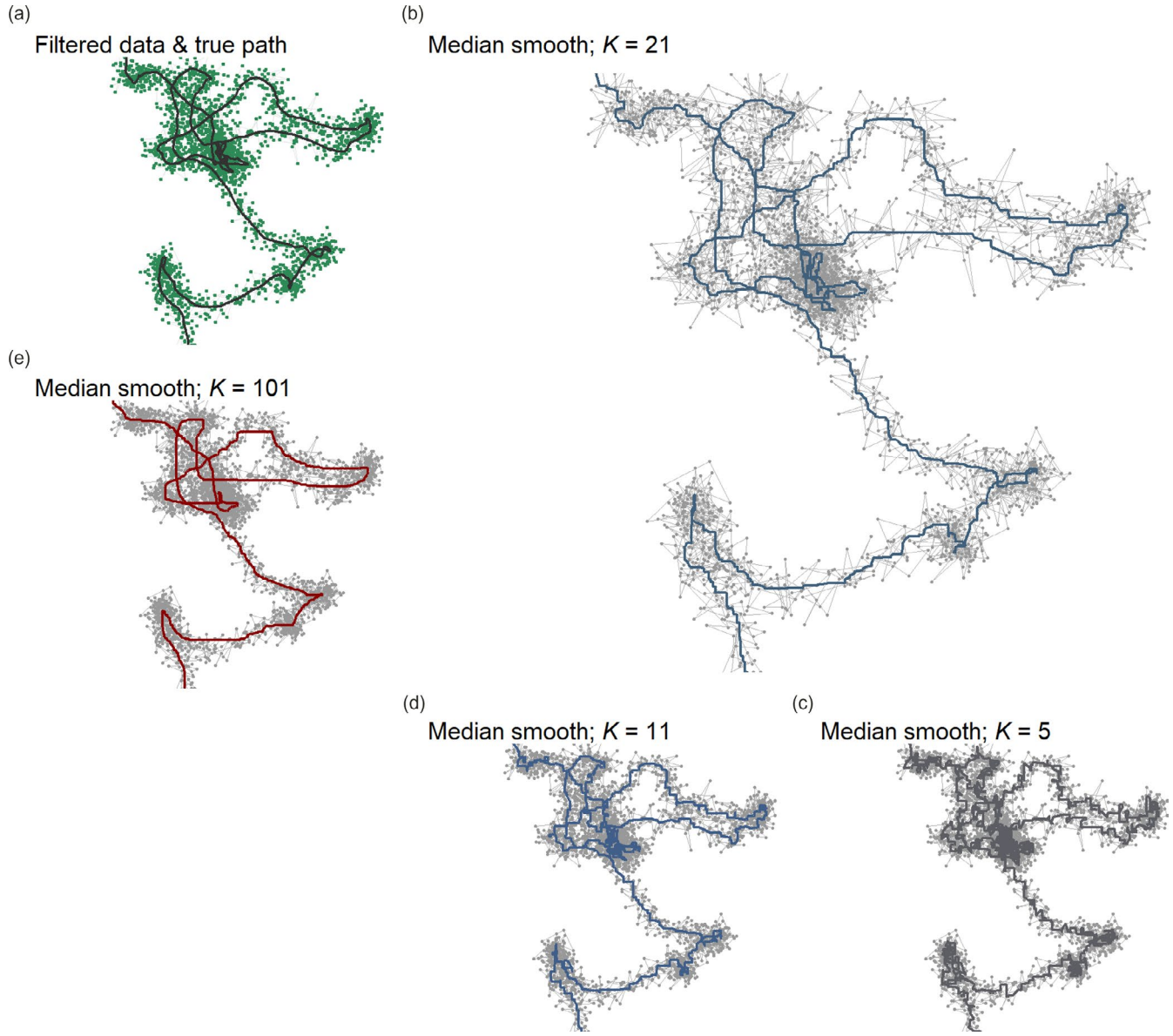

FIGURE 4 Median smoothing position coordinates reduces small-scale location error in tracking data. The goal of this step is to approximate the simulated canonical track (black line, (a)), given positions with small-scale error that remains after filtering in previous steps (green points). (b) Median smoothing the position coordinates (green points, in (a)) over a moving window (*K*) of 21 positions gives a good approximation (blue line) of the canonical track, and is a significant improvement on the unsmoothed track (grey lines and points). While *K* should usually be at least two orders of magnitude less than the number of positions in the track, users are cautioned that there is no correct *K*, and they must subjectively choose a *K* which most usefully trades small-scale details of the track for large-scale accuracy. Here, smoothing with a *K* of (c) 5 (dark grey line) and (d) 11 (blue line) leads to a jagged track, compared to the true path in (a), and the distance moved by the animal would be overestimated. (e) Using extremely large values of *K* (101) may lead to a loss of both large- and small-scale detail (red line). Across panels, grey lines and points show the track without smoothing

spikes, one relatively simple solution is identifying the bounds of displaced segments, and removing positions between them. This identification can be based on relatively simple rules—for example, the beginning of a prolonged spike could be identified as a position with a high *incoming* speed, but a low *outgoing* speed, while the end of such a spike would have a low incoming, but a high outgoing speed. We have implemented an illustrative example of such an algorithm in the form of track-subset filtering for prolonged spikes using the `atlastools` function `atl_remove_reflections` (see the `atlastools` documentation for details on the algorithm). Users are strongly encouraged to visualise their data before and after applying such algorithms; as these methods are not foolproof, and data that are heavily distorted by errors affecting entire track-subsets should be used with care when making further inferences.

6 | SMOOTHING AND THINNING DATA

6.1 | Median smoothing

After filtering out large location errors, the track may still look 'spiky' at small scales, and this is due to smaller location errors that are especially noticeable when the individual is stationary or moving slowly (Noonan et al., 2019). These smaller errors are challenging to remove since their attributes (such as speed and turning angles) are within the expected range of movement behaviour for the study species. The large data-volumes of high-throughput tracking allow users to resolve this problem by smoothing the positions. The most basic 'smooths' work by approximating the value of an observation based on neighbouring values. For a one-dimensional series of observations, the neighbouring values are the K observations centred on each index value i. The range $i-(K-1)/2 \ldots i+(K-1)/2$ is referred to as the moving window as it shifts with i, and K is the moving window size. A common smooth is nearest neighbour averaging, in which the value of an observation x_i is the average of the moving window K. The median smooth is a variant of nearest neighbour averaging which uses the median rather than the mean, and is more robust to outliers (Tukey, 1977). The median smoothed value of the X coordinate, for instance, is

$$X_i = \text{Median}(X_{i-(K-1)/2} \ldots X_{i+(K-1)/2}).$$

Users can apply a median smooth with an appropriate K independently to the X and Y coordinates of a movement track to smooth it (see Figure 4a–e). The median smooth is robust to even very large temporal and spatial gaps, and does not interpolate between positions when data are missing. Thus, it is not necessary to split the data into segments separated by periods of missing observations when applying the filter (see Figure 4).

Some data sources, such as GPS, provide tracks that have already been smoothed in quite sophisticated ways, such as with a Kalman filter, making a median smooth unnecessary (Kaplan & Hegarty, 2005). Furthermore, smoothing is not a panacea for data quality issues, and has its drawbacks. Smoothing does not change the number of observations, but does decouple the coordinates from some of their attributes. For instance, smoothing breaks the relationship between a coordinate and the location error estimate around it (VARX, VARY and SD in ATLAS systems). Since the X and Y coordinates are smoothed independently, the smoothed coordinates of an observation will likely differ from all the coordinates used to compute the smoothed value. Any position covariates (e.g. environmental values such as landcover or elevation) obtained before smoothing should be replaced with the covariates obtained at the smoothed coordinates. Similarly, instantaneous track metrics, such as speed and turning angle, should also be updated at this stage to reflect the smoothed coordinates. Furthermore, the location error estimate around each coordinate, and around the localisation overall, become invalid and should be ignored. This makes subsequent filtering on measures of data quality unreliable, and smoothed data are unsuitable for use with methods that model location uncertainty (Calabrese et al., 2016; Fleming et al., 2014, 2020; Noonan et al., 2019). Thus, when applying location error modelling methods, users should ensure that the error measure bears a mechanistic relationship with the location estimate (see Fleming et al., 2020; Noonan et al., 2019, for more details). Additionally, excessively large K may result in a loss in detail of the individual's small-scale movement (compare Figure 4e with 4a). Users must themselves judge how best to balance large-scale and small-scale accuracy, and choose K accordingly. Median smoothing is provided by the `atlastools` function `atl_median_smooth`, with the only option being the moving window size, which must be an odd integer (Listing 4).

```
atl_median_smooth(
    data = track_data,
    x = "x", y = "y",
    time = "time",
    moving_window = 5
)
```

LISTING 4 Median smoothing a movement track using the function `atl_median_smooth` function with a moving window $K = 5$. Larger values of K yield smoother tracks, but K should always be some orders of magnitude lower than the number of observations

6.2 | Thinning movement tracks

Most data at this stage are technically 'clean', yet the volume alone may pose challenges for lower specification or older hardware and software if these are not optimised for efficient computation. Thinning data, that is, reducing their volume, need not compromise researchers' ability to answer ecological questions; for instance, proximity-based social interactions lasting 1–2 min would still be detected on thinning from a sampling interval of 1 s to 1 min (Aspillaga, Arlinghaus, Martorell-Barceló, Barcelo-Serra, et al., 2021). Thinning data also do not imply that efforts to collect high-throughput movement data are 'wasted', as rich movement datasets enable more detailed and more accurate representation of the true track, as elaborated above. Indeed, some analyses require that temporal autocorrelation in the data be broken by subsampling the data to a lower resolution; these include traditional kernel density estimators for animal home-range, as well as resource selection functions (Dupke et al., 2017; Fleming et al., 2014; Manly et al., 2007). Furthermore, a number of powerful methods in movement ecology, including Hidden Markov Models and integrated Step-Selection Analysis, recommend uniform sampling intervals (Avgar et al., 2016; Langrock et al., 2012; Michelot et al., 2016). Finally, subsampling data may be an important strategy in exploratory data analysis; for instance, it allows researchers to determine whether computationally intensive methods, such as distance and speed estimates from continuous time movement model fitting, are required for their data, or whether the movement metrics stabilise at a certain time scale (Noonan et al., 2019). Two

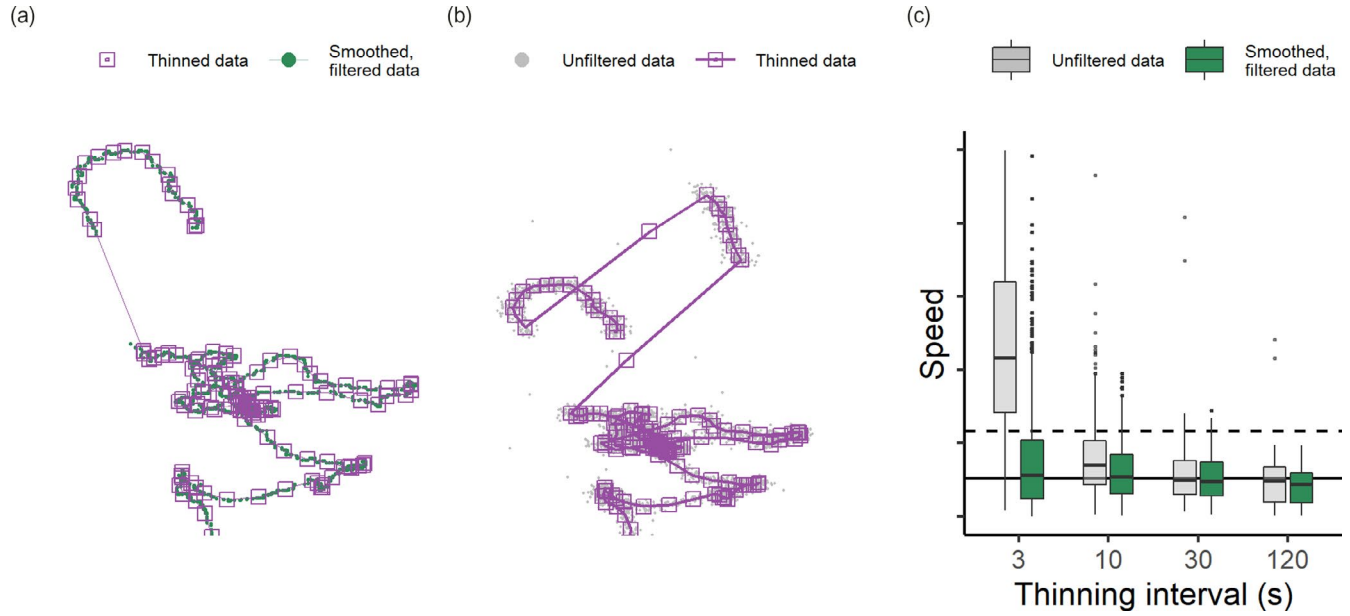

FIGURE 5 Thinning tracking data can aid computation but must be approached carefully. Aggregating a filtered and smoothed movement track (a) preserves track structure while reducing data-volume, but (b) aggregating before filtering gross location errors and unrealistic movement leads to the persistence of large-scale errors (such as prolonged spikes). (c) Thinning before data cleaning can lead to significant mis-estimations of essential movement metrics such as speed at lower intervals. Boxplots show the median and interquartile ranges for speed estimates of tracks aggregated over intervals of 3, 10, 30 and 120 s. For comparison, the median and 95th percentile of speed of the canonical track are shown as solid and dashed horizontal lines, respectively

```
thinned_data <- atl_thin_data(
                    data,
                    interval = 60,
                    id_columns = c("animal_id"),
                    method = "aggregate"
                )
```

LISTING 5 Code to thin data by aggregation in `atlastools`. The method can be either 'aggregate' or 'subsample'. The time interval is specified in seconds, while the `id _ columns` allows a character vector of column names to be passed to the function, with these columns used as identity variables. Both methods return a dataset with one rows per time-interval

plausible approaches here are subsampling and aggregation, and both approaches begin with identifying time-interval groups (e.g. of 1 min). Subsampling picks one position from each time-interval group while aggregation involves computing the mean or median of all system-generated attributes for positions within a time-interval group. Both approaches yield one position per time-interval group (Figure 5a). Categorical variables, such as the habitat type associated with each position, can be aggregated using a suitable measure such as the mode. We caution users that thinning causes an extensive loss of small-scale detail in the data and should be used carefully.

Both aggregation and subsampling have their relative advantages. The aggregation method is less sensitive to selecting point outliers by chance than subsampling. However, to account for location error with methods such as state-space models (Johnson et al., 2008; Jonsen et al., 2003, 2005) or continuous time movement models (Calabrese et al., 2016; Fleming et al., 2014, 2020; Gurarie et al., 2017; Noonan et al., 2019), correctly propagating the location error is important, and subsampling directly propagates these errors without further processing. In the aggregation method, the location error around each coordinate provided by either GPS or reverse GPS systems can be propagated—assuming the errors are normally distributed—to the averaged position as the sum of errors divided by the square of the number of observations contributing to each average (N):

$$\mathrm{Var}(X)_{\mathrm{agg}} = \left(\sum_{i=1}^{i=N} \mathrm{Var}\, X_i \right) / N^2$$

Similarly, the overall location error estimate for the average of N positions in a time interval can be calculated by treating it as a variance. For instance, the ATLAS error and GPS error measures (SD and HDOP, respectively) can be aggregated as:

$$\mathrm{SD}_{\mathrm{agg}} \text{ or } \mathrm{HDOP}_{\mathrm{agg}} = \sqrt{\left(\sum_{i=1}^{i=N} \mathrm{SD}_i^2 \text{ or } \mathrm{HDOP}_i^2 \right) / N^2}.$$

Users may question why thinning, which can obtain consensus positions over an interval and also reduce data-volumes should not be used directly on the raw data. We caution that thinning prior to excluding unrealistic movement and smoothing (Figure 5b) can lead to preserving artefacts in the data, and estimates of essential metrics—such as straight-line displacement (and hence, speed)—that

are substantially different from the true value (see Figure 5c; Noonan et al., 2019). In our example, the data with errors would have to be thinned to 130th of its volume for the median speed of the thinned data to be comparable with the overall median speed—this is an undesirable step if the aim is fine-scale tracking. Additionally, the optimal level of thinning can be difficult to determine, especially if there is wide individual variation in movement behaviour, and the mis-estimation of track metrics from inappropriately thinned data could have consequences for the implementation of subsequent filters based on detecting unrealistic movement. However, thinning before data cleaning has its place as a useful step before exploratory visualisation of the movement track, since reduced data-volumes are easier to handle for plotting software. Thinning is implemented in `atlastools` using the `atl_thin_data` function, with either aggregation or subsampling (specified by the method argument) over an interval using the interval argument. Grouping variable names (such as animal identity) may be passed as a character vector to the `id_columns` argument (Listing 5).

7 | SYSTEM-SPECIFIC PRE-PROCESSING TOOLS

When researchers' pre-processing requirements exceed the functionalities of existing tools, they might have to conceptualise and implement their own methods. For instance, an important and common analysis with animal tracking data is to link space use with environmental covariates. This is difficult even with smoothed and thinned high-throughput data, as these may be too large for statistical packages, or have strong autocorrelation. Users aiming for such analyses can benefit from segmenting and clustering the data into spatiotemporally independent bouts of different behavioural modes (Patin et al., 2020). Treating these as the unit of observation also conveniently sidesteps pseudo-replication and reduces computational requirements. While numerous methods of segmenting and clustering data are in use, they may not be scalable to very large or gappy datasets (Langrock et al., 2012; Michelot et al., 2016; Patin et al., 2020). As an alternative, a first-principles approach that segments data based on the movement capacity (top speed) of tracked animals could provide a fast, yet useful way to cluster data. Here, as a working example that may be suitable for some systems, we present a simple segmentation-clustering algorithm to make 'residence patches', identified as bouts of relatively stationary behaviour (Barraquand & Benhamou, 2008; Bijleveld et al., 2016; Oudman et al., 2018). Details of the implementation may be found in the package code, and examples are provided in the Supporting Information.

7.1 | Conceptualising a simple segmentation-clustering algorithm: The residence-patch example

Before implementing the algorithm, users should identify positions where the animal is relatively stationary, for instance on its speed or first-passage time (Barraquand & Benhamou, 2008; Bracis et al., 2018). Our suggested algorithm begins by assessing whether consecutive stationary positions are spatiotemporally independent, and clusters them together into a residence patch if they are not. This clustering could be based on a simple proximity threshold—points farther apart than some threshold distance are likely to represent two different residence patches. In cases where animals visit multiple sites in sequence (such as traplining; Thomson et al., 1997), and which researchers might wish to consider as a single residence patch, a larger-scale distance threshold can help cluster nearby residence patches together, and this can also be applied to cluster together patches artificially separated due to missing data. Our algorithm separates two observations at a similar location, but at two very different time points, by comparing the intervening time-lag against a time-difference threshold, which can also apply to patches that would otherwise be clustered by the large-scale distance threshold. Users are encouraged to base these thresholds on the movement habits of their study species (see the Worked Out Example).

We have implemented a working example of the simple clustering concept presented here as the function `atl_res_patch` (see Figure 6b; Listing 6), which requires three parameters: (a) the distance threshold between positions (`called buffer_size`), (b) the large-scale distance threshold between clusters of positions (called `lim_spat_indep`) and (c) the time-difference threshold between clusters (called `lim_time_indep`). Clusters formed of fewer than a minimum number of positions can be excluded. Our algorithm performs well when movement modes are clearly separated, and is capable of correctly separating positions that are close together in space and time, but which comprise different behavioural sequences (see Figure 6). While the algorithm may not cover all possible use-cases and study species, we provide it here as an example of a user-built exploratory method for animal tracking data. It is important to systematically test such custom-made algorithms, to ensure reproducibility and reliability (Marwick et al., 2018; Wickham, 2015). Simple examples of such tests for the residence patch method and other functions in `atlastools` may be found in the `tests/` directory in the associated Github repository.

7.2 | A real-world test of user-built pre-processing tools

We applied the pre-processing pipeline using `atlastools` functions described above to an ATLAS dataset to verify that the residence patch method could correctly identify known stopping points (see Figure 7). We collected the data (n = 50,816) on foot and by boat, with a hand-held WATLAS tag (sampling interval = 1s) around the island of Griend (53.25°N, 5.25°E) in August 2020 (WATLAS: Wadden Sea ATLAS system Beardsworth, Gobbens, et al., 2021; Bijleveld et al., 2021). Since the data were intended to test the accuracy of the WATLAS system, we were able to log stops in the track as waypoints using a hand-held GPS device, and manually annotate the WATLAS data with the timestamp of each waypoint (Garmin Dakota

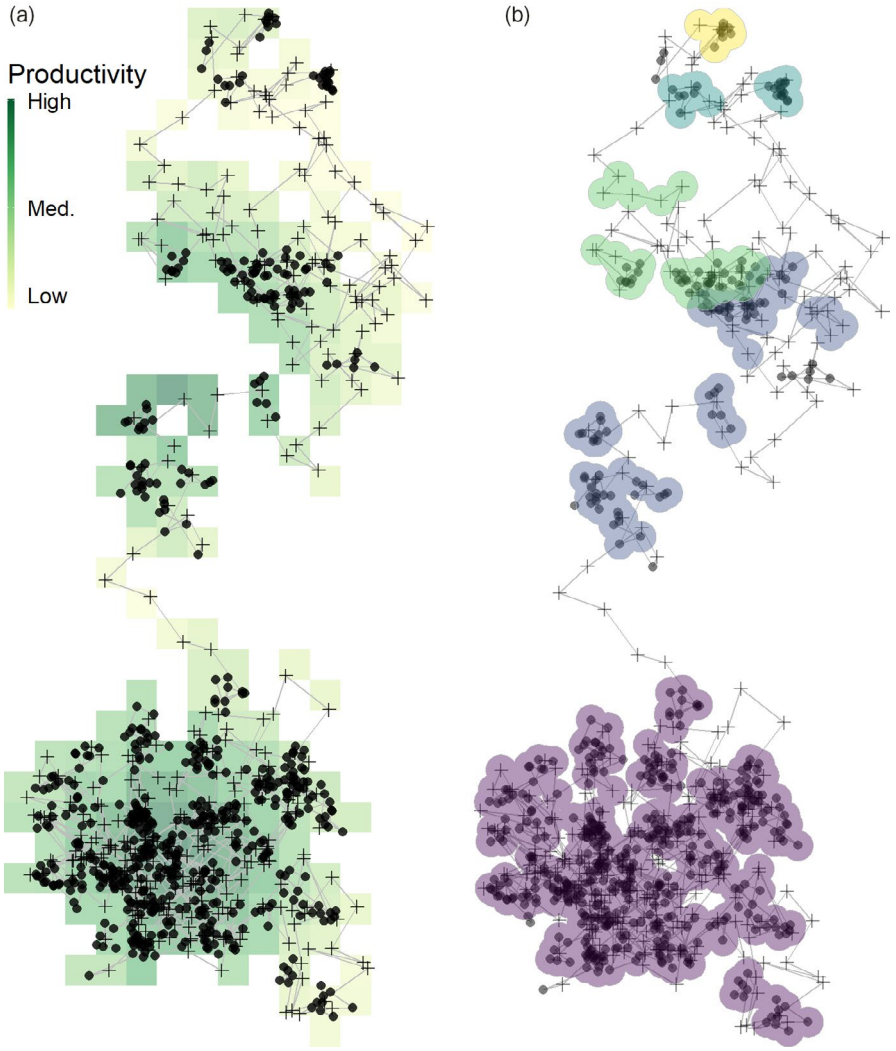

FIGURE 6 Movement tracks can be classified into residence patches while leaving out the transit between them. (a) A simulated animal movement track from Gupte et al. (2021), where an agent uses local cues to make movement decisions to maximise intake. The agent tends to stop (solid circles) on high-productivity areas of the landscape, as these are more likely to generate prey items. Transit points between stationary phases are shown as crosses. (b) Our simple, first-principles-based clustering algorithm classifies the track into five residence patches. Some transit points are erroneously classified as being part of a residence patch (top, yellow), illustrating why is it important to remove such data before applying this method. Simultaneously, some points where the animal is not stationary for long are not picked up by the method. While the large purple patch (bottom) is composed almost entirely of consecutive positions, the subsequent patches are composed of multiple parts. This is because our method was designed to be robust to missing data from empirical tracks; the spatial and temporal limits of splitting and lumping can be controlled using the arguments passed to `atl_res_patch`, and can be adjusted to fit the study system. Users are cautioned that there are no 'correct' options, and the best guide is the behavioural biology of the tracked individual

10; see Beardsworth, Gobbens, et al., 2021). We estimated the real duration of each stop as the time difference between the first and last position recorded within 50 m of each waypoint, within a 10-min window before and after the waypoint timestamp (to avoid biased durations from revisits). Stops had a median duration of 10.28 min (range: 1.75–20 min; see Supporting Information). We cleaned the data before constructing residence patches by (a) removing a single outlier (>15 km away), (b) removing unrealistic movement (≥15 m/s), (c) smoothing the data ($K = 5$) and (d) thinning the data by subsampling over a 30-s interval. The cleaning steps retained 37,324 positions (74.45%), while thinning reduced these to 1,803 positions (4.8% positions of the smoothed track). Details and code are provided in the Supporting Information (see VALIDATING THE RESIDENCE PATCH METHOD WITH CALIBRATION DATA).

We identified stationary positions as those where the median smoothed speed ($K = 5$) was <2 m/s, as people or a boat moving any faster are likely to be in transit. We clustered these positions into residence patches with a buffer radius of 5 m, spatial independence limit of 50 m, temporal independence limit of 5 min and a minimum of 3 positions per patch. Inferred residence patches corresponded well to the locations of stops (see Figure 7c). However, the residence patch algorithm detected seven more stops ($n = 28$) than there were waypoints

```
patches <- atl_res_patch(
                data = track_data,
                buffer_radius = 10,
                lim_spat_indep = 100,
                lim_time_indep = 30,
                min_fixes = 3,
                summary_variables = c("speed"),
                summary_functions = c("mean", "sd")
            )
```

LISTING 6 The `atl_res_patch` function can be used to classify a track into residence patches. The arguments `buffer_radius` and `lim_spat_indep` are specified in metres, while the `lim_time_indep` is provided in minutes. In this example, specifying `summary_variables = c("speed")`, and `summary_functions = c("mean", "sd")` will provide the mean and standard deviation of instantaneous speed in each residence patch. The `atl_patch_summary` function is used to access the classified patch in one of three ways, here using the summary option which returns a table of patch-wise summary statistics

(*n* waypoints = 21). One of these was the field station on Griend where the tag was stored between trips (red triangle, Figure 7c), while another patch was formed of positions recorded while waiting for the boat; such unintended stops, not recorded as waypoints, likely accounted for the remaining five 'extra' residence patches. Our analysis also did not detect two stops of 105 and 563 s (1.75 and 9.4 min) since they were data poor and were cleaned away during pre-processing (*n* positions = 6, 15), highlighting that the quality of the raw data (as in the rest of the track) is still a limiting factor on the inferences that are possible after pre-processing. To determine whether the residence patch method correctly identified the duration of detected stops in the calibration track, we first extracted the patch attributes using the function `atl_patch_summary`. We then matched the patches to the waypoints by their median coordinates (rounded to 100 m). We assigned the inferred duration of the stop as the duration of the spatially matched residence patch. We compared the inferred duration with the real duration using a linear model with the inferred duration as the only predictor of the real

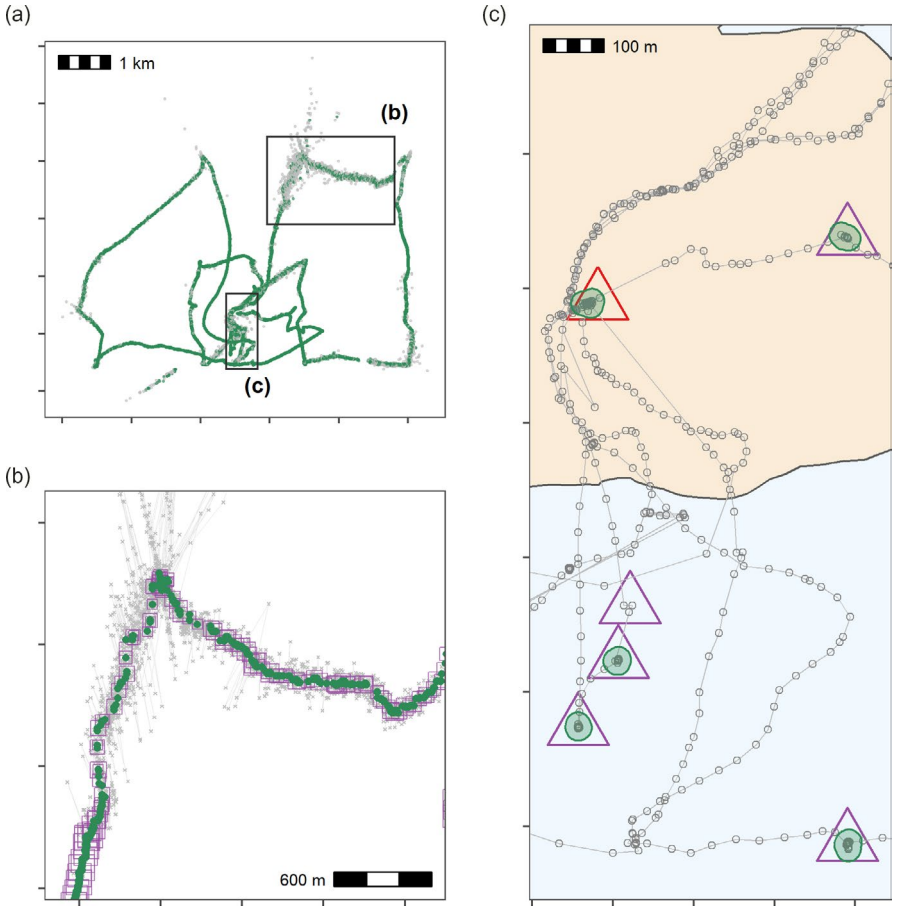

FIGURE 7 Pre-processing steps for WATLAS calibration data showing filtering on speed, median smoothing and thinning by aggregation, and making residence patches. (a) Positions with incoming and outgoing speed >15 m/s are removed (grey crosses = removed, green points = retained). (b) Raw data (grey crosses), median smoothed positions (green circles; moving window $K = 5$) and the smoothed track thinned by aggregation to a 30-s interval (purple squares). Square size corresponds to the number of positions used to calculate the averaged position during thinning. (c) Clustering thinned data into residence patches (green polygons) yields robust estimates of the location of known stops (purple triangles). The algorithm identified all areas with prolonged residence, including those which we had not intended to be recorded, such as stops at the field station ($n = 12$; red triangle). Our analysis could not find two stops of 105- and 563-s duration (6 and 15 fixes, respectively), since these were lost in the data thinning step; one of these is shown here (purple triangle without green polygon)

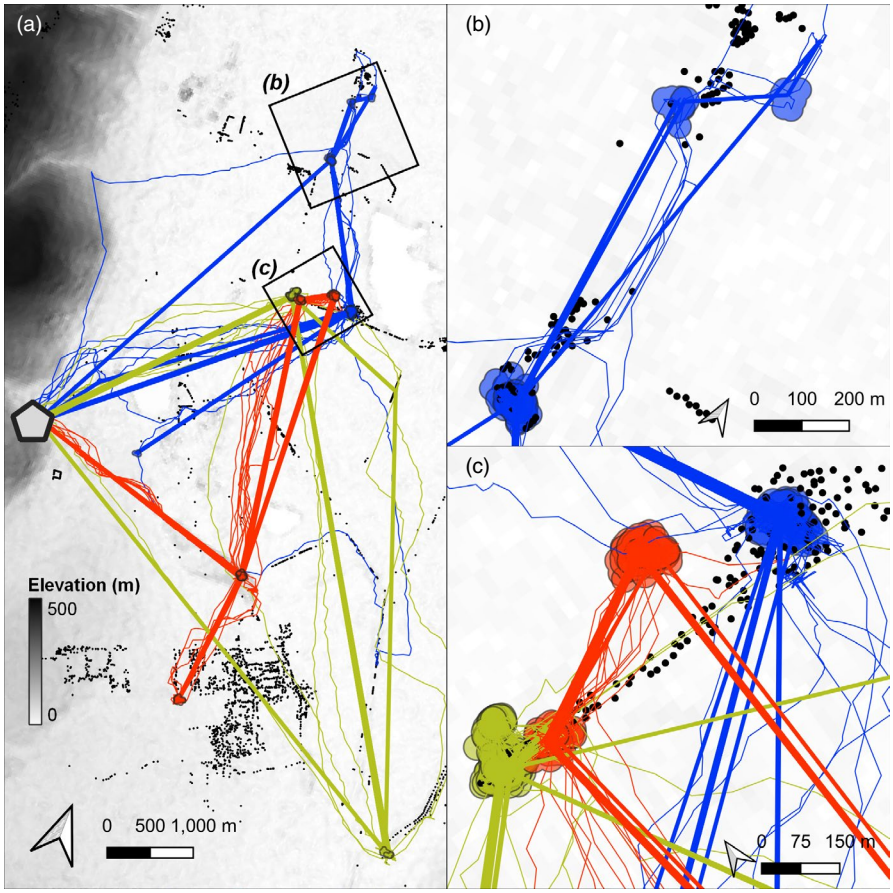

FIGURE 8 Synthesising animal tracks into residence patches can reveal movement in relation to landscape features, prior exploration and other individuals. (a) Linear approximations of the paths (coloured straight lines) between residence patches (circles) of three Egyptian fruit bats *Rousettus aegyptiacus*, tracked over three nights in the Hula Valley, Israel. Real bat tracks are shown as thin lines below the linear approximations, and colours show bat identity. The grey hexagon represents the roost-cave at Gar Hershom. Black points represent known fruit trees. Background is shaded by elevation at 30-m resolution. (b) Spatial representations of an individual bat's residence patches (green polygons) can be used to study site-fidelity by examining overlaps between patches, or to study resource selection by inspecting overlaps with known resources such as fruit trees (black circles). In addition, the linear approximation of movement between patches (straight green lines) can be contrasted with the estimated real path between patches (irregular green lines), for instance, to determine the efficiency of movement between residence patches. (c) Fine-scale tracks (thin coloured lines), large-scale movement (thick lines), residence patch polygons and fruit tree locations show how high-throughput data can be used to study movement across scales. Patches and lines are coloured by bat identity

duration. Inferred duration was a good predictor of the real duration of a stop (linear model estimate = 1.021, *t*-value = 12.965, $p < 0.0001$, $R^2 = 0.908$; see Supporting Information Figure S1.7). This translates to a 2% underestimation of the stop duration at a tracking interval of 30 s. Finally, any classification algorithm will present users with a trade-off between over-sensitivity (erroneously finding stops where there were none), and under-sensitivity (missing stops where they are not local or long enough)—users should balance between these based on the broader questions sought to be answered.

8 | A WORKED OUT EXAMPLE ON ANIMAL TRACKING DATA

We present a fully worked out example of our pre-processing pipeline and residence patch method using movement data from three Egyptian fruit bats *R. aegyptiacus* tracked using the ATLAS system in the Hula Valley, Israel (33.1°N, 35.6°E; Lourie et al., 2021; Toledo et al., 2020). Code and data can be found in the Supporting Information and Zenodo repository (see PROCESSING EGYPTIAN FRUIT BAT TRACKS). Data selected for this example were collected over three nights (5th–7th May 2018), with an average of 13,370 positions (SD = 2,173; range = 11,195–15,542; interval = 8 s) per individual. Plotting the tracks revealed potential location errors (see Figure 1; see also Supporting Information Figure S2.1), which we filtered out by removing observations with ATLAS SD >20 (see Supporting Information Section 2.5), as well as removing observations calculated using fewer than four base stations, altogether trimming 22% of the raw data (mean positions remaining = 10,447 per individual). Then, we removed unrealistic movement represented by positions with incoming and outgoing speeds >20 m/s that exceed the maximum flight speed recorded in this species (15 m/s; Tsoar et al., 2011),

leaving 10,337 positions per individual on average (98% of previous step). We median smoothed the data with a moving window *K* size = 5, and no observations were lost.

We aimed to study bats' nighttime foraging on fruit trees by quantifying the duration of bats' residence patches. We began the construction of residence patches by finding the residence time within 50 m of each position; this is the maximal radius of a 'cloud of points' around fruit trees (Bracis et al., 2018). Foraging bats repeatedly traverse the same routes (Lourie et al., 2021; Toledo et al., 2020; Tsoar et al., 2011) and this could artificially inflate the residence time of positions along these routes. To avoid confusing revisits with residence, we limited the summation of residence times at each position to the period until the first departure of 5 min or more. Thus, two nearby locations (≤50 m apart) each visited for 1 min at a time, but separated by an interval of some hours would not be clustered together as a residence patch. To focus on bats' nighttime foraging behaviour, we also excluded positions during the day (5 a.m.–8 p.m.), and at or near the roost-cave (see Figure 8a) to focus on nighttime foraging behaviour; 22,910 of 31,012 positions remained (73.9%). To determine the true duration of foraging, we opted for a first-principles approach and first selected only locations with a residence time >5 min, reasoning that a flying animal stopping for >5 min at a location should plausibly indicate resource use or another interesting localised behaviour. This step retained 5,736 positions per bat on average (17,208 total), or 72.4% of the nighttime positions. We then constructed residence patches with a buffer distance of 25 m, a spatial independence limit of 100 m, a temporal independence limit of 30 min and rejected patches with fewer than three positions. These values are meant as examples; users should determine the sensitivity of their results to parameter choices. Bats spent 56.95 min at foraging sites (SD = 62.20), and were stationary in particular fruit trees and roosting trees during 83.8% of their foraging time (Figure 8). Although all three bats roosted at the same cave during the day, and all their tracks are within the typical foraging area of bats roosting in this cave (Lourie et al., 2021), they used distinct foraging sites across the area at night (Figure 8a). The lack of overlap among individuals in tree use, obtained with the residence patch algorithm, shows that although co-roosting bats share the same cave-specific foraging area (Lourie et al., 2021), they often forage on different trees. Contrasting the actual movement path with the linear path between residence patches can help reveal details of how animal cognition affects space use (Toledo et al., 2020). Bats tended to show prolonged residence near known food sources (fruit trees), but also where no fruit trees were recorded (Figure 8b,c), in line with previous evidence for their use of non-fruiting trees to rest, to handle and digest food, and presumably for social interactions (Tsoar et al., 2011).

9 | DISCUSSION AND PERSPECTIVE

Recent technical advances in wildlife tracking have already yielded exciting new insights from massive high-resolution movement datasets (Aspillaga, Arlinghaus, Martorell-Barceló, Barcelo-Serra, et al., 2021; Aspillaga, Arlinghaus, Martorell-Barceló, Follana-Berná, et al., 2021; Baktoft et al., 2017, 2019; Beardsworth, Whiteside, Capstick, et al., 2021; Beardsworth, Whiteside, Laker, et al., 2021; Corl et al., 2020; Harel et al., 2016; Harel & Nathan, 2018; Lourie et al., 2021; Oudman et al., 2018; Papageorgiou et al., 2019; Strandburg-Peshkin et al., 2015; Toledo et al., 2020; Tsoar et al., 2011; Vilk et al., 2021), and high-throughput animal tracking is expected to become increasingly more common in the near future. Tackling the very large datasets that high-throughput tracking generates requires a different approach from that used for traditionally smaller volumes of data. We foresee that movement ecologists will have to adopt ever more practices from fields accustomed to dealing with 'big data', and that the field will become increasingly computational (Peng, 2011). Researchers have long used some of these approaches ad hoc, such as exploratory data analysis on small subsets before applying methods to the full data, using efficient tools, and basic batch processing. Yet formally prescribing these steps can help practitioners avoid pitfalls and implement techniques that make their analyses quicker and more reliable. Standardised principles, implemented a basic pipeline, for approaching data cleaning promote reproducibility across studies, making comparative inferences more robust. While massive datasets make reliance on standardised pipelines necessary, the output of such pipeline should periodically manually double-checked to ensure 'realistic' output. The open-source `R` package `atlastools` serves as a starting point for methodological collaboration among movement ecologists, and as a simple working example on which researchers may wish to model their own tools. Efficient location error modelling approaches (Aspillaga, Arlinghaus, Martorell-Barceló, Follana-Berná, et al., 2021; Fleming et al., 2020) may eventually make data-cleaning optional. Yet, cleaning tracking data even partially before modelling location error is faster than error-modelling on the full data, and the removal of large location errors may improve model fits. Thus, we see our pipeline as complementary to these approaches (Fleming et al., 2014, 2020). Finally, we recognise that the diversity and complexity of animal movement and data collection techniques often requires system-specific, even bespoke, pre-processing solutions. Though the principles outlined here are readily generalised to numerous data sources (including terrestrial radio-based reverse GPS: e.g. Toledo et al., 2020, marine acoustic reverse GPS: e.g. Aspillaga, Arlinghaus, Martorell-Barceló, Follana-Berná, et al., 2021, high-resolution GPS: e.g. Strandburg-Peshkin et al., 2015 and video tracking: Rathore et al., 2020), users' requirements will eventually exceed the particular tools we provide. We see the diversity of animal tracking datasets and studies as an incentive for more users to be involved in developing methods for their systems. We offer our approach to large tracking datasets, and our pipeline and package as a foundation for system-specific tools in the belief that simple, robust concepts are key to methods development that balances system specificity and broad applicability.

ACKNOWLEDGEMENTS

P.R.G. would like to thank Pedro M. Santos Neves for introducing P.R.G. to R package development, for help with setting up atlastools, and for help with archiving it on Zenodo; Geert Aarts, Evy Gobbens and Roos Kentie for feedback that improved the manuscript; members of the Modelling Adaptive Response Mechanisms Group (Weissing Lab), and the Theoretical Biology department at the University of Groningen for helpful discussions on atlastools and the manuscript. We thank the many volunteers, students and NIOZ staff involved in operating the WATLAS tracking system, and most importantly Frank van Maarseveen, Bas Denissen and Anne Dekinga. We also thank Yotam Orchan, Yoav Bartan, Sivan Margalit, Anat Levi, David Shohami, Ohad Vilk and other members of the Minerva Center for Movement Ecology for their valuable support, and especially the attendees of ATLAS workshops held in May and June 2020 at the Hebrew University of Jerusalem for helpful comments on the pipeline and atlastools. Improvements to atlastools based on users' feedback are acknowledged on Github. Finally, we thank Ulrike Schlägel and three anonymous reviewers whose comments improved this manuscript. This work was partly funded by the Dutch Research Council grant VI.Veni.192.051 awarded to A.I.B. ATLAS development was funded by the Minerva Foundation grant and the Adelina and Massimo Della Pergola Professor of Life Sciences to R.N., and by the Israel Science Foundation grant (ISF ISF-965/15) to R.N. and S.T. P.R.G. was supported by an Adaptive Life Programme grant in the Weissing Lab, made possible by the University of Groningen's Faculty of Science and Engineering, and the Groningen Institute for Evolutionary Life Sciences (GELIFES).

COMPETING INTERESTS

The authors declare that they have no competing interests.

AUTHORS' CONTRIBUTIONS

P.R.G. wrote the manuscript and inline code snippets, performed the analyses, prepared the figures and developed the R package atlastools; C.E.B. and A.I.B. collected the calibration track, and E.L. collected the bat movement data, roost and fruit tree locations; R.N. conceived the idea of writing this manuscript, and P.R.G., A.I.B., O.S., C.E.B., S.T. and E.L. contributed to its design, and the design of atlastools. All authors contributed to the writing of the manuscript and the design of figures.

DATA AVAILABILITY STATEMENT

The data and source code to reproduce the figures and analyses in this article and in the Supporting Information can be found in the Zenodo repository at https://doi.org/10.5281/zenodo.4287462 (Gupte, 2021).

ORCID

Pratik Rajan Gupte https://orcid.org/0000-0001-5294-7819
Christine E. Beardsworth http://orcid.org/0000-0003-1308-1455
Orr Spiegel http://orcid.org/0000-0001-8941-3175
Emmanuel Lourie https://orcid.org/0000-0001-7364-0082
Sivan Toledo http://orcid.org/0000-0002-9524-7115
Ran Nathan http://orcid.org/0000-0002-5733-6715
Allert I. Bijleveld https://orcid.org/0000-0002-3159-8944

REFERENCES

Aarts, G., MacKenzie, M., McConnell, B., Fedak, M., & Matthiopoulos, J. (2008). Estimating space-use and habitat preference from wildlife telemetry data. *Ecography*, *31*, 140–160. https://doi.org/10.1111/j.2007.0906-7590.05236.x

Alston, J. M., & Rick, J. A. (2020). A beginner's guide to conducting reproducible research. *The Bulletin of the Ecological Society of Americ*, e01801. https://doi.org/10.1002/bes2.1801

Archmiller, A. A., Johnson, A. D., Nolan, J., Edwards, M., Elliott, L. H., Ferguson, J. M., Iannarilli, F., Vélez, J., Vitense, K., Johnson, D. H., & Fieberg, J. (2020). Computational reproducibility in The Wildlife Society's Flagship Journals. *The Journal of Wildlife Management*, *84*, 1012–1017. https://doi.org/10.1002/jwmg.21855

Aspillaga, E., Arlinghaus, R., Martorell-Barceló, M., Barcelo-Serra, M., & Alós, J. (2021). High-throughput tracking of social networks in marine fish populations. *Frontiers in Marine Science*, *8*. https://doi.org/10.3389/fmars.2021.688010

Aspillaga, E., Arlinghaus, R., Martorell-Barceló, M., Follana-Berná, G., Lana, A., Campos-Candela, A., & Alós, J. (2021). Performance of a novel system for high-resolution tracking of marine fish societies. *Animal Biotelemetry*, *9*, 1. https://doi.org/10.1186/s40317-020-00224-w

Avgar, T., Potts, J. R., Lewis, M. A., & Boyce, M. S. (2016). Integrated step selection analysis: Bridging the gap between resource selection and animal movement. *Methods in Ecology and Evolution*, *7*, 619–630. https://doi.org/10.1111/2041-210X.12528

Baktoft, H., Gjelland, K. Ø., Økland, F., Rehage, J. S., Rodemann, J. R., Corujo, R. S., Viadero, N., & Thygesen, U. H. (2019). Opening the black box of high resolution fish tracking using yaps. bioRxiv, 2019.12.16.877688.

Baktoft, H., Gjelland, K. Ø., Økland, F., & Thygesen, U. H. (2017). Positioning of aquatic animals based on time-of-arrival and random walk models using YAPS (Yet Another Positioning Solver). *Scientific Reports*, *7*, 14294. https://doi.org/10.1038/s41598-017-14278-z

Barnett, A. H., & Moorcroft, P. R. (2008). Analytic steady-state space use patterns and rapid computations in mechanistic home range analysis. *Journal of Mathematical Biology*, *57*, 139–159. https://doi.org/10.1007/s00285-007-0149-8

Barraquand, F., & Benhamou, S. (2008). Animal movements in heterogeneous landscapes: Identifying profitable places and homogeneous movement bouts. *Ecology*, *89*, 3336–3348. https://doi.org/10.1890/08-0162.1

Beardsworth, C. E., Gobbens, E., van Maarseveen, F., Denissen, B., Dekinga, A., Nathan, R., Toledo, S., & Bijleveld, A. I. (2021). Validating a high-throughput tracking system: ATLAS as a regional-scale alternative to GPS. bioRxiv, 2021.02.09.430514.

Beardsworth, C. E., Whiteside, M. A., Capstick, L. A., Laker, P. R., Langley, E. J. G., Nathan, R., Orchan, Y., Toledo, S., van Horik, J. O., & Madden, J. R. (2021). Spatial cognitive ability is associated with transitory movement speed but not straightness during the early stages of exploration. *Royal Society Open Science*, *8*. https://doi.org/10.1098/rsos.201758

Beardsworth, C. E., Whiteside, M. A., Laker, P. R., Nathan, R., Orchan, Y., Toledo, S., van Horik, J. O., & Madden, J. R. (2021). Is habitat selection in the wild shaped by individual-level cognitive biases in orientation strategy? *Ecology Letters*, *24*, 751–760. https://doi.org/10.1111/ele.13694

Bijleveld, A. I., MacCurdy, R. B., Chan, Y.-C., Penning, E., Gabrielson, R. M., Cluderay, J., Spaulding, E. L., Dekinga, A., Holthuijsen, S., ten Horn, J., Brugge, M., van Gils, J. A., Winkler, D. W., & Piersma, T.

(2016). Understanding spatial distributions: Negative density-dependence in prey causes predators to trade-off prey quantity with quality. *Proceedings of the Royal Society B: Biological Sciences*, *283*(1828), 20151557–https://doi.org/10.1098/rspb.2015.1557

Bijleveld, A. I., van Maarseveen, F., Denissen, B., Dekinga, A., Penning, E., Ersoy, S., Gupte, P., de Monte, L., ten Horn, J., Bom, R., Toledo, S., Nathan, R., & Beardsworth, C. (2021). WATLAS: High resolution and real-time tracking of many small birds in the Dutch Wadden Sea. *bioRxiv*, *11*(8), 467683. https://doi.org/10.1101/2021.11.08.467683

Bjørneraas, K., Moorter, B. V., Rolandsen, C. M., & Herfindal, I. (2010). Screening global positioning system location data for errors using animal movement characteristics. *The Journal of Wildlife Management*, *74*, 1361–1366. https://doi.org/10.1111/j.1937-2817.2010.tb01258.x

Boone, M., Joo, R., & Basille, M. (2020). *sftrack: Modern Classes for Tracking and Movement Data*. R package version 0.5.3. Retrieved from https://mablab.org/sftrack/

Bracis, C., Bildstein, K. L., & Mueller, T. (2018). Revisitation analysis uncovers spatio-temporal patterns in animal movement data. *Ecography*, *41*, 1801–1811.

Calabrese, J. M., Fleming, C. H., & Gurarie, E. (2016). Ctmm: An R package for analyzing animal relocation data as a continuous-time stochastic process. *Methods in Ecology and Evolution*, *7*, 1124–1132.

Calenge, C. (2006). The package adehabitat for the R software: Tool for the analysis of space and habitat use by animals. *Ecological Modelling*, *197*, 1035.

Calenge, C., Dray, S., & Royer-Carenzi, M. (2009). The concept of animals' trajectories from a data analysis perspective. *Ecological Informatics*, *4*, 34–41. https://doi.org/10.1016/j.ecoinf.2008.10.002

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, *13*, 377–387. https://doi.org/10.1145/362384.362685

Corl, A., Charter, M., Rozman, G., Toledo, S., Turjeman, S., Kamath, P. L., Getz, W. M., Nathan, R., & Bowie, R. C. K. (2020). Movement ecology and sex are linked to barn owl microbial community composition. *Molecular Ecology*, *29*, 1358–1371. https://doi.org/10.1111/mec.15398

Dai, Z. (2021). *Disk.Frame: Larger-than-Ram Disk-Based Data Manipulation Framework*. Retrived from https://diskframe.com/

Dowle, M., & Srinivasan, A. (2020). *data.table: Extension of `data.frame`*. R package version 1.14.2. Retrieved from https://CRAN.R-project.org/package=data.table

Dupke, C., Bonenfant, C., Reineking, B., Hable, R., Zeppenfeld, T., Ewald, M., & Heurich, M. (2017). Habitat selection by a large herbivore at multiple spatial and temporal scales is primarily governed by food resources. *Ecography*, *40*, 1014–1027. https://doi.org/10.1111/ecog.02152

Eddelbuettel, D. (2013). *Seamless R and C++ Integration with Rcpp. Use R!*. Springer-Verlag.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*, 381–395. https://doi.org/10.1145/358669.358692

Fleming, C. H., Calabrese, J. M., Mueller, T., Olson, K. A., Leimgruber, P., & Fagan, W. F. (2014). From fine-scale foraging to home ranges: A semivariance approach to identifying movement modes across spatiotemporal scales. *The American Naturalist*, *183*, E154–E167. https://doi.org/10.1086/675504

Fleming, C. H., Drescher-Lehman, J., Noonan, M. J., Akre, T. S. B., Brown, D. J., Cochrane, M. M., Dejid, N., DeNicola, V., DePerno, C. S., Dunlop, J. N., Gould, N. P., Hollins, J., Ishii, H., Kaneko, Y., Kays, R., Killen, S. S., Koeck, B., Lambertucci, S. A. & Calabrese, J. M. … (2020). A comprehensive framework for handling location error in animal tracking data*. bioRxiv, 2020.06.12.130195.

Getz, W. M., & Saltz, D. (2008). A framework for generating and analyzing movement paths on ecological landscapes. *Proceedings of the National Academy of Sciences of the United States of America*, *105*, 19066–19071. https://doi.org/10.1073/pnas.0801732105

Gupte, P. R. (2020). Atlastools: Pre-processing tools for high frequency tracking data. *Zenodo*, https://doi.org/10.5281/zenodo.4033154

Gupte, P. R. (2021). Source code, data, and supplementary material in the form of tutorials for "A Guide to Pre-Processing High-Throughput Animal Tracking Data" (v1.09). *Zenodo*, https://doi.org/10.5281/zenodo.5554729

Gupte, P. R., Netz, C. F., & Weissing, F. J. (2021). The joint evolution of movement and competition strategies. bioRxiv, 2021.07.19.452886.

Gurarie, E., Fleming, C. H., Fagan, W. F., Laidre, K. L., Hernández-Pliego, J., & Ovaskainen, O. (2017). Correlated velocity models as a fundamental unit of animal movement: Synthesis and applications. *Movement Ecology*, *5*, 13. https://doi.org/10.1186/s40462-017-0103-3

Haddaway, N. R., & Verhoeven, J. T. (2015). Poor methodological detail precludes experimental repeatability and hampers synthesis in ecology. *Ecology and Evolution*, *5*, 4451–4454. https://doi.org/10.1002/ece3.1722

Harel, R., Horvitz, N., & Nathan, R. (2016). Adult vultures outperform juveniles in challenging thermal soaring conditions. *Scientific Reports*, *6*, 27865. https://doi.org/10.1038/srep27865

Harel, R., & Nathan, R. (2018). The characteristic time-scale of perceived information for decision-making: Departure from thermal columns in soaring birds. *Functional Ecology*, *32*, 2065–2072. https://doi.org/10.1111/1365-2435.13136

Holyoak, M., Casagrandi, R., Nathan, R., Revilla, E., & Spiegel, O. (2008). Trends and missing parts in the study of movement ecology. *Proceedings of the National Academy of Sciences of the United States of America*, *105*, 19060–19065. https://doi.org/10.1073/pnas.0800483105

Horvitz, N., Sapir, N., Liechti, F., Avissar, R., Mahrer, I., & Nathan, R. (2014). The gliding speed of migrating birds: Slow and safe or fast and risky? *Ecology Letters*, *17*, 670–679. https://doi.org/10.1111/ele.12268

Hurford, A. (2009). GPS Measurement error gives rise to spurious 180 turning angles and strong directional biases in animal movement data. *PLoS ONE*, *4*, e5632. https://doi.org/10.1371/journal.pone.0005632

Hussey, N. E., Kessel, S. T., Aarestrup, K., Cooke, S. J., Cowley, P. D., Fisk, A. T., Harcourt, R. G., Holland, K. N., Iverson, S. J., Kocik, J. F., Mills Flemming, J. E., & Whoriskey, F. G. (2015). Aquatic animal telemetry: A panoramic window into the underwater world. *Science*, *348*, 1255642. https://doi.org/10.1126/science.1255642

Johnson, D. S., London, J. M., Lea, M.-A., & Durban, J. W. (2008). Continuous-time correlated random walk model for animal telemetry data. *Ecology*, *89*, 1208–1215. https://doi.org/10.1890/07-1032.1

Jonsen, I. D., Flemming, J. M., & Myers, R. A. (2005). Robust state-space modeling of animal movement data. *Ecology*, *86*, 2874–2880. https://doi.org/10.1890/04-1852

Jonsen, I. D., Myers, R. A., & Flemming, J. M. (2003). Meta-analysis of animal movement using state-space models. *Ecology*, *84*, 3055–3063. https://doi.org/10.1890/02-0670

Joo, R., Boone, M. E., Clay, T. A., Patrick, S. C., Clusella-Trullas, S., & Basille, M. (2020). Navigating through the R packages for movement. *Journal of Animal Ecology*, *89*, 248–267.

Joo, R., Picardi, S., Boone, M. E., Clay, T. A., Patrick, S. C., Romero-Romero, V. S., & Basille, M. (2020). A decade of movement ecology. arXiv:2006.00110 [q-bio].

Jung, K. W., Deng, Z. D., Martinez, J. J., Geist, D. R., McMichael, G. A., Stephenson, J. R., & Graf, P. J. (2015). Performance of an acoustic telemetry system in a large fishway. *Animal Biotelemetry*, *3*, 17. https://doi.org/10.1186/s40317-015-0052-9

Kaplan, E., & Hegarty, C. (2005). *Understanding GPS: Principles and applications*. Artech House.

Kays, R., Crofoot, M. C., Jetz, W., & Wikelski, M. (2015). Terrestrial animal tracking as an eye on life and planet. *Science*, *348*. https://doi.org/10.1126/science.aaa2478

Klarevas-Irby, J. A., Wikelski, M., & Farine, D. R. (2021). Efficient movement strategies mitigate the energetic cost of dispersal. *Ecology Letters*, *24*, 1432–1442. https://doi.org/10.1111/ele.13763

Kranstauber, B., Cameron, A., Weinzerl, R., Fountain, T., Tilak, S., Wikelski, M., & Kays, R. (2011). The Movebank data model for animal tracking. *Environmental Modelling & Software*, *26*, 834–835. https://doi.org/10.1016/j.envsoft.2010.12.005

Langrock, R., King, R., Matthiopoulos, J., Thomas, L., Fortin, D., & Morales, J. M. (2012). Flexible and practical modeling of animal telemetry data: Hidden Markov models and extensions. *Ecology*, *93*, 2336–2342. https://doi.org/10.1890/11-2241.1

Lewis, K. P., Vander Wal, E., & Fifield, D. A. (2018). Wildlife biology, big data, and reproducible research. *Wildlife Society Bulletin*, *42*, 172–179. https://doi.org/10.1002/wsb.847

Lourie, E., Schiffner, I., Toledo, S., & Nathan, R. (2021). Memory and conformity, but not competition, explain spatial partitioning between two neighboring fruit bat colonies. *Frontiers in Ecology and Evolution*, *9*, 732514. https://doi.org/10.3389/fevo.2021.732514

MacCurdy, R. B., Bijleveld, A. I., Gabrielson, R. M., & Cortopassi, K. A. (2019). Automated wildlife radio tracking. In R. Zekavat & R. M. Buehrer (Eds.), *Handbook of position location* (Chap. 33, pp. 1219–1261). John Wiley & Sons Ltd.

MacCurdy, R., Gabrielson, R., Spaulding, E., Purgue, A., Cortopassi, K., & Fristrup, K. (2009). Automatic animal tracking using matched filters and time difference of arrival. *Journal of Communications*, *4*, 487–495. https://doi.org/10.4304/jcm.4.7.487-495

Manly, B., McDonald, L., Thomas, D. L., McDonald, T. L., & Erickson, W. P. (2007). *Resource selection by animals: Statistical design and analysis for field studies*. Springer Science & Business Media.

Marwick, B., Boettiger, C., & Mullen, L. (2018). Packaging data analytical work reproducibly using R (and friends). *The American Statistician*, *72*, 80–88. https://doi.org/10.1080/00031305.2017.1375986

Michelot, T., Langrock, R., & Patterson, T. A. (2016). moveHMM: An R package for the statistical modelling of animal movement data using hidden Markov models. *Methods in Ecology and Evolution*, *7*, 1308–1315.

Nathan, R., Getz, W. M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D., & Smouse, P. E. (2008). A movement ecology paradigm for unifying organismal movement research. *Proceedings of the National Academy of Sciences of the United States of America*, *105*, 19052–19059. https://doi.org/10.1073/pnas.0800375105

Netz, C. F., & Gupte, P. R. (2021). Kleptomove: Source code for an individual-based model of the co-evolution of animal movement and competition strategies. *Zenodo*, https://doi.org/10.5281/zenodo.4905475

Noonan, M. J., Fleming, C. H., Akre, T. S., Drescher-Lehman, J., Gurarie, E., Harrison, A.-L., Kays, R., & Calabrese, J. M. (2019). Scale-insensitive estimation of speed and distance traveled from animal tracking data. *Movement Ecology*, *7*, 35. https://doi.org/10.1186/s40462-019-0177-1

Oudman, T., Piersma, T., Ahmedou Salem, M. V., Feis, M. E., Dekinga, A., Holthuijsen, S., ten Horn, J., van Gils, J. A., & Bijleveld, A. I. (2018). Resource landscapes explain contrasting patterns of aggregation and site fidelity by red knots at two wintering sites. *Movement Ecology*, *6*, 24. https://doi.org/10.1186/s40462-018-0142-4

Papageorgiou, D., Christensen, C., Gall, G. E. C., Klarevas-Irby, J. A., Nyaguthii, B., Couzin, I. D., & Farine, D. R. (2019). The multilevel society of a small-brained bird. *Current Biology*, *29*, R1120–R1121. https://doi.org/10.1016/j.cub.2019.09.072

Patin, R., Etienne, M.-P., Lebarbier, E., Chamaillé-Jammes, S., & Benhamou, S. (2020). Identifying stationary phases in multivariate time series for highlighting behavioural modes and home range settlements. *Journal of Animal Ecology*, *89*, 44–56. https://doi.org/10.1111/1365-2656.13105

Patterson, T. A., Thomas, L., Wilcox, C., Ovaskainen, O., & Matthiopoulos, J. (2008). State–space models of individual animal movement. *Trends in Ecology & Evolution*, *23*, 87–94. https://doi.org/10.1016/j.tree.2007.10.009

Pebesma, E. (2018). Simple features for R: Standardized support for spatial vector data. *The R Journal*, *10*, 439–446. https://doi.org/10.32614/RJ-2018-009

Peng, R. D. (2011). Reproducible research in computational science. *Science*, *334*, 1226–1227. https://doi.org/10.1126/science.1213847

Pérez-Escudero, A., Vicente-Page, J., Hinz, R. C., Arganda, S., & de Polavieja, G. G. (2014). idTracker: Tracking individuals in a group by automatic identification of unmarked animals. *Nature Methods*, *11*, 743–748. https://doi.org/10.1038/nmeth.2994

Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F. D. V., Fufezan, C., Ternent, T., Eglen, S. J., Katz, D. S., Pollard, T. J., Konovalov, A., Flight, R. M., Blin, K., & Vizcaíno, J. A. (2016). Ten simple rules for taking advantage of Git and GitHub. *PLOS Computational Biology*, *12*, e1004947. https://doi.org/10.1371/journal.pcbi.1004947

Powers, S. M., & Hampton, S. E. (2019). Open science, reproducibility, and transparency in ecology. *Ecological Applications*, *29*, e01822.

R Core Team. (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.

Ranacher, P., Brunauer, R., Trutschnig, W., der Spek, S. V., & Reich, S. (2016). Why GPS makes distances bigger than they are. *International Journal of Geographical Information Science*, *30*, 316–333. https://doi.org/10.1080/13658816.2015.1086924

Rathore, A., Sharma, A., Sharma, N., Torney, C. J., & Guttal, V. (2020). Multi-Object Tracking in Heterogeneous environments (MOTHe) for animal video recordings. bioRxiv, 2020.01.10.899989.

Schadt, E. E., Linderman, M. D., Sorenson, J., Lee, L., & Nolan, G. P. (2010). Computational solutions to large-scale data management and analysis. *Nature Reviews Genetics*, *11*, 647–657. https://doi.org/10.1038/nrg2857

Seidel, D. P., Dougherty, E., Carlson, C., & Getz, W. M. (2018). Ecological metrics and methods for GPS movement data. *International Journal of Geographical Information Science*, *32*, 2272–2293. https://doi.org/10.1080/13658816.2018.1498097

Signer, J., Fieberg, J., & Avgar, T. (2017). Estimating utilization distributions from fitted step-selection functions. *Ecosphere*, *8*, e01771. https://doi.org/10.1002/ecs2.1771

Slingsby, A., & van Loon, E. (2016). Exploratory visual analysis for animal movement ecology. *Computer Graphics Forum*, *35*, 471–480. https://doi.org/10.1111/cgf.12923

Stine, P. A., & Hunsaker, C. T. (2001). An introduction to uncertainty issues for spatial data used in ecological applications. In C. T. Hunsaker, M. F. Goodchild, M. A. Friedl, & T. J. Case (Eds.), *Spatial uncertainty in ecology: Implications for remote sensing and GIS applications* (pp. 91–107). Springer.

Strandburg-Peshkin, A., Farine, D. R., Couzin, I. D., & Crofoot, M. C. (2015). Shared decision-making drives collective movement in wild baboons. *Science*, *348*, 1358–1361. https://doi.org/10.1126/science.aaa5099

Thomson, J. D., Slatkin, M., & Thomson, B. A. (1997). Trapline foraging by bumble bees: II. Definition and detection from sequence data. *Behavioral Ecology*, *8*, 199–210.

Toledo, S., Kishon, O., Orchan, Y., Bartan, Y., Sapir, N., Vortman, Y., & Nathan, R. (2014). Lightweight low-cost wildlife tracking tags using integrated transceivers. In *2014 6th European Embedded Design in Education and Research Conference (EDERC)*, pp. 287–291.

Toledo, S., Kishon, O., Orchan, Y., Shohat, A., & Nathan, R. (2016). Lessons and experiences from the design, implementation, and

deployment of a wildlife tracking system. In *2016 IEEE International Conference on Software Science, Technology and Engineering (SWSTE)*, pp. 51–60.

Toledo, S., Shohami, D., Schiffner, I., Lourie, E., Orchan, Y., Bartan, Y., & Nathan, R. (2020). Cognitive map–based navigation in wild bats revealed by a new high-throughput tracking system. *Science*, *369*, 188–193. https://doi.org/10.1126/science.aax6904

Tsoar, A., Nathan, R., Bartan, Y., Vyssotski, A., Dell'Omo, G., & Ulanovsky, N. (2011). Large-scale navigational map in a mammal. *Proceedings of the National Academy of Sciences of the United States of America*, *108*, E718–E724. https://doi.org/10.1073/pnas.1107365108

Tukey, J. W. (1977). *Exploratory Data Analysis* (Vol. 2). Addison-Wesley Pub. Co.

Vilk, O., Orchan, Y., Charter, M., Ganot, N., Toledo, S., Nathan, R., & Assaf, M. (2021). Ergodicity breaking and lack of a typical waiting time in area-restricted search of avian predators. arXiv:2101.11527 [cond-mat, physics:physics, q-bio].

Visscher, D. R. (2006). GPS measurement error and resource selection functions in a fragmented landscape. *Ecography*, *29*, 458–464. https://doi.org/10.1111/j.0906-7590.2006.04648.x

Weiser, A. W., Orchan, Y., Nathan, R., Charter, M., Weiss, A. J., & Toledo, S. (2016). Characterizing the accuracy of a self-synchronized reverse-GPS wildlife localization system. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 1–12.

Wickham, H. (2015). *R packages: Organize, test, document, and share your code*. O'Reilly Media Inc.

Wikelski, M., Kays, R. W., Kasdin, N. J., Thorup, K., Smith, J. A., & Swenson Jr., G. W. (2007). Going wild: What a global small-animal tracking system could do for experimental biologists. *Journal of Experimental Biology*, *210*, 181–186. https://doi.org/10.1242/jeb.02629

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Gupte, P. R., Beardsworth, C. E., Spiegel, O., Lourie, E., Toledo, S., Nathan, R., & Bijleveld, A. I. (2021). A guide to pre-processing high-throughput animal tracking data. *Journal of Animal Ecology*, 00, 1–21. https://doi.org/10.1111/1365-2656.13610