Animal Biotelemetry

METHODOLOGY Open Access

Tools for integrating inertial sensor data with video bio-loggers, including estimation of animal orientation, motion, and position

David E. Cade[1,2*], William T. Gough[2], Max F. Czapanskiy[2], James A. Fahlbusch[2,3], Shirel R. Kahane-Rapport[2], Jacob M. J. Linsky[4], Ross C. Nichols[1], William K. Oestreich[2], Danuta M. Wisniewska[2,5], Ari S. Friedlaender[1] and Jeremy A. Goldbogen[2]

Abstract

Bio-logging devices equipped with inertial measurement units—particularly accelerometers, magnetometers, and pressure sensors—have revolutionized our ability to study animals as necessary electronics have gotten smaller and more affordable over the last two decades. These animal-attached tags allow for fine scale determination of behavior in the absence of direct observation, particularly useful in the marine realm, where direct observation is often impossible, and recent devices can integrate more power hungry and sensitive instruments, such as hydrophones, cameras, and physiological sensors. To convert the raw voltages recorded by bio-logging sensors into biologically meaningful metrics of orientation (e.g., pitch, roll and heading), motion (e.g., speed, specific acceleration) and position (e.g., depth and spatial coordinates), we developed a series of MATLAB tools and online instructional tutorials. Our tools are adaptable for a variety of devices, though we focus specifically on the integration of video, audio, 3-axis accelerometers, 3-axis magnetometers, 3-axis gyroscopes, pressure, temperature, light and GPS data that are the standard outputs from Customized Animal Tracking Solutions (CATS) video tags. Our tools were developed and tested on cetacean data but are designed to be modular and adaptable for a variety of marine and terrestrial species. In this text, we describe how to use these tools, the theories and ideas behind their development, and ideas and additional tools for applying the outputs of the process to biological research. We additionally explore and address common errors that can occur during processing and discuss future applications. All code is provided open source and is designed to be useful to both novice and experienced programmers.

Keywords: Bio-logging, MATLAB, Video tags, Video processing, Kinematic sensors, Sensor integration, Tag tutorial, CATS videos

Background

Animal-borne bio-logging devices, also referred to herein as "tags," have undergone a revolution in utility over the last two decades. The consumer cellphone and microelectronics industries have driven the development of critical power sources, memory chips, and sensors small enough to run at high sample rates for long periods of time on minimal battery power at price points that make them accessible for research. Variations of these devices have been used to produce insights into the behavioral ecology of both terrestrial and marine species ranging from whales [1, 2] to aardvarks [3] to mussels [4].

However, despite the ubiquity of these sensors in devices ranging from consumer cellphones to military avionics, open access software to process and interpret

*Correspondence: davecade@stanford.edu
[2] Hopkins Marine Station, Stanford University, 120 Ocean View Blvd, Pacific Grove, CA 93950, USA
Full list of author information is available at the end of the article

the resulting complex data sets have lagged behind the development of hardware, generally because biological research does not have the financial incentives of a modern tech company to hire developers. Additionally, using these tags effectively requires a degree of accuracy that is not required of consumer cell phones, implying that widely available "apps" generally do not provide the requisite accuracy for comparative biological insights. For example, to determine the location of an aquatic animal in 3D space, the interval of time between consecutive known locations (often from GPS positions acquired when the animal is at the surface), the orientation and motion of the animal must be integrated. The microelectromechanical systems (MEMS) sensors that most tags rely on, however, do not have the accuracy of expensive military-grade sensors suitable for use when the time interval between known locations is large. In contrast, consumer cellphones have near-constant access to GPS data from which to determine location, so commercial algorithms with MEMS devices do not prioritize the determination of precise positions from widely spaced GPS positions.

Direct attachment of high-resolution bio-logging devices to animals began with time-depth recorders built from the bones of kitchen timer mechanisms attached to Weddell seals in the 1960s [5, 6]. The time-depth recorder remains an essential component of bio-logging tags despite the integration of many other types of sensors that broaden their application and utility. These include the use of hydrophones for quantifying the acoustic behavior of tagged animals as well as environmental soundscapes [7–9], biomedical sampling devices for measuring gas management during diving [10–13], and multi-axial motion sensors and video to understand fine-scale kinematics of swimming and feeding [14–16]. The expanding diversity of bio-logging uses has resulted in the generation of corresponding analytical code for viewing and processing bio-logging data, sometimes published alongside original research papers but also sometimes disseminated publicly (e.g., in MATLAB, Octave and R at http://www.animaltags.org/ and for Igor at https://sites.google.com/site/ethographer/). However, a comprehensive "volts to useful metrics" guide may decrease barriers for entry into the field for early career researchers and novice users. This approach alongside a standardization framework for bio-logging data [17] should enhance collaborative efforts among bio-logging research groups and communities.

Here, we detail the use of tools developed in MATLAB (MathWorks, Inc., v2014a–2021a) for converting raw bio-logging data into biologically meaningful metrics of orientation (e.g., pitch, roll and heading), motion (e.g., speed, specific acceleration) and position (e.g., depth and spatial coordinates). Although aspects of this process have been described elsewhere (e.g., [18, 14, 19, 20, 7, 21, 22], this manuscript provides details for a start-to-finish process and includes all code, such that a new user should be able to follow steps from opening a tag out of the case to conducting comparative bio-logging studies. Most tools are applicable to a variety of tag platforms, but example data and platform-specific information, including integration with various video options, is provided for video tags developed by Customized Animal Tracking Solutions (CATS, www.cats.is) and deployed via suction cups on cetaceans. Examples and assumptions below reflect the nature of the example data (e.g., we refer to animal depth repeatedly, which would not be relevant to deployments on terrestrial animals, and assume regular returns of the tagged animal to the surface to acquire oxygen). This manuscript provides information on downloading, importing, calibrating and processing tag data, using both custom-written tools and tools that have been publicly shared (primarily from the Animal Tag Tools Project at http://www.animaltags.org/ and the MATLAB File Exchange at https://www.mathworks.com/matlabcentral/fileexchange/). We additionally discuss tools that can be applied to the processed data for displaying and interpreting animal data. Example applications are specifically focused on cetacean deployments, but many can be adapted for other species.

Methods

Below we describe the theory and basic operations involved in processing data from inertial measurement units (IMUs). The primary sensors discussed are 3-axis accelerometers, 3-axis magnetometers and a pressure sensor for measuring depth, but our scripts also allow integration of video, audio, gyroscope, light and temperature data. Step-by-step instructions including additional figures, video tutorials, updated code and frequently asked questions are available through the webpage of a workshop offered in December 2020 (https://catsworkshop.sites.stanford.edu/). The workshop's home page contains direct links to the MATLAB code repository (https://GitHub.com/wgough/CATS-Methods-Materials), which contains a step-by-step wiki,[1] as well as to a dryad depository (https://datadryad.org/stash/share/KFi8G5QC7DFPYXynQeSotxtXqANZL70LFUGEiiDTSMU) with example data for a user to practice with. We discuss tag processing in four parts:

[1] Note that GitHub uses the term "wiki" to mean a place for users to write long-form content about their project (see: https://docs.github.com/en/communities/documenting-your-project-with-wikis/about-wikis). This is different than the traditional implication of a collaborative document. We use the GitHub parlance throughout for consistency.

I. Downloading, viewing and importing tag data.
II. Bench calibrations for individual tags.
III. Calculating orientation, motion and position (the "prh.mat" file).
IV. Applications (see "Results" section).

Part 0—platform requirements and setup

The described tools have been tested on a Windows system running MATLAB versions from 2014 to 2021a. With some third-party exceptions, such as Trackplot, most described packages can be run on other systems, such as Macintoshes, but have not been tested; some known compatibility issues are listed in the Discussion. All MATLAB tools described herein are stored at the above GitHub link, allowing for a living, open-source set of tools, where version history can be tracked and updates from collaborators are encouraged [23, 24]. To install code, we recommend that users download the GitHub desktop client (https://desktop.github.com), which allows code to be updated to match the current online version, ensuring seamless updates. The MATLAB environment then needs to be pointed to the GitHub folder by either adding the folder directly to the path upon opening MATLAB, or creating or editing a **startup.m** file in the default MATLAB directory that includes a line pointing to the tools folder, e.g., "addpath(genpath('C:\Users\Dave\Documents\GitHub\CATS-Methods-Materials\CATSMatlabTools'));". Example data used in the tutorial can be accessed through the above-linked dryad depository.

Though scripts are designed to be folder-structure independent, generally they work more seamlessly and efficiently, with fewer necessary user inputs, if scripts and data are organized according to the file structure outlined on the tag wiki. A compressed ".rar" file containing a template folder structure and a template TAG GUIDE to store metadata is available in the "templates" folder in the CATSMatlabTools.

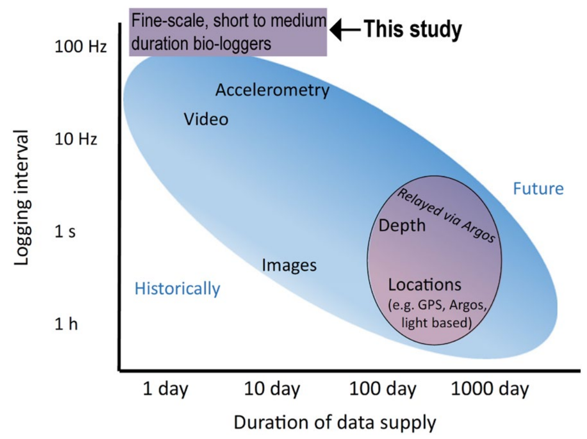
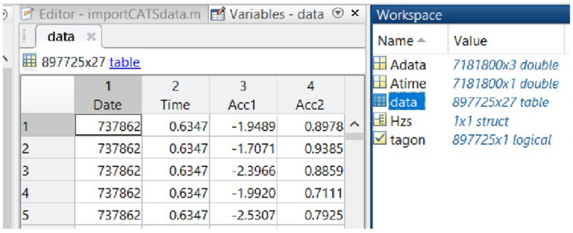


Fig. 1 Bio-logging data typically involves trade-offs between sampling resolution and sampling duration. Recent advances have allowed sampling at high resolution over longer time scales. This study provides tools for analyzing data from these high-resolution devices. Figure modified with permission from Fig. 1a in Hays [28] under Wiley publishing license number 5030590588688



897725x27 table

	1 Date	2 Time	3 Acc1	4 Acc2
1	737862	0.6347	-1.9489	0.8978
2	737862	0.6347	-1.7071	0.9385
3	737862	0.6347	-2.3966	0.8859
4	737862	0.6347	-1.9920	0.7111
5	737862	0.6347	-2.5307	0.7925

Name	Value
Adata	7181800x3 double
Atime	7181800x1 double
data	897725x27 table
Hzs	1x1 struct
tagon	897725x1 logical

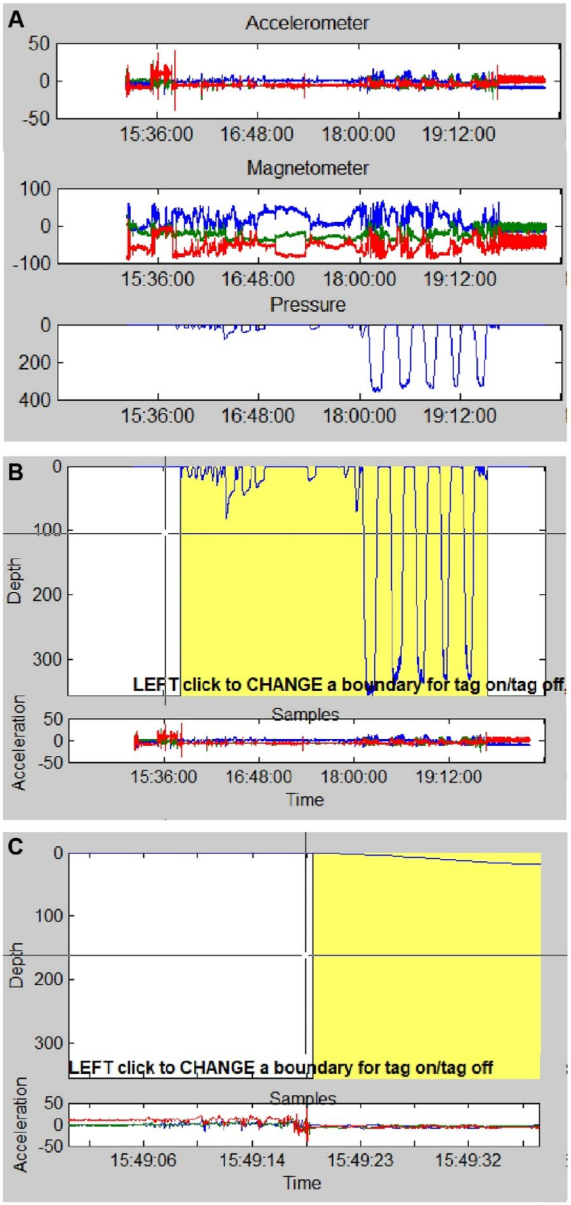
Fig. 2 MATLAB variables created from **importCATSdata.m**. These are the raw outputs from the tag imported into a *data* table that can be used for downstream tag processing across tag types. *Adata* and *Atime* are the accelerometer data and timestamps, respectively, maintaining the original sample rate of the accelerometer, whereas *data* are sampled at the highest non-accelerometer sample rate (or a user-defined sample rate). *Hzs* is a structure containing the sensor sample rates, and *tagon* is a user selected logical (Boolean) index of whether the tag is on the animal at any given data point (Fig. 3B). *data.Date* and *data.Time* are whole and fractional days since January 0, 0000 (MATLAB date number format)

Part I—downloading, viewing and importing tag data

With the miniaturization of storage chips and batteries, the amount of data that can be collected at high resolution has increased rapidly (Fig. 1). For example, daily diaries from Wildlife Computers can record 32 Hz accelerometer and pressure data for weeks at a time [25, 26], suction-cup attached CATS deployments have remained on animals for upwards of 96 h in Arctic and Antarctic waters, and DTAGs attached to seals have recorded 240 kHz acoustics and 200 Hz IMU data for 21 days [27]. Each tag manufacturer copes with this data abundance in a proprietary way, usually by compressing data to maximize storage capacity and minimize download time in a way that also minimizes errors during data write.

Because the format of raw data varies across tag type, a critical step is to import data into a common format to facilitate downstream processing using the same tools. Our import scripts conglomerate data into two formats: (1) an *Adata* matrix and corresponding *Atime* vector with the accelerometer data at its original sampling rate—often much higher than other sensors to facilitate detection of low-frequency vocalizations [29] or to estimate speed from the amplitude of tag vibrations [18] and (2) a data table format with common header names (Fig. 2). We provide import scripts for CATS, Acousondes [30], Little Leonardos [31], Wildlife Computers' TDR-10s and SPLASH tags [25], and Loggerhead Instruments' openTags [32], and other tags can be processed using subsequent scripts if the data is organized with the variable names described above. The CATS data we describe in detail offloads from the tag as a series of CSV (comma-separated value) files, and the script **importCATSdata.m** automatically combines these files into a single data table. Early versions of CATS tags offloaded in a single CSV file which needed to be broken up into smaller formats before import (e.g., using CSV splitter: https://www.erdconcepts.com/dbtoolbox.html) due to memory constraints. Also generated by the import script is a variable (*Hzs*) that reports the original sample rate for all of the sensors, allowing for downstream matching of sampling rates using the **dec_dc.m** and **interp2length.m** scripts from http://www.animaltags.org/.

Our best practice recommendation is to run the import script (e.g., **importCATSdata.m**) immediately upon download of data from a recovered tag. Embedded in the script are three tools that can aid the researcher in the field: (1) a plot of depth vs time that can allow for a first run estimation of animal behavior (Fig. 3A); (2) a plot of other critical sensors (specifically accelerometer and magnetometer data) to gauge immediately whether there may be errors in the deployment data or whether the tag is okay to deploy again in the field (Fig. 3B); and (3) a "tag on time" tool that allows for precise determination of the tag on and tag off time, which can be useful to help researchers adjust deployment procedures (Fig. 3C).



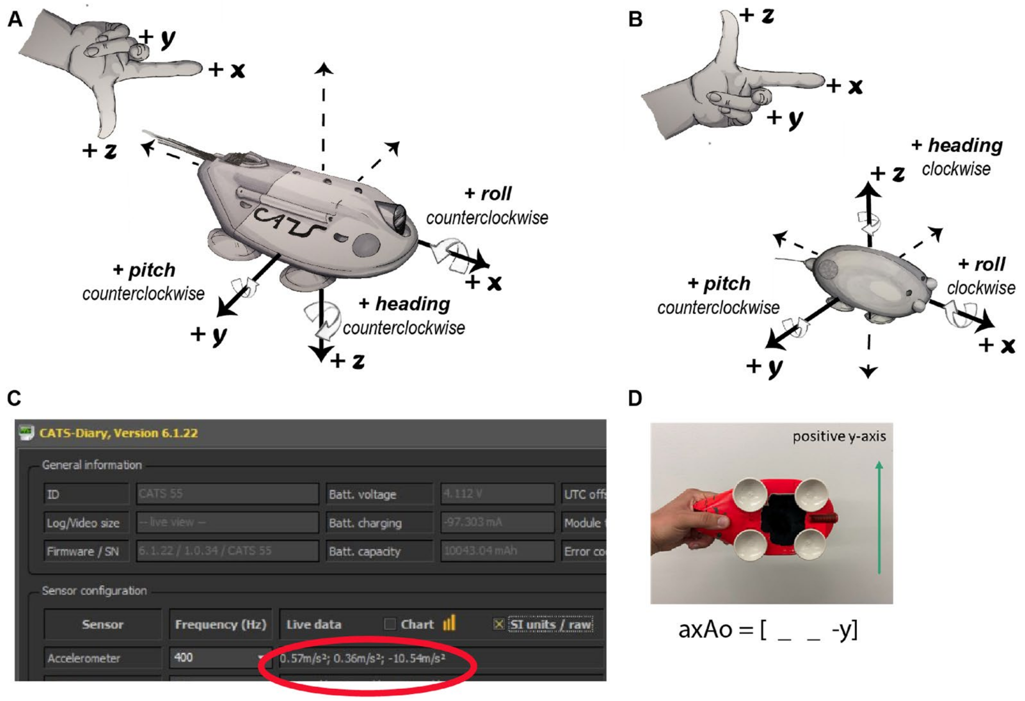
LEFT click to CHANGE a boundary for tag on/tag off

LEFT click to CHANGE a boundary for tag on/tag off

Fig. 3 Plots from **importCATSdata.m**. **A** Accelerometer, magnetometer and pressure data are plotted so that a user can determine at a glance if the deployed tag collected critical data as expected. This can inform future deployments. **B** Using the tag pressure sensor and accelerometer, a precise tag on and tag off time can be determined. **C** User can use graphical controls (right clicking in this case) to zoom in to the plot for fine scale determination of tag on and off times

Part II—bench calibrations (performed once for each individual tag)

A critical aspect of bio-logging is comparing data across deployments and individuals. While some derived data sets will always be tag-placement dependent—e.g., Overall Dynamic Body Acceleration (ODBA) [33, 34] and Minimum Specific Acceleration (MSA) [21]—many data streams are comparable across deployments provided that units are consistent and accurate, which requires calibration. Before deploying a tag, we recommend applying a series of bench tests to (a) determine the tag-specific axis conventions of each device (Fig. 4); (b) convert the engineering units into consistent scientific/engineering units (typically SI, though we relate acceleration to the acceleration due to gravity); (c) provide a baseline calibration to increase the accuracy of deployment-specific calibrations using in situ data; (d) test the flotation of tags before deployment to ensure recovery antennae are maximally extended above the surface; and (e) test the recovery methods (e.g., ARGOS or VHF). Although calibration steps a–c can also be completed after a tag has been deployed in the field, the chance of tag loss or malfunction after a first deployment, but before calibrations can be done, is substantial. Some tag manufacturers provide bench calibrations and information about



CATS-Diary, Version 6.1.22

General information

ID	CATS 55	Batt. voltage	4.112 V	UTC off
Log/Video size	-- live view --	Batt. charging	-97.303 mA	Module
Firmware / SN	6.1.22 / 1.0.34 / CATS 55	Batt. capacity	10043.04 mAh	Error co

Sensor configuration

Sensor	Frequency (Hz)	Live data ☐ Chart ☒ SI units / raw
Accelerometer	400	0.57m/s²; 0.36m/s²; -10.54m/s²

axAo = [_ _ -y]

Fig. 4 Axis conventions. **A MainCATSprhTool.m** analyzes tags with a right-hand orientation such that rotation around each axis is counterclockwise (when viewed from the positive direction of the rotation axes), and heading and pitch have intuitive orientations (+ pitch is up, + heading is like a compass). In standard position (e.g., a whale at the surface), [x y z] = [0 0 − 1] g, where g is acceleration due to gravity. **B** Standard DTAG processing, as utilized by the scripts available at http://www.animaltags.org/, uses a left-hand orientation with heading and pitch oriented intuitively and roll assigned arbitrarily to be clockwise (to the animal's left). In standard position [x y z] = [0 0 1] g. To convert between CATS conventions and DTAG conventions, multiply *z*-axis values and roll by − 1. **C** Live view display of a tag flat on a table as in panel A, whose axis conventions align with the processing conventions. **D** If instead the display in C is for a tag oriented as in the image, it implies that the third axis is actually displaying the –y orientation, so the *axAo* variable in axisconventions.m would need to be adjusted as in Eq. 1. In this example, the first two positions are left blank, because they have not yet been tested. Illustrations by Jessica Bender

axis conventions with each tag purchased. In these cases, though it may still be useful to test the tag to confirm the calibrations, it may also be sufficient to construct calibration matrices without running additional tests. Examples are below and through the wiki to construct rotation matrices that can be applied to rotate manufacturer's recommendations to the conventions used in downstream processing (described below).

Axis conventions are mathematically arbitrary, though for convenience in processing it is useful to have the same conventions across deployments. In downstream processing, our scripts assume a north-east-down (NED) orientation (Fig. 4), such that the first sensor axis (*data.Acc1* or *Adata(:,1)*) is the *x*-axis, reading positive values when the front of the tag is facing up (opposite the direction of the force of gravity), the second sensor axis (*y*) faces to the tag's right (positive when the right side of the tag faces up), and the third axis (*z*) points down (positive when the tag is on its back with the bottom side facing up). Although the assignment of axis conventions is arbitrary, the principles that inspired our choice of a right-handed, NED orientation were: (a) all rotations should be in the same direction around an axis—we chose counter-clockwise to match conventions from trigonometry; (b) we wanted pitch to be positive when an animal is ascending to the surface (rostrum facing towards the surface); and (c) we wanted animal heading to match conventional compass bearings. Roll in the scenario is determined by the above constraints and ends up being positive when rolling to the animal's right. Other commonly used axis conventions do not meet all of these criteria. For instance, a north-east-up (NEU) convention—the baseline in DTAGs and some other tag types—with the same (b) and (c) restrictions forces pitch and heading to have directionally opposite rotations (counterclockwise and clockwise, respectively, Fig. 4b), and the roll is arbitrarily determined in DTAG nomenclature to be to the animal's left (clockwise rotation). To convert between an NED and an NEU reference frame (e.g., when using DTAG scripts with CATS data), switch the sign of the *z*-axis in all sensors, and note that roll calculated with our tools would have opposite sign of roll calculated with DTAG tools. Note, however, that tools that use a netCDF structure format (described below), usually have axis convention information embedded in the structure for each sensor, so users should be aware if no adjustments are necessary.

An individual tag's internal axis conventions depend on the orientation of the sensor package within the tag; however, so different tag versions may arrive to the user with different axis conventions. That is, though our analysis scripts assume an NED orientation, the raw data exported from the tag could have the *x*-values in any of the three data columns, for example, and those values could have the opposite sign from our assumptions. The first step we outline on the wiki site, then, is to determine the axis conventions used by the tag by maneuvering the device through a series of static positions (for the accelerometer and magnetometer) and motions (for the gyroscope) to reveal how the sensor package is arranged within the tag. The user is then prompted to edit the script **axisconventions.m** to account for any deviations from the convention (in which the first axis reads x, second y and third z). As an example, if an uncalibrated tag displays positive values in the first column of the accelerometer matrix when the tag is upside down (and zero in the other 2-axes), positive values in the second column when the tag has the anterior side facing to the sky, and negative values in the third column when the tag is on its left side, a user would edit **axisconventions.m** to define the original axis accelerometer axis conventions (*axAo*) as

$$axAo = [\mathrm{z\ x\ -y}] \tag{1}$$

and the script would automatically calculate a rotation matrix (*axA*) that is right-multiplied by the raw tag data in downstream processing:

$$axA = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}. \tag{2}$$

Then, once these values are stored in an individual tag's "<tagID>cal.mat" file, downstream processing that imports that calibration file will automatically correct the tag's internal alignment to be consistent with all processed data, ensuring that axis conventions do not have to be considered in future processing steps.

Other calibration steps are detailed in the tag wiki. They involve using the earth's gravitational and magnetic fields, which have known values at given locations, to convert raw sensor data into engineering units. The main **calibrateCATS.m** script guides users through the application of calibration steps to collected data, relying on the spherical calibration scripts from http://www.animaltags.org/ to create the base calibration procedures.

Calibration files can also contain other information, such as pressure and temperature factors and offsets, which are specific to individual tags. For example, in our cal files, the user can set *pcal* (a pressure factor) and *pconst* (a pressure offset). Many tag types, including CATS tags, include pressure and temperature factors preprogrammed into the default outputs by the manufacturer, but others may need to be written into the cal files or may need bench calibrations. Float tests are additionally recommended in the water conditions for each environment in which tags will be deployed as even small differences in water density can affect the vertical tilt and flotation of the tag (Fig. 5).

Part III—calculating orientation, motion and position (the prh file) for each deployment

The **MainCATSprhTool.m** script is divided into sections (termed "cells" in MATLAB parlance), each of which performs a set of tasks that lead to the "prh" (pitch-roll-heading) file with the filename "<*deploymentID*> <*samplerate*>prh.mat". For example, the example data results in a file name of "mn200312-58 10Hzprh.mat", where mn200312-58 is the deployment ID consisting of a species ID ('mn' for *Megaptera novaeangliae*), a date in YYMMDD format and a tag ID number (58) of the specific device deployed on that animal, and the sample rate of the resulting file is 10 Hz. The prh file concept was introduced for DTAG processing by Mark Johnson [20, 7] and the output files from our scripts are designed to be compatible with DTAG prh files (though see note above and Fig. 4 regarding axis conventions). The prh file contains additional variables that are the basis for biological studies, such as speed, depth, time, accelerometer and magnetometer readings in the animal's frame of reference, and northing and easting distance from the start position. A full list of variables created in this process are defined in the file "CATSVarNames.txt" in the GitHub repository.

Each cell of **MainCATSprhTool.m** performs a discrete set of tasks that build on each other. The process can be paused at any point and progress is automatically saved in an "INFO.mat" file that can be used to return to previously completed steps of the process and make edits. The beginning of each cell contains some parameters that can be adjusted depending on the deployment type. For instance, deployments without video can set the variable *nocam* to true in cell 4, which then triggers a simpler version of many processes, and creates empty variables when video-specific values are called for.

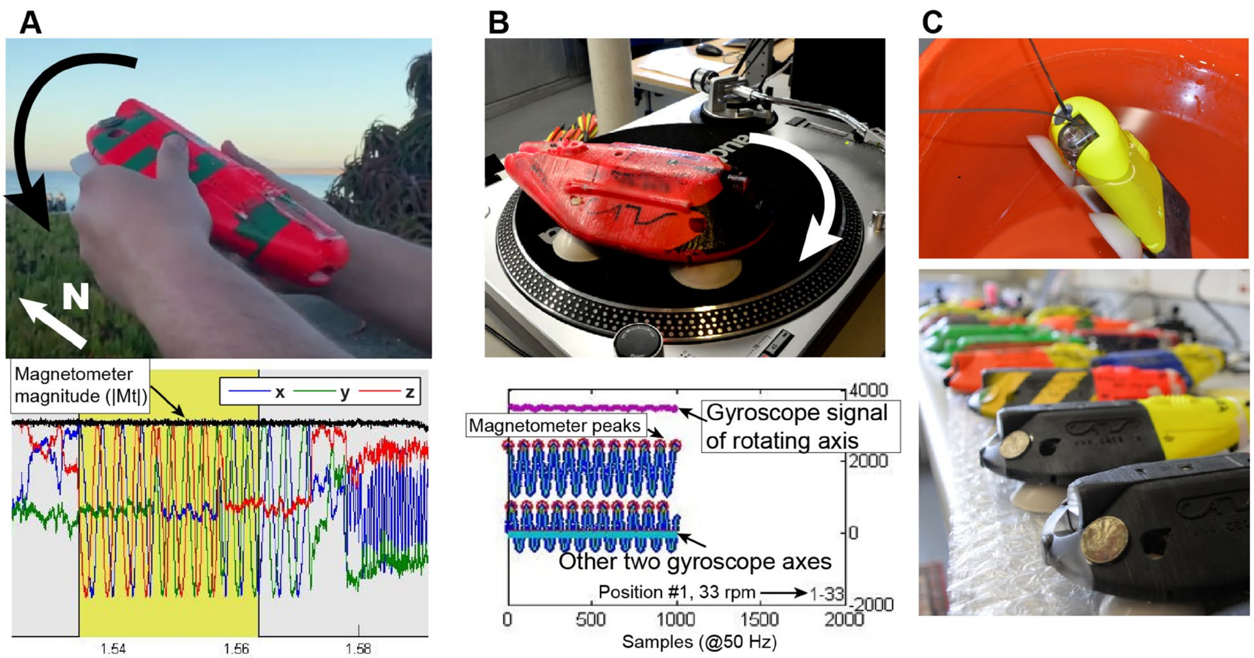


Fig. 5 Examples from **calibrateCATS.m**. **A** Magnetometer calibrations involve rotating the tag around the 3-axes of rotations in line with magnetic north. Bottom graph is a plot of the triaxial magnetometer data after the calibration is applied such that the overall magnitude of the 3-axes (the vector sum, |M|) is constant. **B** Gyroscope calibrations involve spinning the tag in six different positions (positive and negative for each axis) at two speeds. The actual speed is calculated from the peaks in the magnetometer data as 2-axes rotate through north and south poles. **C** Checking the flotation of new tags is critical. Bottom image—occasionally a small amount of ballast (in this case two US quarters) may need to be added for tags that were designed in warm water that are deployed in cold, more dense water to ensure that tags float upright (but still float)

IIIa—video processing

We bring up the camera variable example, since a specific focus of these tools is the integration of video data with inertial sensor data. From a theoretical perspective, video data should be relatively straight-forward to work with, despite the basic constraints of increased storage and battery needs. If you know the start time of a video and the video's sample rate in frames per second (fps), nominally the video can be easily lined up with other data streams. However, in practice video data is particularly prone to several sources of error, the most challenging of which are time offsets from when the video is signaled to "turn on" to when it starts recording, as well as skipped frames when the processor is overloaded—a common occurrence in visually complex pelagic environments with light conditions that change rapidly as an animal changes orientation in three dimensions (Additional file 1: Video S1). Video recorded on commercial consumer devices is typically "finalized", meaning that it stores metadata, such as file duration, in the file and can be read by a variety of media types and is typically free from errors. Early versions of CATS tags utilized off-the-shelf products from GoPro, Oregon Scientific and others that finalize videos before writing to a memory card. The problem with finalized videos is that if there is a write error in any part of the video, e.g., from a sudden reduction in power supply, the video cannot be written and the whole video, which could be 30 min of data, is unreadable. To avoid this problem, modern CATS tags are unfinalized in the raw format that downloads from the tag, which means data is more likely to be available, but also more likely to have errors and can only be read by certain media players, such as VLC (https://www.videolan.org/vlc/) or MPC-HC (https://mpc-hc.org/) until they are processed further.

Cell 1 of the **MainCATSprhTool.m** script has a similar function as the **importCATSdata.m** script in that it can read a variety of video types and resolutions that have been included in various versions of CATS tags. This cell can be skipped if there is no video or audio in the utilized tag type. The basic functionality has two phases: (1) read in audio data from both audio and video files, storing ".wav" files and raw data in an "audioData" folder and (2) read the timestamps off each video frame for purposes of synchronization. Part 2 is driven by an **mmread.m** script (https://www.mathworks.com/matlabcentral/fileexchange/8028-mmread) that reads the encoded timestamp of each frame, and the workhorse function

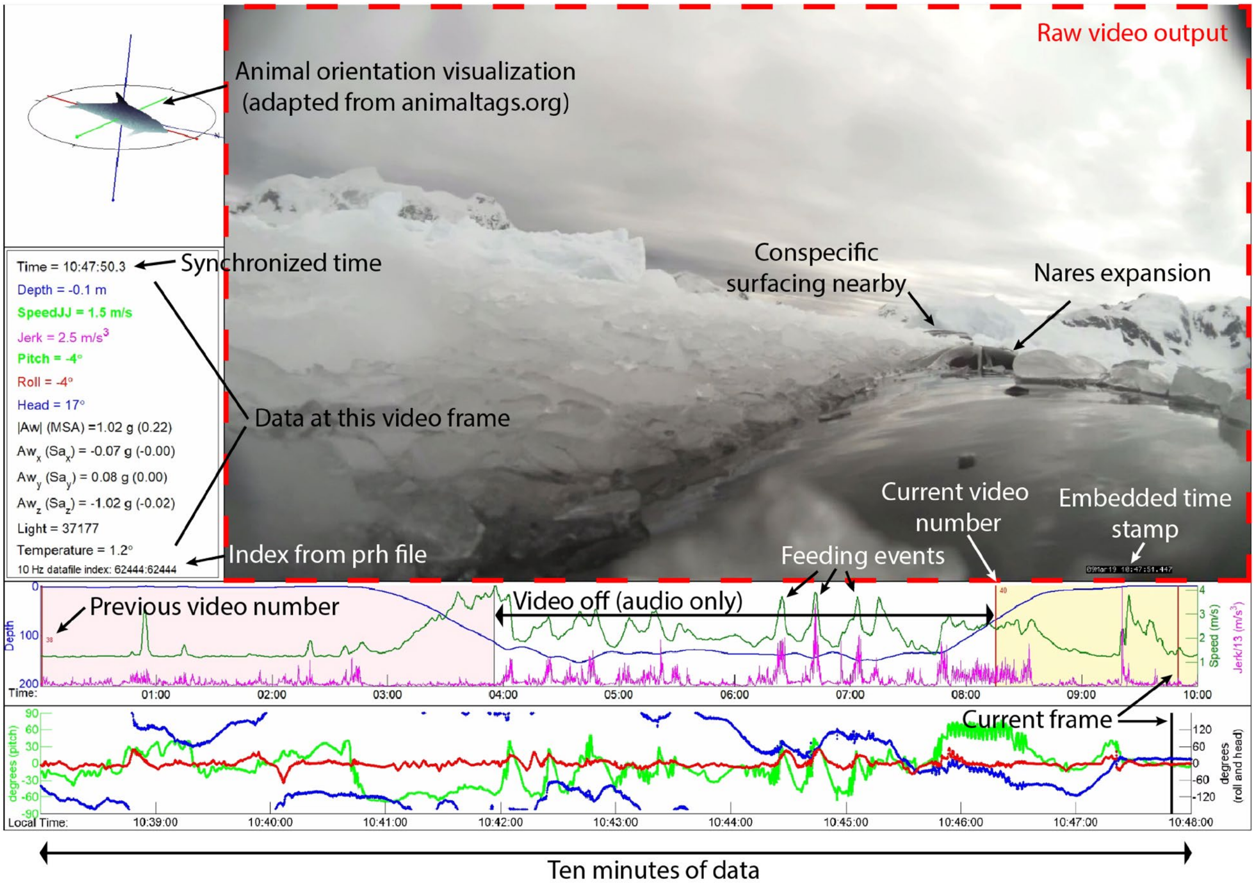


Fig. 6 Still from tag video from an Antarctic minke whale (deployment ID: bb190309-52) stitched together with tag data. The stitched video and data could be considered the final output of these tools (see Additional file 2: Video. S2, Additional file 3: Video S3). Red dashed line outlines the original video frame—the original resolution is maintained and processed sensor data is written onto the outer edge of the video. Ten minutes of data are displayed at a time, and a vertical line indicates the current time step. On import, cell 1 of **MainCATSprhTool.m** can use optical character recognition to read the embedded timestamp off each frame of the video (lower right corner). For this video, a processing delay in the camera firmware was later discovered. The corrected time is indicated in the data box on the left side. For most deployments the camera is set to stop recording when light levels drop below a threshold (around 100 m depth for this deployment)

makeMovieTimes.m also provides the option of using an optical character reader to directly read the embedded timestamp in the corner of the video (Fig. 6A). The final metadata about each frame is stored in a “movieTimes.mat” file that is read in later.

At the conclusion of the prh process, a **StitchDataonVideo.m** script allows for processed sensor data to be written on top of the video frame, expanding the frame size of the resulting movie to maintain original video resolution (Additional file 2: Video S2, Fig. 6B). This process facilitates biological interpretation of the video as well as video auditing, as the orientation, motion and depth data can easily indicate where points of interest are (e.g., feeding events that have characteristic signatures). This process requires significant computer processing time, working in small chunks (typically 10–15 s) of video, and creating a folder full of 15 s partial videos. On a typical personal computer with 32 GB RAM, it can take 10–20 min for 1 min of video to process, and requires a screen width at least 21.5% greater than the video frame width (or two monitors), as well as ~ 100 GB of hard drive space for every hour of raw video. The final step, synching the partial videos using Adobe Media Encoder or similar video stitching software, reduces the video sizes back to standard sizes (~ 3–4 GB/h of video). The resulting videos are finalized in “.mp4” format and can be read on any standard media player. The intermediately created videos can then be deleted.

IIIb—data processing

After cell 1, which could be skipped if there is no video or audio data, the remainder of the cells in

MainCATSprhTool.m should be run regardless of the specific data being processed. The **MainCATSprhTool.m** script guides the user step by step through the data analysis process in the following order (with additional details of critical steps below): cell 2 loads data; cell 3 loads calibration data and trims non-biological data from the raw data; cell 4 synchs the video and the data—for data that does not have embedded timestamps, there is an option to use another synchronization method (for whales this can be the times of surfacings observed in the video); cell 5 locates the precise beginning and end of the deployment in the data; cell 6 begins making the animal-frame variables and performs an in situ pressure calibration (using **fix_pressure.m** from http://www.animaltags.org/); cell 7 performs in situ calibrations of the accelerometer and magnetometer data using the **spherical_cal.m** scripts from http://www.animaltags.org/; cell 8 calculates the orientation of the tag on the animal and looks for places, where the orientation may have changed (tag slip); cell 9 is currently inactive, but remains as a placeholder for users who wish to more finely calibrate gyroscope data; cell 10 imports metrics of turbulence—acoustic flow noise [35, 19] and accelerometer jiggle [18]—that can be used as proxies for forward speed; cell 11 regresses those proxies against orientation-corrected depth rate (OCDR) [36]; cell 12 saves a simple version of the prh file that has comparable variables as DTAG prh files (though see notes on axis conventions above); cell 13 adds any tag-collected GPS data or other known animal locations into the prh file, then creates a geo-referenced pseudotrack of animal position [37]; and cell 14 summarizes the processed deployment information into a visual "quicklook.jpg" that allows a researcher to quickly scroll through deployments to see the critical data from each.

Specific guidance for implementing each step is available on the GitHub wiki, and we provide additional descriptions for some of the unique processing points below. Synching data with video in cell 4 has two main resulting variables: *vidDN* that records the start time of each video (where *DN* stands for Date Number, a MATLAB date-time format equivalent to days since the start of year 0), and *vidDurs,* the duration of each video in seconds. Occasionally, videos before deployment will be recorded but discarded. In this case, where, for example, video number 3 is the first video of the deployment, there would be a value of *NaN* (the MATLAB not-a-number signifier) in the first two entries of the *vidDN* and *vidDurs* vectors.

For most cetacean deployments, where the orientation of the tag on the target animal cannot be finely controlled due to the deployment method on free-swimming animals, the tag axes must be mathematically rotated so that they align with the animal axes in NED orientation. Johnson and Tyack [7] refer to this process as rotating the tag's frame of reference (tag frame) to the whale's frame of reference (whale frame). The procedure we use (based on [20, 7], is similar in theory to the currently available **prhpredictor.m** tool from http://www.animaltags.org/, though our **estimateprh.m** script also directly includes the ability to detect and modify tag slips, an increased ability to zoom in and out of data regions, and has more thorough descriptions of the user controls displayed directly on the plots. Mathematically, the procedure involves calculating a rotation matrix, W, that is the product of a rotation matrix that accounts for pitch and roll of the tag (W_{pr}) and a rotation matrix (W_y) that accounts for the yaw of the tag in relation to the whale's axes (Fig. 7). W can then be applied to the tag sensor data for the accelerometer (At), magnetometer (Mt) and gyroscope (Gt) to create the whale frame (or animal frame) matrices Aw, Mw, and Gw. The **estimateprh.m** script in step 8b of **MainCATSprhTool.m** calculates these matrices automatically by asking a user to identify periods of time when the animal is thought to be in a "typical" orientation—that is, when its body is aligned with the earth's frame of reference (which often occurs for whales, while they are breathing or just between breaths)—and constructing a rotation matrix that rotates the tag data during that time such that the z-axis reads − 1 g (Fig. 7A). This does not resolve the orientation of the x- and y-axes, however, so an additional period of time, often as the animal finishes an ascent to the surface or starts a descent from the surface, where the animal can be assumed to be rotating nearly exclusively around the y-axis (a change in pitch) is identified and used to adjust the rotation matrix in the yaw direction until rotation around the y-axis is isolated during the identified period (Fig. 7B, C). This procedure involves some amount of user selection of the defined period and an understanding of "typical" cetacean behavior. To limit the amount of trial and error, which may be especially difficult for deployments with a lot of tag motion (i.e., tag slips resulting in substantial changes in tag frame relative to whale frame) over time, our script allows for iterative changes to the user selections and immediate feedback of the resulting calculated Euler angles (pitch, roll and heading, Fig. 8). If the tag moves at all during a deployment, as is common in tags attached with suction-cups, the rotation matrix must be calculated for each distinct period of tag orientation. Our script allows for tag slips to be identified using either the accelerometer data or video data, and the prh estimator allows for those identified tag slips to be adjusted. If tag slips take place over a period of time, the prh estimator calculates a unique rotation matrix for each time step between the start and end of the slip, using the calculated rotation matrices as the start and end points (Fig. 8B).

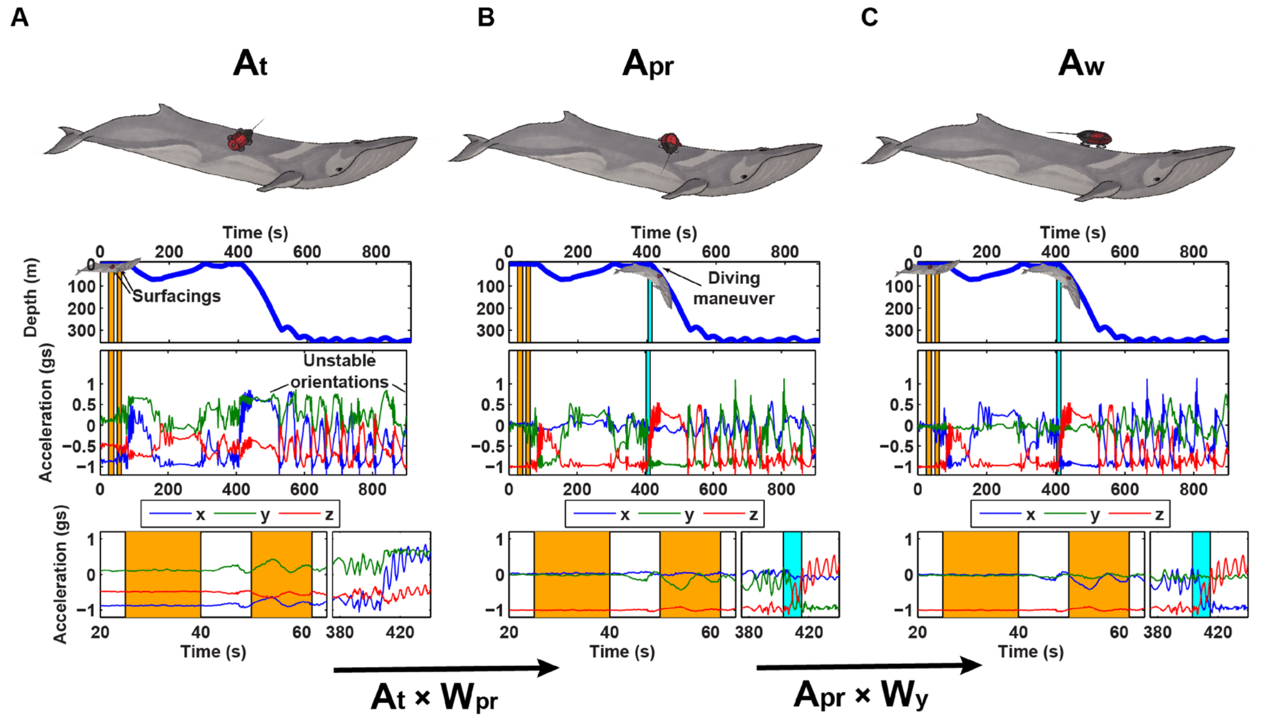


Fig. 7 Orienting tag frame to animal frame. For cetacean tagging, the orientation of the tag on the animal cannot always be finely controlled (Fig. 8A). Similar reorientation procedures that we describe can be used for tag on other animal species, where tag axes cannot be affixed to align with the animal axes. **A** *At* data displayed for a tag that is deployed on an animal with Euler rotations of 150° in the yaw direction, − 60° in the pitch direction and − 10° in the roll direction relative to animal frame. Orange boxes highlight surfacing periods, where the animal is relatively stable and averages an orientation commensurate with the navigational frame of reference (note that the animal does not have to be as still as in this example for this procedure to work). Blue box highlights a period at the start of a dive, where the whale should be rotating around the *y*-axis (i.e., the *y*-axis should remain stable in whale frame with *x*- and *z*-axes changing as their measurement of gravity changes). **B** Rotation matrix W_{pr} is constructed to mathematically rotate tag frame to the top of the whale, with *z*-axis ≈ − 1 g during surfacing periods. **C** Rotation matrix (W_y) is constructed to rotate the tag *x*- and *y*-axes to align with the whale frame such that the *y*-axis has minimal change during the diving maneuver as its relation to gravity should be stable. **MainCATSprhTool.m** accounts for all of these rotations automatically in the sub-function **tagframe2whaleframe.m** that is run as part of cell 8. Illustrations by Jessica Bender

Animal speed through the water can be difficult to measure directly (though see sensors described in [39, 35, 40, 41, 42, 43], particularly for deployments, where tag orientation on the animal is unpredictable or the flow over the sensor structure varies from laboratory conditions [43]. If a speed sensor is included on a bio-logging device our process allows for easy inclusion in the prh file as a *speed* variable—a table with various columns representing different speed metrics and the associated prediction errors. If a speed sensor is not included, as in typical CATS tags, cell 10 steps the user through analysis of two metrics of turbulent noise that have been shown to increase commensurate with animal speed: flow noise over a hydrophone [44, 45, 35, 9] from acoustic files and the vibrations of the tag as measured by high sample rate (preferably ≥ 50 Hz) accelerometers [18]. Cell 11 then regresses those metrics against periods of steep ascent or descent, where speed can be estimated from changes in depth (as in [19, 36]. The **speedfromRMS.m** script provides increased flexibility to adjust pitch and depth restrictions for OCDR calculations (Fig. 9) than an earlier version published in Cade et al. [18]. During the processing of acoustic files for flow noise and the alignment of acoustic files—which in CATS tags may have temporal gaps between the files—with the sensor data, the **stitchaudio.m** script combines all the audio into a single file, a useful tool for acoustic auditing.

After a basic prh file is created in cell 12, cell 13a adds in any surface position data available from surface observations (e.g., [46, 47], or on animal GPS locations (usually from a fast-acquistion system, e.g., FastLoc: [48, 49]. Smoothing the speed, pitch and heading data (by first low-pass filtering accelerometer and magnetometer matrices using a finite impulse response filter, available at http://www.animaltags.org/ and then recalculating orientation) allows for a track of the animal to be estimated from motion data using the http://www.animaltags.org/ script **ptrack.m**. Using the known surface positions, the

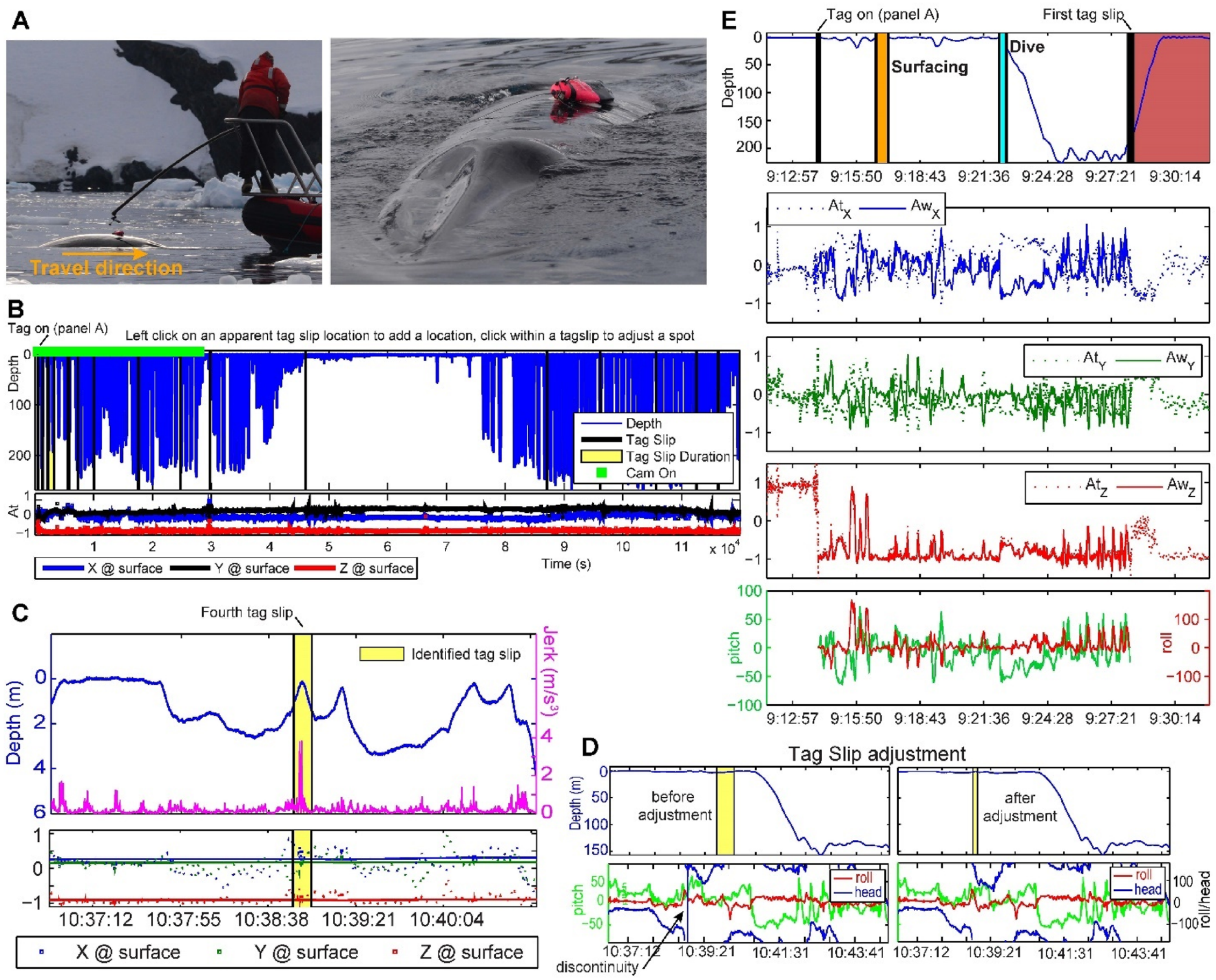


Fig. 8 Tag orientation correction user interface. **A** For this example, a friendly minke whale (bb190309-52, also see Figs. 6, 12) approaches directly at the tagging boat, resulting in a tag on the whale in reverse orientation from the whale's natural axes (see Fig. 4). **B** Step one is to identify the approximate locations of tag slips. Exact times can sometimes be seen on tag videos, or can be inferred from where the tag's surface accelerometer values change. **C** Cell 8 of **MainCATSprhTool.m** facilitates zooming in on tag data to identify likely tag slips, often corresponding to rapid changes in acceleration of the tag (increased *jerk*, see [38]). **D** In cell 8b, when tag frame is rotated to whale frame (Fig. 7), the calculated pitch, roll or heading can be used to indicate probable tag slip locations as well, as a discontinuity is often a sign of a tag slip. **E** User selected surfacings and dives (Fig. 7A, B) give immediate feedback to the user on the final rotated frame of reference (*Aw*), as well as the calculated animal pitch and roll. In this example, the x-accelerometer is rotated from backwards to forwards (aligned with the whale's frame of reference), with very few changes to the *y*- or *z*-axes. *Pitch*, *roll* and *Aw* are not yet calculated for the red highlighted period after the first tag slip

error accumulated from integrating the motion data is smoothed between known positions (as described in [37]) to generate an estimate of position using the provided script **gtrack.m** as part of cell 13b. The resulting *geoPtrack* provides x (Eastings), y (Northings) and z (depth) values in meters from the start of the track. Given a known start position, our scripts provide code at the end of cell 13b that can convert this track into a GPS position at each time step. It should be noted that without sufficient surface positions, this process can diverge from the true position quickly due to the repeated integration of small errors. However, with sufficient anchor points, this process can create a robust estimate of position (Fig. 10). The number of points that is "sufficient" will depend on the accuracy of the speed metric, pitch and heading determination, as well as the presence of any subsurface currents (as integrated inertial sensors will, even if perfect, only give position through the water). A user can test the expected accuracy of the track between known positions by comparing the calculated

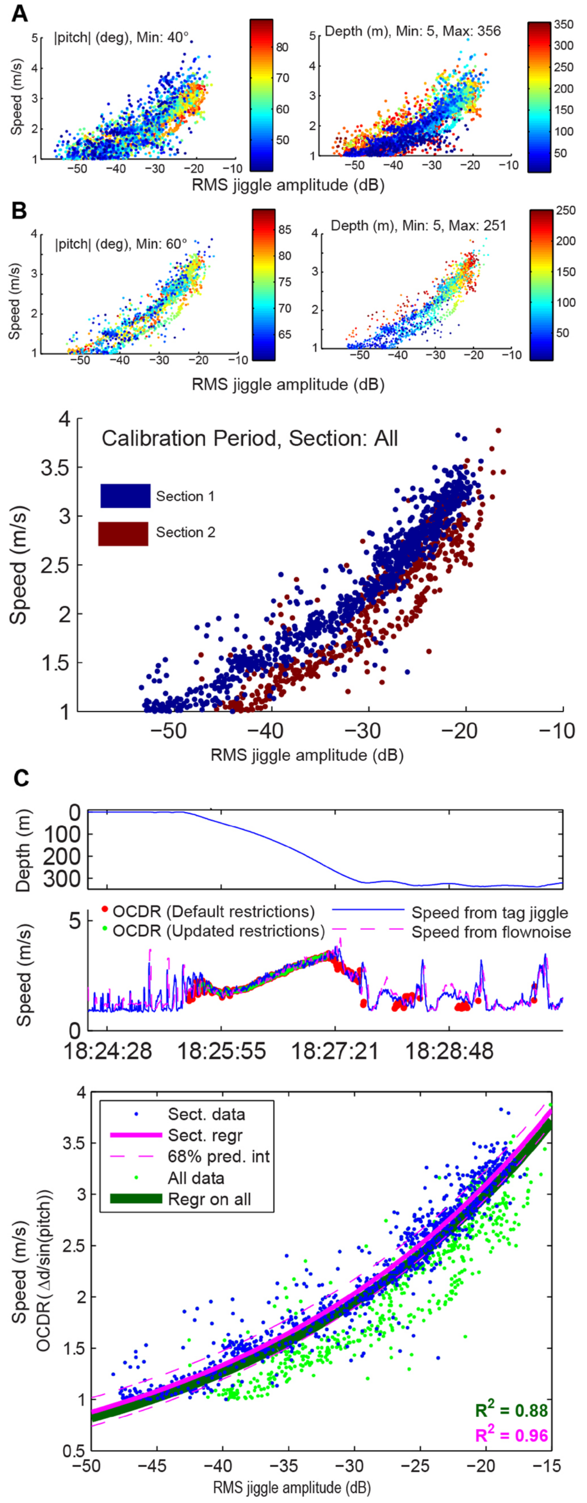


Fig. 9 Speed calibration curves for deployment mn200312-58. Plots result from cell 11 in **MainCATSprhTool.m**. "Speed" in all cases is the estimated speed from orientation-corrected depth rate (OCDR). Steep descents or ascents are necessary to have accurate estimations of speed using this method. **A** OCDR vs amplitude of tag vibrations as measured by the accelerometer (tag jiggle), colored by animal pitch and animal depth using the default restrictions (|pitch| > 40°, depth > 5 m). **B** User interface allows for clicking on the color bar to increase the restriction to exclude points, where OCDR is less accurate as a metric. In this panel, restrictions were updated to |pitch| > 60°, 5 m < depth < 251 m. Lower panel shows the separation of the data into two distinct calibration sections that result from different orientations of the tag on the whale (thus different turbulent flow regimes causing different relationships with speed). **C** Final check that plots speed derived from a regression with tag jiggle as well as speed derived from a regression with flow noise against individual OCDR-derived speed estimates as a time series. Bottom panel shows the regression and correlation coefficients for the regression on just this section's data (pink line and blue dots) as well as if all data from the deployment are used (green line and dots)

pseudotrack to the calculated georeferenced pseudotrack to determine how quickly the track diverges from known positions. This process also allows for some flexibility, depending on the research question, to iteratively adjust the track to account for obvious errors (such as going over land). For instance, in an environment adjacent to a complex shoreline, an animal's movement in a track that parallels the contours of a shoreline may be able to be used as an approximate anchor point for the track (e.g., [50]).

Results

In the last two cells of **MainCATSprhTool.m**, our scripts create a series of files that can be used to view the data in a number of different formats outside of MATLAB. The format recommended by Sequeira et al. [17] as a standard for sharing bio-logging data is the netCDF structure, a data-standard developed by UCAR/Unidata (http://doi.org/10.5065/D6H70CW6) that is portable (i.e., machine-independent) and self-describing. Each netCDF file (in the form "<*ID*>_prh<*samplerate*>.nc") contains all of the data arrays from the prh file (e.g., pitch, roll, heading, depth, etc.) as well as a metadata structure with information about the deployment using the conventions described at http://www.animaltags.org/. As an alternative portable output, cell 13c also writes a "txt" file with speed, depth, accelerometer data and orientation information. This file is specifically formatted for use with Acqknowledge software (BioPac Systems Inc.), but can be read by a variety of other platforms. Header descriptions are included in the "templates" subfolder of the GitHub repository. Cell 13c also creates a "txt" file with a

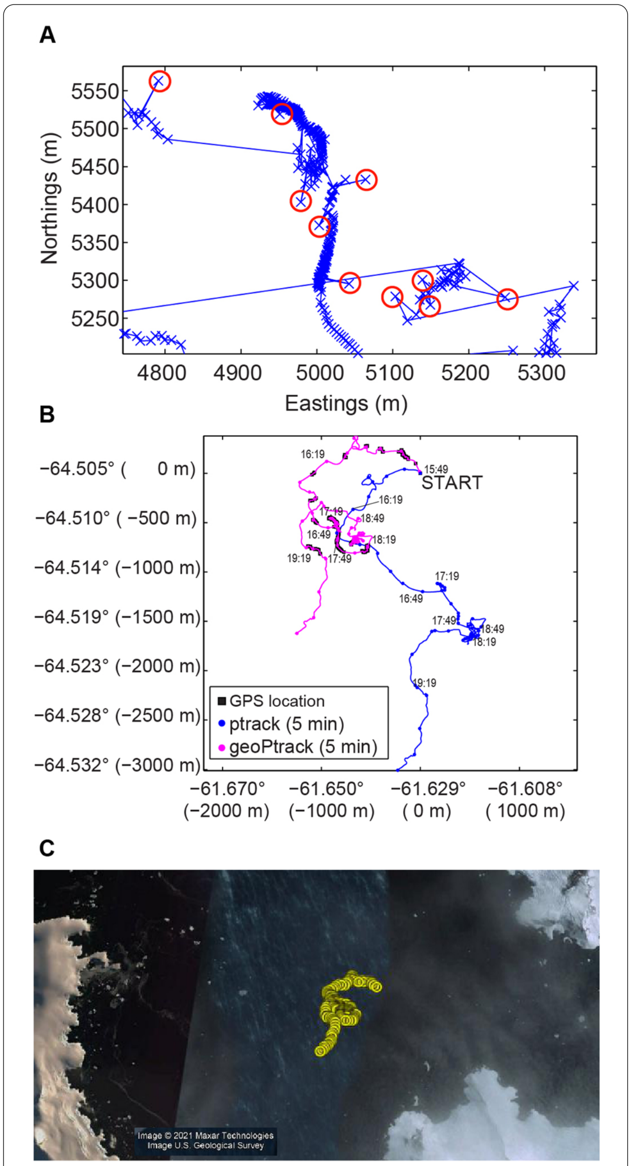


Fig. 10 Creating animal tracks from inertial sensor and GPS data. **A** GPS points received on deployment mn200312-58 by a fast-acquisition GPS system used by CATS for taking snapshots of satellite positions during animal surfacings. Depending on the threshold used for removal of erroneous GPS points, some erroneous points (red circles) may need to be manually removed. For display, only the fine scale UTM coordinates are listed. To locate this plot in space, add 2840 km to the northings and 560 km to the eastings in UTM zone 20D. **B** Geo-referenced pseudotrack (*geoPtrack*) diverges from the pseudotrack created from the inertial data alone. **C** **MainCATSprhTool.m** leads the user through creation of a ".kml" file for easy processing of spatial data (here displayed using GoogleEarth)

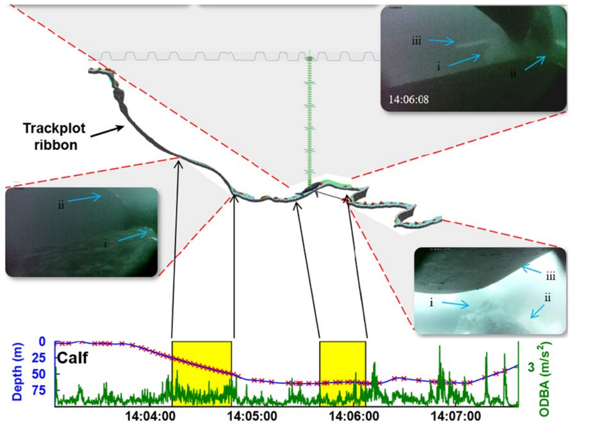


Fig. 11 Trackplot [22] can be used to visualize tag data using the outputs of the **MainCATSprhTool.m** script. This plot, cropped from Fig. 1 in Tackaberry et al. [51] under creative commons CC-BY 4.0 (https://creativecommons.org/licenses/by/4.0/), shows a humpback whale calf's depth, ODBA and fluke strokes (red xs) collocated along its mother's Trackplot during times when it can be seen nursing (blue arrows in images point to spilled milk)

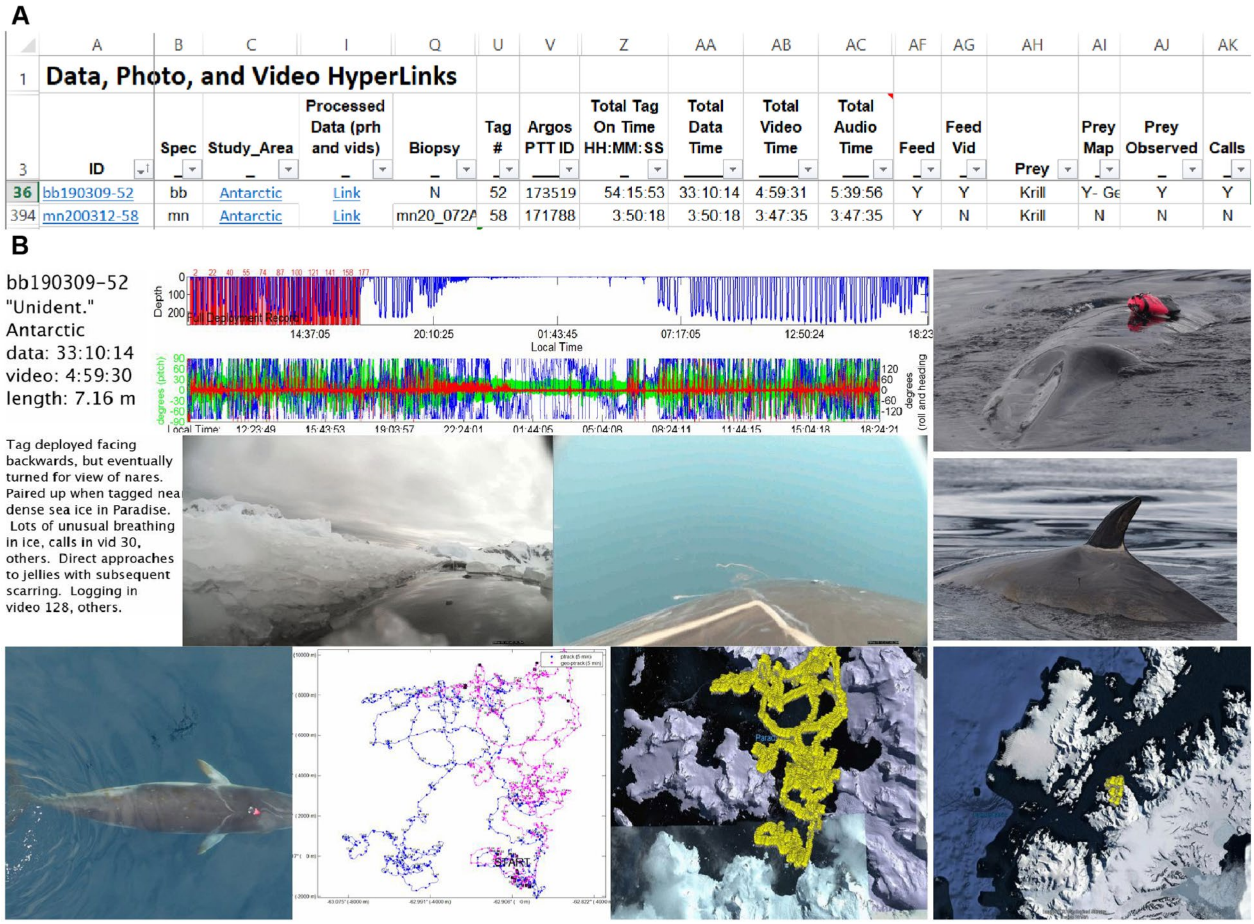
smoothed track and accelerometer data that can be read by Trackplot [22] (Fig. 11).

The final cell, 14, of **MainCATSprhTool.m** creates a "quicklook.jpg" file (Fig. 12) that allows for general information about the deployment to be examined at a glance, and individual deployments to be compared. As bio-logging data become more available and studies with large sample sizes become more feasible, quickly differentiating deployments can be critical. The quicklook takes information from what we refer to as a tag guide (Fig. 12B) that lists all metadata in one place. Tag guide information is integrated with information from the prh file as well as surface imagery and video imagery to create the overall snapshot.

After the prh file is completed, the data can be stitched onto the video (using the script **StitchDataonVideo.m** as discussed in part IIIa above) so that all data streams may be visualized simultaneously (Fig. 6, Additional file 2: Video S2). The script **renameVids.m** takes the stitched together clips with the video and data on them, now renamed with the deployment ID and video number, and appends on to the filename a timestamp of the video start for ease in searching for the right video.

Part IV—applications

Variations of the CATS tools processing scripts described herein have been utilized in at least 36 studies to date (full citation list: https://catsworkshop.sites.stanford.edu/citing-literature) in fields ranging from biomechanics to ecology to physiology. Many studies involve tag data from multiple tag types with varying sensors and sampling resolutions (e.g., [53, 54]), so our tools create several convergence points at which these data can be compared and analyzed in parallel. At the first convergence point, in the "other tag tools"



A

	A	B	C	I	Q	U	V	Z	AA	AB	AC	AF	AG	AH	AI	AJ	AK
1	Data, Photo, and Video HyperLinks																
3	ID	Spec	Study_Area	Processed Data (prh and vids)	Biopsy	Tag #	Argos PTT ID	Total Tag On Time HH:MM:SS	Total Data Time	Total Video Time	Total Audio Time	Feed	Feed Vid	Prey	Prey Map	Prey Observed	Calls
36	bb190309-52	bb	Antarctic	Link	N	52	173519	54:15:53	33:10:14	4:59:31	5:39:56	Y	Y	Krill	Y- Ge	Y	Y
394	mn200312-58	mn	Antarctic	Link	mn20_072A	58	171788	3:50:18	3:50:18	3:47:35	3:47:35	Y	N	Krill	N	N	N

B

bb190309-52
"Unident."
Antarctic
data: 33:10:14
video: 4:59:30
length: 7.16 m

Tag deployed facing backwards, but eventually turned for view of nares. Paired up when tagged nea dense sea ice in Paradise. Lots of unusual breathing in ice, calls in vid 30, others. Direct approaches to jellies with subsequent scarring. Logging in video 128, others.

Fig. 12 Storing metadata—the quicklook and TAG GUIDE. **A** TAG GUIDE is an excel file that stores metadata about all deployments for easy sorting of projects and deployment types and includes links to files and folders for easy access. **B** Quicklook file created for each deployment as the last step of **MainCATSprhTool.m** allows for visual identification of critical research elements of each deployment. The file creation step pulls information from the TAG GUIDE, the finished prh file, image files created as part of the prh file creation process (depth and prh graphs, as well *geoPtrack* plots) and user supplied image files (tag video stills, ID photos, overhead image, GoogleEarth images). Red numbers across the time-depth profile indicate the start times of video files with the corresponding numbers. Overhead image © Duke Marine Robotics and Remote Sensing. Whale length calculation described in Kahane-Rappport et al. [52]

folder, we include scripts to import raw tag data into our workflow for Acousondes [30], Wildlife Computers' TDR10 and Splash tags [25], Loggerhead Instruments' openTags [32] as well as CATS data, and users can adapt their own import scripts to convert raw tag data into the format used in the CATS workflow (see Part I). The CATS workflow in subsequent steps is modular, allowing a user to utilize different portions to accommodate tags with different combinations of sensors and varying analytical needs. The resulting prh, netCDF, Trackplot and "txt" files for all tag types contain the same structures, variables and naming conventions, facilitating downstream workflow and integration with tag data processed via alternate tag tools (e.g., the DTAG workflow, [7]. The netCDF file can also be easily imported into python or R using a suite of tools available at https://github.com/stacyderuiter/TagTools/tree/master/Python/tagiofuns or https://github.com/FlukeAndFeather/catsr, [55] respectively.

After the prh file is completed, we also provide a number of tools for utilizing animal orientation, motion, location. The scripts described below can be found in the "Applications" folder within the CATSMatlabTools folder.

Visualizing tag data

The prh file is designed to be an interchangeable package across tag platforms, so a user could use the same scripts and tools to process, plot and visualize data. We include two plotting scripts, with detailed comments for beginning users, of how different data streams can be time synched and examined using the built-in MATLAB zoom functions. **SimpleDataVisualization.m** plots depth, speed, pitch, roll, heading, jerk, and gyroscope data from a prh file, while **plot_overview.m** is a more flexible script that loads either a netCDF or prh file and utilizes stored metadata to additionally plot daylight/nighttime hours, detected events (see below), and the georeferenced pseudotrack with associated land contours and ice conditions (if relevant).

To import and visualize prh data in R [56], we also provide a "catsr" package, stored separately from the MATLAB tools at: https://github.com/FlukeAndFeather/catsr [55]. R is among the most widely used programming languages in ecological research and, as a free and open source language, it contributes to the growth of open and reproducible science [57]. The functions available in "catsr" facilitate cross-platform analyses and allow R users to interact with prh data. The R script "read_nc()" reads a prh file into memory from a netCDF file, while "view_cats()" produces an interactive plot for viewing multivariate time series [58]. For example, to plot depth, pitch and roll of the example data, a user could type "view_cats(mn200312_58, c("p", "pitch", "roll"))", or the "view_cats_3d()" script can also be used to render the georeferenced pseudotrack in three dimensions in an interactive plot. For additional details, we have included a help page within each function (e.g., enter "?read_nc" at the R console).

Accelerometer call detection

Some users might have an interest in using tag data to study bioacoustics and communication in their study species. While such users likely use existing software—e.g., Raven (Center for Conservation Bioacoustics, 2014), Triton [59], etc.—to analyze acoustic data like that collected by CATS and other tags, there can also be benefits to exploring low-frequency sound production through the lens of the triaxial accelerometer data from these tags. For example, previous studies [29, 60, 61, 62] have used the detection of vocalizations via high-frequency accelerometers to distinguish between vocalizations produced by a tagged individual from those produced by nearby conspecifics, a useful distinction for a range of behavioral and physiological research questions. The provided script, **accwav.m**, can be used to filter and save triaxial accelerometer data in audio format (.wav) for subsequent analysis. The resulting ".wav" file can be further analyzed using the aforementioned bioacoustics software options, but can also be manually audited using **accwav_audit.m**. This interactive function allows for visual inspection of the triaxial accelerometer data in spectrogram format, and saves manually detected vocalizations (and their associated time, depth, and sample index, determined via synchronization with the prh file) as a ".mat" data file. It should be noted that currently, accelerometers rarely record more rapidly than 1 kHz, meaning only the frequency components of vocalizations below 500 Hz (the Nyquist frequency, see [63] could be detected. For higher frequency vocalizations, such as those produced by odontocetes, other analytical techniques should be utilized (e.g., estimating the angle of arrival of the vocalization using two hydrophones, [2, 64]).

Fluke stroke/tailbeat detection

For animals that use the oscillatory movement of body structures to traverse through their environments (e.g., flapping wings, beating tail), the frequency of oscillations can impact the economy of transit and overall energetic performance [65–68]. Previous methods for calculating oscillatory frequency such as the http://www.animaltags.org/ script **dsf.m** use a Fast Fourier transform algorithm, which converts an orientation signal into the frequency domain with peaks for common oscillatory signals. This method works well for determining the dominant stroking frequency of a data signal, but does not allow for the fine-scale alignment of multiple data streams. A second method, developed by Martín López et al. [69], uses a **stroke_glide.m** script to determine periods of active stroking and gliding using zero-crossings. We provide here a similar method, **TailbeatDetect.m**, that utilizes zero-crossings of the sensor signals to determine the location of individual oscillations (one full wave cycle—upstroke and downstroke) within a data signal. This script uses a series of thresholds that can be modulated to include a greater or lesser number of individual oscillations, based upon the requirements of subsequent analyses (specific threshold suggestions available within the script). If there are time-synced depth or speed data streams available, we can align these data streams and determine the depth and speed during periods of active oscillatory movement. The output from this method is a series of individual oscillations with the start and end signal positions, the period (oscillatory frequency = 1/period), and the mean depth and mean speed (measured from the start signal position to the end signal position) if those data streams are available.

Event detection

For marine species that hunt at depth, behavior is often unobservable to researchers on the surface. However, high sample rate bio-loggers enable biologically and ecologically significant events to be quantitatively described. Event detection scripts can be used to extract specific behavioral information from the broader prh file, providing a user-friendly means to analyze the prh data, and allowing analysts to bring together prh files from different projects and species. As an example, one event of particular interest for rorqual whales (family: Balaenopteridae) is a feeding lunge. Lunge filter feeding is a two-step process, consisting of a high-speed engulfment of prey-laden water followed by a period of filtration through baleen plates [70]. A lunge feeding event can be identified in the record by finding a period of acceleration, often associated with fluking, leading to a distinct speed maximum followed by a rapid deceleration with some continued forward momentum and a gap in fluking during the filtration process [52]. Procedures to automatically detect lunge feeding events from prh files have been described in the literature [71–73]; however, for increased precision, we recommend manual examination of the data streams and have included an auditing script, **SimpleLungeDetection.m**, which creates a "*_lunge.mat" file with stored indices (*LungeI*) and times (*LungeDN*) that can be used to identify the feeding events for later analysis, and could additionally be adapted for other types of events. This script also allows the user to record their confidence in the identification for lunges that appear to be non-stereotypical.

While lunge feeding is ubiquitous among the rorqual whales, the animal's kinematics surrounding the lunge event can exhibit high variability [71, 14, 72, 73, 74]. The scripts **StrategyandLungeDetection.m** and **SimpleLungeDetectionBehaviorState.m** facilitate analysis of these events by allowing a user to mark start and end periods of these variable strategies in addition to identifying a single point in time that represents the lunge. These time periods can then be used to analyze each behavioral period discretely (e.g., duration of prey approach or distance traveled within a bubble net). A specific period of interest for rorqual whales is the duration of filtering time after a lunge when engulfed water must be filtered through a set of baleen plates attached to a rorqual whale's upper jaw. The filtration period can be visualized as the period of gliding following the rapid deceleration during buccal cavity inflation (see Fig. 2 in [52]. **Filtration.m** steps a user through an existing lunge file and allows additional auditing of the kinematic signature of filtration.

Discussion

As discussed by Boyd et al. [75], the discipline of bio-logging, and modern science in general, is "dominated by instruments that churn out data," leading to the temptation to sacrifice hypothesis-testing-based investigation in favor of the collection of an ever increasing amount of data. This temptation can be especially acute when the pace of technological development is faster than the pace of rigorous scientific testing, validation and reporting. The code accompanying this guide, for instance, has undergone nearly constant iterations and innovations since its first applications to CATS tags in the summer of 2014 as video, audio and sensor data have improved and changed. Yet, as of this writing, many customized CATS tags include 4 k resolution video to increase utility for outreach, but our tools are not yet enabled to process these data efficiently. Additionally, with the increasing pace of data acquisition generally, the standardization of approaches to processing and sharing bio-logging data will be a critical facet of the big data era of bio-logging [17, 76].

The purpose of this manuscript, then, is threefold: 1) to familiarize novice users with the procedures, variables and potential pitfalls of accelerometer data so that researchers can form informed hypotheses with better knowledge of the kinds of data they can expect to test (i.e., reduce the "black box" aspect of bio-logging devices); 2) to provide a forum through the GitHub repository for continued development of tools that meet the current needs of the data; 3) to supplement and expand the standardization of accelerometer data advocated by prior researchers (e.g., http://www.animaltags.org/; [20, 7]).

To accomplish these goals, all accompanying code has been published open source. As much as possible, code was written with best scientific computing practices in mind [77], but at times multiple iterations of code, differing MATLAB platform version requirements, adjustments to make the software more adaptable to potential data issues or alternate tag types, or time limitations mean that some parts of the provided code are not yet as clean as they could be. We hope that by providing this start, we can harness some of the strengths of platforms such as GitHub to make it "easier to grow pools of participants" [78] and facilitate group development of future tools.

To help grow the user base, in December 2020 we hosted a virtual workshop to train new users and get feedback in implementing the described methodology. Attendance ranged from 52 to 77 unique participants over the 5 days (mean 62.6). Course materials are all available online (https://catsworkshop.sites.stanford.edu/), and registration fees were by donation to raise funds to provide paid internship opportunities at the Hopkins Marine Station for Monterey Bay area high school students. Overall reception of the workshop was positive, with 27 out of 30 survey respondents agreeing or strongly agreeing that they are "likely to use the workshop CATS tools in [their] future work". Voluntary pre- and post-assessments were also given, with results summarized in Table 1.

Table 1 Pre- and post-self-assessments of skills given in the Dec. 2020 workshop presenting this methodology

Phrase	Pre-assessment % agreement		Post-assessment % agreement	
	58 total respondents (%)	28 paired respondents (%)	31 total respondents (%)	28 paired respondents (%)
I know how to examine bio-logging data in MATLAB	29	**36**	87	**86**
I can plot time series in MATLAB	38	**43**	77	**75**
I could teach a new user how to operate a CATS tag (setup for deployment and download data)	21	**18**	42	**43**
I currently use MATLAB regularly as part of my job	38	**43**	45	**43**
I plan to use MATLAB more often as part of my job	74	**75**	90	**89**
I understand how accelerometers and magnetometers interact to determine animal orientation	60	**57**	94	**93**
I understand how CATS data is synchronized with video data	14	**11**	90	**93**
I understand in principle how speed can be estimated from high frequency accelerometer data or flow noise	50	**46**	90	**93**
I understand how to convert data collected in situ into engineering units	21	**18**	55	**57**
I have the tools I need to create synched video and data files from CATS data	24	**21**	90	**93**

% agreement is the percentage of respondents who answered "yes" they agree with the statement. Paired respondents are those who answered both a pre- and a post-survey

Known potential issues

Animal bio-logging involves attaching sensitive electronics in as small a form as possible to an animal that generally does not want it attached in often inhospitable environments (e.g., pressurized saltwater), often in ways that are impossible to test except in the field in real research situations. As such, there are a variety of errors that can arise in the presented workflow. Our scripts endeavor to account for potential errors, but in their attempt to be robust to known issues, they may be more susceptible and harder to fix when unforeseen errors arise. To conclude our discussion, we wish to point out several places, where known errors may arise in processing tag data.

Time synchronization is a critical aspect of integrating multiple data streams into a usable package, yet errors in missed data points or clock drift arise commonly. While some tag types operate off a single clock and have integrated error checking into data writing and/or extraction, others will record a single tag on time and assume all collected data is sequential with no gaps. CATS tags, among others, record a timestamp on each collected data point (both the inertial sensor data and video) as a check for any processing issues. We maintain this design choice in the *data* and *DN* variables, using MATLAB date numbers (which are days and partial days since the start of year 0) to keep track of time and look for any missing data points. The advantage to this is accuracy and ease of converting any section of data into local time, but one disadvantage is that sub-millisecond precision is not possible using standard double-precision MATLAB values.

The use of video cameras also introduces a bevy of potential issues. While memory storage is rarely a limiting factor, downloading large files as well as battery and processing power can be. Current CATS tags have a port-free, wireless connection design to minimize the risk of water intrusion. However, wireless downloading is less stable than cabled connections, and when multiple tags are communicating with multiple computers simultaneously, it can lead to signal interference and interrupted downloads that must be restarted. Additionally, if internal processors are not fast enough, problems related to skipped frames, bad data reads, and delays between start triggers and the start of recording can accumulate rapidly in the challenging light environments of typical tag deployments (Additional file 1: Video S1). Bad video reads can additionally create bad audio reads if video data are tied to audio data as in older versions of CATS tags. Much of the processing and code in cell 1 of **MainCATSprhTool.m** relates to looking for and correcting for any skipped video and audio data. The larger battery requirements for video are also not just a question of storage, but also power draw. When cameras are on and running, the increased power draw appears to create its own small magnetic field that affects the magnetometers (thus the separate "cam on" and "cam off" calibration steps) and also interferes with GPS acquisition. These are engineering issues for which current solutions are in process (and current CATS tags have generally solved the magnetometer issue), but they lead to issues with downstream processing.

If GPS points cannot be consistently acquired (whether through interference of cameras, poor resolution of acquired points, low placement on an animal, or a lack of a GPS sensor entirely), integrated dead-reckoning tracks can accumulate error rapidly [2, 79, 80, 37] as any small errors in heading, speed or pitch are integrated multiple times per second, and any motion of the animal that is not forward (e.g., drift from wind or currents) cannot be accounted for. Particularly challenging is that the speed from flow noise and tag jiggle methods described herein have no resolution below ~ 1 m/s [18, 81] and are susceptible to spikes and stalls when the tag breaks the air–water interface. As a result, slowly moving or logging animals [82], which can rest for an hour or more, are assumed by the process to instead be making slow forward progress. The implementation of an orientation-independent, high-dynamic-range, field-accurate speed sensor that can be integrated into multiple bio-logging packages is a high-priority engineering challenge (though see sensors described in [39, 35, 40, 41, 42, 43].

Once data is downloaded, software compatibility can prove to be challenging when trying to craft a standard procedure. All packages presented herein have been tested on Windows systems, but Macintosh or Linux users may find places, where unforeseen issues arise, including dialog boxes that do not appear to display instructions correctly, as well as file path errors that may need to be corrected by switching the direction of the "\".[2] Additionally, the primary tag tool platform we use, MATLAB, is not freeware, so initial processing steps may be challenging for researchers without access to the platform. Some freeware programs, such as Octave (https://www.gnu.org/software/octave/), are compatible with many of our scripts, though a thorough compatibility test has not been completed. For short-term projects, student and temporary licenses may also be available at low cost (www.mathworks.com), but for longer term the need for a MATLAB license adds additional costs to tag purchase and processing. Additionally, MATLAB typically provides semi-annual updates to its base platform. While typically new versions are backwards compatible, there can be small challenges that arise that frustrate users. As one minor example, all figures included herein were made in MATLAB 2014a, but if a user runs code on newer versions, they will find that the default color patterns are changed. Additionally, built-in functions, such as **readtable.m** and **var.m,** have had updates to their input parameters in recent versions. The changes lead to better code, but older code must be constantly updated to ensure both backwards and forwards compatibility.

Finally, with better battery life and attachment methods, some tags have started collecting multi-day data sets (e.g., [25]). Our modular workflow allows for "lazy loading" of only essential variables to minimize the required RAM for any set of calculations. However, a user may still find that their data sets are difficult to process using personal computers. Having too much data could be classified as "a good problem to have," but can still be frustrating. We recommend dividing data up into manageable chunks, perhaps processing 1 day at a time. As this problem is likely deployment specific, we do not currently include tools to easily correct for this issue. As data sets continue to grow, however, we anticipate including code to facilitate processing of multi-day data sets in future versions of these tools. Please be sure to check the GitHub page for the most current version of any tools before beginning the process we describe of integrating video with an estimation of animal orientation and motion from inertial sensors.

Conclusions

The most important attitude that can be formed is that of desire to go on learning. -John Dewey.

Abbreviations

CATS: Customized animal tracking solutions; CSV: Comma separated value file type; GPS: Global positioning system; MEMS: Micro-electromechanical systems (a type of accelerometer/magnetometer/gyroscope package); MSA: Minimum specific acceleration; NaN: Not-a-number (a MATLAB placeholder variable); NED: North-east-down; NEU: North-east-up; OCDR: Orientation-corrected depth rate; ODBA: Overall dynamic body acceleration; prh: Pitch-roll-heading (an abbreviation for the common file type holding all tag variables, see "CATSVarNames.txt" in the main CATSMatlabTools folder).

Variables

Adata: Raw (uncalibrated) accelerometer data (at original sampling rate and the tag's axis conventions); *At*: Calibrated tag-frame accelerometer data (relative to gravity in NED orientation); *Atime*: Matlab date numbers corresponding to each timestep of the accelerometer; *Aw*: Calibrated animal-frame accelerometer data (relative to gravity in NED orientation); *axA*: Rotation matrix to right multiply accelerometer matrix to convert the tag's axis conventions to NED orientation; *axAo*: Original axes of tag data (see Eq. 1); *data*: Data table with all tag data, one row corresponds to one time step. All variables are up or downsampled as appropriate.; *Gt*: Calibrated tag-frame gyroscope data (in radians/s in NED orientation); *Gw*: Calibrated animal-frame gyroscope data (in radians/s in NED orientation); *Mt*: Calibrated tag-frame compass/magnetometer data (in μT in NED orientation); *Mw*: Calibrated animal-frame compass/magnetometer data (in μT in NED orientation); *vidDN*: Matlab date numbers of the start time of each video and audio file; *vidDurs*: Duration, in seconds, of each video or audio file (this format assumes tag versions that write audio files directly onto or between video files). See also: **stitchaudio.m**.

[2] Future iterations of the tag tools we provide are expected to implement the built-in MATLAB functions **ismac.m** and **ispc.m**, as well as search algorithms to automatically change slash directions for the appropriate platform.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s40317-021-00256-w.

Additional file 1: Video S1. A rapidly changing light environment can challenge the on-board processor.

Additional file 2: Video S2. A finished video synchronized with sensor data highlighting a few scenarios in which underwater video can provide insights not available with accelerometer data alone.

Additional file 3: Video S3. CATS tags have come in a variety of camera arrangements. Here we highlight the processed format of a tag with a 360° lens and a tag with dual forward and rear facing lenses.

Acknowledgements

Thanks to Mark Johnson and Stacy DeRuiter and others who have worked on DTAG tools for setting the stage for this work over the years, including designing the Animal Tags Project at http://www.animaltags.org/, and running a workshop in October 2017 on utilizing tag tools attended by several authors. Thanks also to Nikolai Liebsch and Peter Kraft at CATS for their continuous work to improve the devices that make hypothesis-driven bio-logging possible. Thanks to Jim Harvey, Alison Stimpert and Moss Landing Marine Labs for supporting and participating in field efforts in Monterey Bay, and to Kakani Katija and MBARI for use of their pressure chamber. Thanks to Jessica Bender for the illustrations used in Figs. 4 and 7. Thanks also to the workshop participants in Dec 2020 who gave a week of their time to learn these tools and along the way provided valuable feedback on our code and instructional practices. This group includes Taylor Azizeh, Ellen Chenoweth, Leah Crowe, Jacopo Di Clemente, Julia Dombroski, Arina Favilla, Elise Keppel, Jessica Kendall-Bar, Theresa Kirchner, Jessica Kittel, Marc Lammers, Sarah Luongo, Morgan Martin, Raphael Mayaud, Alexandra McInturf, Christie McMillan, Cameron Perry, Nicola Quick, Rhonda Reidy, Kerri Seger, Anna Selbmann, Jeanne Shearer, Andy Szabo, Jenn Tackaberry, Emma Vogel, Mason Weinrich, Suzie Winquist, Eden Zang, and Julia Zeh.

Authors' contributions

DEC wrote the primary code not otherwise credited and organized and designed the workflow. DEC & WTG organized and facilitated the CATS workshop. WTG designed and wrote the workshop webpage and GitHub wiki. WTG, MFC, JAF, SRKR, JMJL, RCN, WKO, DMW contributed "Applications" code and text and tested the code workflow. ASF and JAG conceptualized and supervised the project and procured funding. DEC wrote the manuscript. All authors read and approved the final manuscript.

Funding

This work funded with NSF Grants IOS-1656691 and OPP-1643877, ONR YIP Grant #N000141612477, Grants from the World Wildlife Fund, and Stanford University's Terman and Bass Fellowships. Funds raised from workshop registration were used to support a paid high-school internship program at Stanford University and included a grant from the National Marine Sanctuary Foundation and individual donations from workshop participants.

Availability of data and materials

Workshop and tutorial materials with detailed instructions for materials use, including example exercises and walk-throughs are available at https://catsworkshop.sites.stanford.edu/. This workshop home page contains direct links to the MATLAB code repository at https://GitHub.com/wgough/CATS-Methods-Materials, which contains a step-by-step wiki, and also to a dryad depository (https://datadryad.org/stash/share/KFi8G5QC7DFPYXynQeSotxtXqANZL70LFUGEiiDTSMU) with example data for a user to practice with.

Declarations

Ethics approval and consent to participate

All research was conducted under institutional IACUC protocols, as well as NMFS Permits 14809, 16111, 20430, 21678, 23095 and ACA permit 2015–011.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Institute of Marine Science, University of California, Santa Cruz, 115 McAllister Way, Santa Cruz, CA 95060, USA. [2] Hopkins Marine Station, Stanford University, 120 Ocean View Blvd, Pacific Grove, CA 93950, USA. [3] Cascadia Research Collective, Olympia, WA, USA. [4] Cetacean Ecology and Acoustics Laboratories, University of Queensland, 37 Fraser Street, Dunwich, Queensland, Australia. [5] Present Address: Sound Communication and Behaviour Group, Department of Biology, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark.

Received: 19 May 2021 Accepted: 21 July 2021
Published online: 26 August 2021

References

1. Goldbogen JA, Friedlaender AS, Calambokidis J, McKenna MF, Simon M, Nowacek DP. Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. Bioscience. 2013;63:90–100.
2. Johnson M, de Soto NA, Madsen PT. Studying the behaviour and sensory ecology of marine mammals using acoustic recording tags: a review. Mar Ecol Prog Ser. 2009;395:55–73.
3. Weyer NM, Fuller A, Haw AJ, Meyer LCR, Mitchell D, Picker M, Rey B, Hetem RS. Increased diurnal activity is indicative of energy deficit in a nocturnal mammal, the aardvark. Front Physiol. 2020;11:637.
4. Miller LP, Dowd WW. Multimodal in situ datalogging quantifies inter-individual variation in thermal experience and persistent origin effects on gaping behavior among intertidal mussels (Mytilus californianus). J Exp Biol. 2017;220:4305–19.
5. Goldbogen JA, Meir JU. The device that revolutionized marine organismal biology. J Exp Biol. 2014;217:167–8.
6. Kooyman GL. An analysis of some behavioral and physiological characteristics related to diving in the Weddell seal. Antarct Res Ser. 1967;11:227–61.
7. Johnson MP, Tyack PL. A digital acoustic recording tag for measuring the response of wild marine mammals to sound. IEEE J Oceanic Eng. 2003;28:3–12.
8. Madsen P, Payne R, Kristiansen N, Wahlberg M, Kerr I, Møhl B. Sperm whale sound production studied with ultrasound time/depth-recording tags. J Exp Biol. 2002;205:1899–906.
9. von Benda-Beckmann AM, Wensveen PJ, Samara FI, Beerens SP, Miller PJ. Separating underwater ambient noise from flow noise recorded on stereo acoustic tags attached to marine mammals. J Exp Biol. 2016;219:2271–5.
10. McKnight JC, Bennett KA, Bronkhorst M, Russell DJ, Balfour S, Milne R, Bivins M, Moss SE, Colier W, Hall AJ. Shining new light on mammalian diving physiology using wearable near-infrared spectroscopy. PLoS Biol. 2019;17:e3000306.
11. Meir JU, Champagne CD, Costa DP, Williams CL, Ponganis PJ. Extreme hypoxemic tolerance and blood oxygen depletion in diving elephant seals. Am J Physiol Regulat Integr Comparat Physiol. 2009;297:R927–39.
12. Ponganis PJ, Meir JU, Williams CL. In pursuit of Irving and Scholander: a review of oxygen store management in seals and penguins. J Exp Biol. 2011;214:3325–39.
13. Williams TM, Noren SR, Glenn M. Extreme physiological adaptations as predictors of climate-change sensitivity in the narwhal, Monodon monoceros. Mar Mamm Sci. 2011;27:334–49.
14. Cade DE, Friedlaender AS, Calambokidis J, Goldbogen JA. Kinematic diversity in rorqual whale feeding mechanisms. Curr Biol. 2016;26:2617–24.
15. Goldbogen JA, Cade DE, Boersma AT, Calambokidis J, Kahane-Rapport SR, Segre PS, Stimpert AK, Friedlaender AS. Using digital tags with integrated video and inertial sensors to study moving morphology and associated function in large aquatic vertebrates. Anat Rec. 2017;300:1935–41.

16. Martín López LM, de Soto NA, Miller P, Johnson M. Tracking the kinematics of caudal-oscillatory swimming: a comparison of two on-animal sensing methods. J Exp Biol. 2016;219:2103–9.
17. Sequeira AM, O'Toole M, Keates TR, McDonnell LH, Braun CD, Hoenner X, Jaine FR, Jonsen ID, Newman P, Pye J, Bograd SJ, Hays GC, Hazen E, Holland M, Tsontos VM, Blight C, Cagnacci F, Davidson SC, Dettki H, Duarte CM, Dunn DC, Eguiluz VM, Fedak M, Gleiss AC, Hammerschlag N, Hindell M, Holland KN, Janekovic I, Mckinzie MK, Muelbert MMC, Pattiaratchi C, Rutz C, Sims DW, Simmons S, Townsend B, Whoriskey F, Woodward B, Costa DP, Heupel MR, McMahon CR, Harcourt R, Weise M. A standardisation framework for bio-logging data to advance ecological research and conservation. Methods Ecol Evol. 2021;9:884.
18. Cade DE, Barr KR, Calambokidis J, Friedlaender AS, Goldbogen JA. Determining forward speed from accelerometer jiggle in aquatic environments. J Exp Biol. 2018;221:jeb170449.
19. Goldbogen JA, Calambokidis J, Shadwick RE, Oleson EM, McDonald MA, Hildebrand JA. Kinematics of foraging dives and lunge-feeding in fin whales. J Exp Biol. 2006;209:1231–44.
20. Johnson M. Measuring the orientation and movement of marine animals using inertial and magnetic sensors-a tutorial. Fine-scale animal movement workshop. Australia: Hobart; 2011.
21. Simon M, Johnson M, Madsen PT. Keeping momentum with a mouthful of water: behavior and kinematics of humpback whale lunge feeding. J Exp Biol. 2012;215:3786–98.
22. Ware C, Arsenault R, Plumlee M, Wiley D. Visualizing the underwater behavior of humpback whales. IEEE Comput Graph Appl. 2006;54:14–8.
23. Blischak JD, Davenport ER, Wilson G. A quick introduction to version control with Git and GitHub. PLoS Comput Biol. 2016;12:e1004668.
24. Perkel J. Democratic databases: science on GitHub. Nat News. 2016;538:127.
25. Calambokidis J, Fahlbusch JA, Szesciorka AR, Southall BL, Cade DE, Friedlaender AS, Goldbogen JA. Differential vulnerability to ship strikes between day and night for blue, fin, and humpback whales based on dive and movement data from medium duration archival tags. Front Marine Sci. 2019;6:114.
26. Szesciorka AR, Calambokidis J, Harvey JT. Testing tag attachments to increase the attachment duration of archival tags on baleen whales. Anim Biotelemet. 2016;4:18.
27. Mikkelsen L, Johnson M, Wisniewska DM, van Neer A, Siebert U, Madsen PT, Teilmann J. Long-term sound and movement recording tags to study natural behavior and reaction to ship noise of seals. Ecol Evol. 2019;9:2588–601.
28. Hays GC. New insights: animal-borne cameras and accelerometers reveal the secret lives of cryptic species. J Anim Ecol. 2015;84:587–9.
29. Goldbogen JA, Stimpert AK, DeRuiter SL, Calambokidis J, Friedlaender AS, Schorr GS, Moretti DJ, Tyack PL, Southall BL. Using accelerometers to determine the calling behavior of tagged baleen whales. J Exp Biol. 2014;217:2449–55.
30. Burgess WC. The acousonde: a miniature autonomous wideband recorder. The Journal of the Acoustical Society of America. 2009;125:2588–2588.
31. Muramoto H, Ogawa M, Suzuki M, Naito Y. Little Leonardo digital data logger: its past, present and future role in bio-logging science. Memoirs of National Institute of Polar Research. Special Issue. 2004;58:196–202.
32. Cade DE, Levenson JJ, Cooper B, de la Parra R, Webb DH, Dove A. Whale sharks increase swimming effort while filter feeding, but appear to maintain high foraging efficiencies. J Exp Biol. 2020;223:jeb224402.
33. Gleiss AC, Wilson RP, Shepard EL. Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure. Methods Ecol Evol. 2011;2:23–33.
34. Wilson RP, White CR, Quintana F, Halsey LG, Liebsch N, Martin GR, Butler PJ. Moving towards acceleration for estimates of activity-specific metabolic rate in free-living animals: the case of the cormorant. J Anim Ecol. 2006;75:1081–90.
35. Fletcher S, Le Boeuf BJ, Costa DP, Tyack PL, Blackwell SB. Onboard acoustic recording from diving northern elephant seals. J Acoust Soc Am. 1996;100:2531–9.
36. Miller PJ, Johnson MP, Tyack PL, Terray EA. Swimming gaits, passive drag and buoyancy of diving sperm whales *Physeter macrocephalus*. J Exp Biol. 2004;207:1953–67.
37. Wilson RP, Liebsch N, Davies IM, Quintana F, Weimerskirch H, Storch S, Lucke K, Siebert U, Zankl S, Müller G. All at sea with animal tracks; methodological and analytical solutions for the resolution of movement. Deep Sea Res Part II. 2007;54:193–210.
38. Ydesen KS, Wisniewska DM, Hansen JD, Beedholm K, Johnson M, Madsen PT. What a jerk: prey engulfment revealed by high-rate, supercranial accelerometry on a harbour seal (*Phoca vitulina*). J Exp Biol. 2014;217:2239–43.
39. Aoki K, Amano M, Mori K, Kourogi A, Kubodera T, Miyazaki N. Active hunting by deep-diving sperm whales: 3D dive profiles and maneuvers during bursts of speed. Mar Ecol Prog Ser. 2012;444:289–301.
40. Kawatsu S, Sato K, Watanabe Y, Hyodo S, Breves JP, Fox BK, Grau EG, Miyazaki N. A new method to calibrate attachment angles of data loggers in swimming sharks. EURASIP J Adv Signal Process. 2009;2010:732586.
41. Sato K, Mitani Y, Cameron MF, Siniff DB, Naito Y. Factors affecting stroking patterns and body angle in diving Weddell seals under natural conditions. J Exp Biol. 2003;206:1461–70.
42. Shepard EL, Wilson RP, Liebsch N, Quintana F, Laich AG, Lucke K. Flexible paddle sheds new light on speed: a novel method for the remote measurement of swim speed in aquatic animals. Endang Species Res. 2008;4:157–64.
43. Wilson R, Achleitner K. A distance meter for large swimming marine animals. S Afr J Mar Sci. 1985;3:191–5.
44. Burgess WC, Tyack PL, Le Boeuf BJ, Costa DP. A programmable acoustic recording tag and first results from free-ranging northern elephant seals. Deep Sea Res Part II. 1998;45:1327–51.
45. Finger RA, Abbagnaro LA, Bauer BB. Measurements of low-velocity flow noise on pressure and pressure gradient hydrophones. J Acoust Soc Am. 1979;65:1407–12.
46. Altmann J. Observational study of behavior: sampling methods. Behaviour. 1974;49:227–66.
47. Southall BL, Nowacek DP, Miller PJ, Tyack PL. Experimental field studies to measure behavioral responses of cetaceans to sonar. Endangered Species Research. 2016;31:293–315.
48. Irvine L, Palacios DM, Urbán J, Mate B. Sperm whale dive behavior characteristics derived from intermediate-duration archival tag data. Ecol Evol. 2017;7:7822–37.
49. Owen K, Jenner KCS, Jenner M-NM, McCauley RD, Andrews RD. Water temperature correlates with baleen whale foraging behaviour at multiple scales in the Antarctic. Mar Freshw Res. 2019;70:19–32.
50. Linsky JMJ, Wilson N, Cade DE, Goldbogen JA, Johnston DW, Friedlaender AS. The scale of the whale: using video-tag data to evaluate sea-surface ice concentration from the perspective of individual Antarctic minke whales. Anim Biotelemet. 2020;8:1–12.
51. Tackaberry JE, Cade DE, Goldbogen JA, Wiley DN, Friedlaender AS, Stimpert AK. From a calf's perspective: humpback whale nursing behavior on two US feeding grounds. Peer J. 2020;8:e8538.
52. Kahane-Rapport SR, Savoca MS, Cade DE, Segre PS, Bierlich KC, Calambokidis J, Dale J, Fahlbusch JA, Friedlaender AS, Johnston D, Werth AJ, Goldbogen JA. Lunge filter feeding biomechanics constrain rorqual foraging ecology across scale. J Exp Biol. 2020;15:jeb224196.
53. Goldbogen JA, Cade DE, Wisniewska DM, Potvin J, Segre PS, Savoca MS, Hazen EL, Czapanskiy MF, Kahane-Rapport SR, DeRuiter SL, Gero S, Tønnesen P, Gough WT, Hanson MB, Holt M, Jensen FH, Simon M, Stimpert AK, Arranz P, Johnston DW, Nowacek DP, Parks SE, Visser F, Friedlaender AS, Tyack PL, Madsen PT, Pyenson ND. Why whales are big but not bigger: physiological drivers and ecological limits in the age of ocean giants. Science. 2019;366:1367–72.
54. Segre PS, Potvin J, Cade DE, Calambokidis J, Di Clemente J, Fish FE, Friedlaender AS, Gough WT, Kahane-Rapport SR, Oliveira C. Energetic and physical limitations on the breaching performance of large whales. Elife. 2020;9:e51760.
55. Czapanskiy MF (2021) FlukeAndFeather/catsr: catsr: R package for visualizing CATS PRH files. v1.0.0 Edition.https://doi.org/10.5281/zenodo.5140485.
56. R Core Team. R: a language and environment for statistical computing. Vienna: Austria; 2021.
57. Lai J, Lortie CJ, Muenchen RA, Yang J, Ma K. Evaluating the popularity of R in ecology. Ecosphere. 2019;10:e02567.
58. Sievert C. Interactive web-based data visualization with R, plotly, and shiny. New York: Chapman and Hall; 2020.

59. Wiggins S. Autonomous acoustic recording packages (ARPs) for long-term monitoring of whale sounds. Mar Technol Soc J. 2003;37:13–22.
60. Oestreich WK, Fahlbusch JA, Cade DE, Calambokidis J, Margolina T, Joseph J, Friedlaender AS, McKenna MF, Stimpert AK, Southall BL, Goldbogen JA, Ryan JP. Animal-borne metrics enable acoustic detection of blue whale migration. Curr Biol. 2020;30:1–7.
61. Saddler MR, Bocconcelli A, Hickmott LS, Chiang G, Landea-Briones R, Bahamonde PA, Howes G, Segre PS, Sayigh LS. Characterizing Chilean blue whale vocalizations with DTAGs: a test of using tag accelerometers for caller identification. J Exp Biol. 2017;220:4119–29.
62. Stimpert AK, DeRuiter SL, Falcone EA, Joseph J, Douglas AB, Moretti DJ, Friedlaender AS, Calambokidis J, Gailey G, Tyack PL, Goldbogen JA. Sound production and associated behavior of tagged fin whales (Balaenoptera physalus) in the Southern California Bight. Anim Biotelemet. 2015;3:1.
63. Au WW, Hastings MC. Principles of marine bioacoustics. Springer; 2008.
64. Pérez JM, Jensen FH, Rojano-Doñate L, Aguilar de Soto N. Different modes of acoustic communication in deep-diving short-finned pilot whales (Globicephala macrorhynchus). Mar Mamm Sci. 2017;33:59–79.
65. Fish FE. Comparative kinematics and hydrodynamics of odontocete cetaceans: morphological and ecological correlates with swimming performance. J Exp Biol. 1998;201:2867–77.
66. Gough WT, Segre PS, Bierlich K, Cade DE, Potvin J, Fish FE, Dale J, di Clemente J, Friedlaender AS, Johnston DW, Kahane-Rapport SR, Kennedy J, Long J, Oudejans M, Penry GS, Savoca MS, Simon M, Videsen S, Visser F, Wiley D, Goldbogen JA. Scaling of swimming performance in baleen whales. J Exp Biol. 2019;222:jeb204172.
67. Sato K, Shiomi K, Watanabe Y, Watanuki Y, Takahashi A, Ponganis PJ. Scaling of swim speed and stroke frequency in geometrically similar penguins: they swim optimally to minimize cost of transport. Proc R Soc B. 2010;277:707–14.
68. Sato K, Watanuki Y, Takahashi A, Miller PJ, Tanaka H, Kawabe R, Ponganis PJ, Handrich Y, Akamatsu T, Watanabe Y. Stroke frequency, but not swimming speed, is related to body size in free-ranging seabirds, pinnipeds and cetaceans. Proc R Soc Lond B. 2007;274:471–7.
69. Martín López LM, Miller PJ, de Soto NA, Johnson M. Gait switches in deep-diving beaked whales: biomechanical strategies for long-duration dives. J Exp Biol. 2015;218:1325–38.
70. Goldbogen JA, Cade DE, Calambokidis J, Friedlaender AS, Potvin J, Segre PS, Werth AJ. How baleen whales feed: the biomechanics of engulfment and filtration. Ann Rev Mar Sci. 2017;9:1–20.
71. Allen AN, Goldbogen JA, Friedlaender AS, Calambokidis J. Development of an automated method of detecting stereotyped feeding events in multisensor data from tagged rorqual whales. Ecol Evol. 2016;6:7522–35.
72. Kot BW, Sears R, Zbinden D, Borda E, Gordon MS. Rorqual whale (Balaenopteridae) surface lunge-feeding behaviors: Standardized classification, repertoire diversity, and evolutionary analyses. Mar Mamm Sci. 2014;30:1335–57.
73. Owen K, Dunlop RA, Monty JP, Chung D, Noad MJ, Donnelly D, Goldizen AW, Mackenzie T. Detecting surface-feeding behavior by rorqual whales in accelerometer data. Mar Mamm Sci. 2016;32:327–48.
74. Wiley DN, Ware C, Bocconcelli A, Cholewiak D, Friedlaender A, Thompson M, Weinrich M. Underwater components of humpback whale bubble-net feeding behavior. Behaviour. 2011;148:575–602.
75. Boyd IL, Kato A, Ropert-Coudert Y. Bio-logging science: sensing beyond the boundaries. Berlin: Springer; 2004.
76. Yoda K. Advances in bio-logging techniques and their application to study navigation in wild seabirds. Adv Robot. 2019;33:108–17.
77. Balaban G, Grytten I, Rand KD, Scheffer L, Sandve GK. Ten simple rules for quick and dirty scientific programming. CA USA: Public Library of Science San Francisco; 2021.
78. McDonald N, Goggins S. Performance and participation in open source software on github. CHI'13 extended abstracts on human factors in computing systems. Berlin: Springer; 2013. p. 139–44.
79. Wensveen PJ, Thomas L, Miller PJ. A path reconstruction method integrating dead-reckoning and position fixes applied to humpback whales. Mov Ecol. 2015;3:1–16.
80. Wilson R, Wilson M-P. Dead reckoning: a new technique for determining penguim movements at sea. Meeresforschung. 1988;32:155–8.
81. Goldbogen JA, Pyenson ND, Shadwick RE. Big gulps require high drag for fin whale lunge feeding. Mar Ecol Prog Ser. 2007;349:289–301.
82. Iwata T, Biuw M, Aoki K, Miller PJOM, Sato K. Using an omnidirectional video logger to observe the underwater life of marine animals: humpback whale resting behaviour. Behav Process. 2021;104:369.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.