Received: 19 September 2019 | Accepted: 25 March 2020

DOI: 10.1111/2041-210X.13397

APPLICATION

Methods in Ecology and Evolution

AMMonitor: Remote monitoring of biodiversity in an adaptive framework with R

Cathleen Balantic[1] | Therese Donovan[2]

[1]Vermont Cooperative Fish and Wildlife Research Unit, University of Vermont, Burlington, VT, USA

[2]Vermont Cooperative Fish and Wildlife Research Unit, U.S. Geological Survey, Rubenstein School of Environment and Natural Resources, University of Vermont, Burlington, VT, USA

Correspondence
Therese Donovan
Email: tdonovan@uvm.edu

Funding information
National Park Service; U.S. Bureau of Land Management; US Geological Survey; University of Vermont; Vermont Department of Fish and Wildlife; Wildlife Management Institute

Handling Editor: Sarah Goslee

Abstract

1. Ecological research and management programs are increasingly using autonomous monitoring units (AMUs) to collect large volumes of acoustic and/or photo data to address pressing management objectives or research goals. The data management requirements of an AMU-based monitoring effort are often overwhelming, with a considerable amount of processing to translate raw data into models and analyses that have research and management utility.
2. We created the R package AMMonitor to simplify the process of moving from remotely collected data to analysis and results, using a comprehensive SQLite database for data management that tracks all components of a remote monitoring program. This framework enables the tracking of analyses and research/management objectives through time.
3. We illustrate the AMMonitor approach with the example of evaluating an occurrence-based management objective for a target species. First, we provide an overview of the database and data management approach. Next, we illustrate a few available workflows: temporally adaptive sampling, automated detection of species sounds from acoustic recordings and aggregation of automated detections into an encounter history for use in an occupancy analysis, the outcome of which can be analysed with respect to the motivating management objective.
4. Without a comprehensive framework for efficiently moving from raw remote monitoring data collection to results and analysis, monitoring programs are limited in their capacity to systematically characterize ecological processes and inform management decisions through time. AMMonitor provides an option for such a framework. Code, comprehensive documentation and step-by-step examples are available online at https://code.usgs.gov/vtcfwru/AMMonitor

KEYWORDS

adaptive framework, AMMonitor, autonomous monitoring units, biodiversity, monitoring, R package, remote monitoring

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2020 University of Vermont. *Methods in Ecology and Evolution* published by John Wiley & Sons Ltd on behalf of British Ecological Society.

1 | INTRODUCTION

Amid climate change and rapidly shifting land uses, effective methods for monitoring natural resources are critical to support scientifically informed resource management decisions (Allen & Garmestani, 2015; Holling, 1978; Lee, 1994; Pollock et al., 2002; Walters, 1986).

The practice of using autonomous monitoring units (AMUs) to monitor ecological systems has grown in the past decade, with monitoring projects focused on target species (including birds, bats, amphibians, insects and terrestrial/marine mammals; August et al., 2015; Burton et al., 2015), phenology (Crimmins & Crimmins, 2008; Miller-Rushing, Evenden, Gross, Mitchell, & Sachs, 2011) and soundscapes (Lynch, Joyce, & Fristrup, 2011; Miller, 2008; Pijanowski et al., 2011). AMUs confer many benefits. Primarily, they can be deployed for long periods of time to collect massive amounts of data, such as audio recordings and photographs. Having a record of audio and photo data allows researchers to carefully verify and analyse species identifications or other research targets a posteriori (Hobson, Rempel, Greenwood, Turnbull, & Wilgenburg, 2002; Willi et al., 2019).

However, automated methods have several limitations. First, the data management requirements of an AMU research effort are often immense. A monitoring program is a collection of research objectives, monitoring locations, equipment, people and data files, with multiple components to manage. Lacking a systematic data management approach, new entrants to the field may make imprudent choices about data organization and subsequent automated processing. Second, monitoring data are typically stored on AMUs until retrieved by researchers, causing time lapses between data collection, analysis and results. Such delays hamper the ability to efficiently address pressing ecological challenges and track progress towards management objectives. Without a comprehensive framework for moving from raw data collection to results and analysis, monitoring programs are limited in their capacity to characterize ecological processes and to inform management decisions (Williams, 2011; Williams & Brown, 2016).

To address several of these issues, we developed AMMonitor, an open source R package dedicated to collecting, storing, organizing and analysing AMU data in a way that (a) can efficiently process and store large quantities of diverse information; (b) use statistical learning algorithms to detect acoustic targets of interest; and (c) leverage existing R analytics, while inviting new analytical approaches. The AMMonitor package builds upon our R packages, monitoR (Hafner & Katz, 2018; Katz, Hafner, & Donovan, 2016a, 2016b) and AMModels (Donovan & Katz, 2018; Katz & Donovan, 2018), and is freely available at https://code.usgs.gov/vtcfwru/ammonitor. The website's Wiki provides thorough documentation and instructions for use, including extensive examples and sample data. Windows users can download the zip file from https://code.usgs.gov/vtcfwru/ammonitor/-/archive/master/ammonitor-master.zip. Mac or Linux users can download the tar.gz file from https://code.usgs.gov/vtcfwru/ammonitor/-/archive/master/ammonitor-master.tar.gz. These condensed files can be installed in R from the package archive file. We invite the R community to collaborate by submitting a pull request.

Broadly, the AMMonitor approach starts with ecological hypotheses or natural resource management objectives (Figure 1a). To test hypotheses or to evaluate the state of a resource with respect to a management objective, data are collected using AMUs (Figure 1b). The AMUs optionally use a wireless network to deliver acoustic recordings and photographs to disk or cloud storage (Figure 1c). Raw and processed data are stored in a SQLite database (Figure 1d). Data can be analysed with a variety of analytical methods, such as species occurrence models or soundscape analyses (Figure 1e). Analyses can be summarized and reported (Figure 1f), and resulting outputs assessed with respect to hypotheses or objectives to track progress towards research or management goals. The cycle can then repeat. Thus, the AMMonitor package places the monitoring data into an adaptive management framework (Williams, 2011).

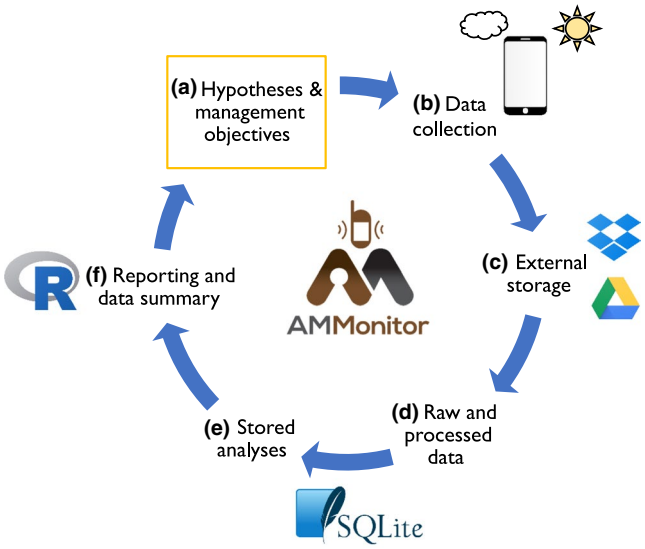

FIGURE 1 The AMMonitor framework begins with basic research hypotheses or applied resource management objectives (a), proceeds to data collection (b) with external storage on cloud or disk (c), where raw and processed data are stored in a SQLite database (d), and finishes with stored analyses (e) that can be compared to the motivating objectives via reporting and data summary that provides results a researcher or manager can evaluate (f)

The AMMonitor approach was developed with a prototype of 20 smartphone-based AMUs, with a focus on acoustic monitoring (Balantic & Donovan, 2019a, 2019b, 2019c). Since then, we have added the capacity to use the smartphone's camera by enabling timed photographs and motion-triggered photographs, allowing the smartphones to act as camera traps (Donovan et al., in prep.). However, AMMonitor does not require the use of smartphones. Its flexibility permits the analysis of data collected by other autonomous devices, and further facilitates storage of results from other analytical systems for additional processing in R.

2 | OVERVIEW OF FUNCTIONALITY

AMMonitor consists of R functions for analysis of AMU data and an accompanying data management system based on SQLite. SQLite is

a 'self-contained, high reliability, embedded, full-featured, public domain, SQL database engine' (SQLite, 2019). Unlike client/server SQL database options like MySQL, Oracle, SQL Server or PostgreSQL, SQLite poses the advantages of simple user set-up and no requirement to configure or maintain a server process. AMMonitor employs the R package RSQLite (Müller, Wickham, James, & Falcon, 2018) to connect R with the database and uses R functions to retrieve externally stored AMU files. Users can create a new, empty AMMonitor database with the *dbCreate*() function, or a sample database with the *dbCreateSample*() function, which allows new users to explore and test the package functions using simulated data. Both functions create a single '.sqlite' file, which can store all of the monitoring program's data up to 140 terabytes. New tables can be added by SQLite code passed with function, *DBI::dbSendQuery*(). Monitoring programs that process extremely large amounts of data (>1 TB) have multiple concurrent users, and have a dedicated database manager may consider adapting the SQLite database to a server/client version.

Database tables (highlighted in bold in this paragraph) store data and metadata about the overall monitoring effort. The tables in Figure 2 illustrate the generalized workflow. First, a monitoring effort is driven by an agency's or researcher's **objectives** (Figure 2a). These objectives are often, but not always, **species**-centred (Figure 2b). The **people** table (Figure 2c) stores information about members of the monitoring team. People deploy **equipment** (Figure 2d) across **locations** (Figure 2e) to monitor ecosystems via smartphones (optionally), tracked through the **deployment**

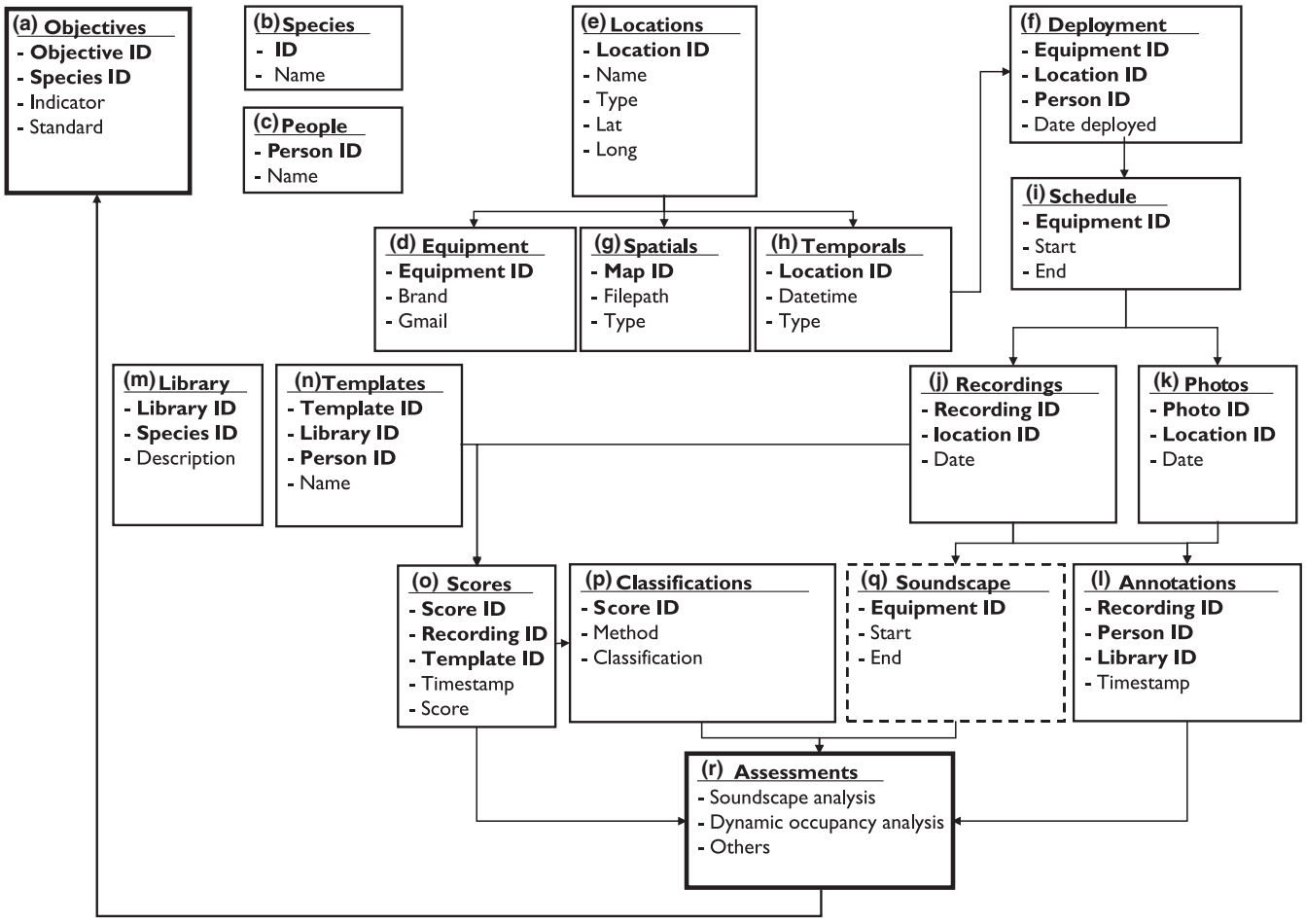

FIGURE 2 Overview of AMMonitor SQLite database schema. Database tables (capitalized in bold in boxes) store data and metadata about the overall monitoring effort. Example field names are identified with a bullet, with primary or foreign keys highlighted in bold. The tables Objective and Assessments are highlighted with a dark border to indicate that a monitoring or research program normally starts with objectives and ends with assessments. Monitoring efforts are driven by objectives (a) which are often species-centered (b). People on the monitoring team (c) deploy equipment (d) at locations (e), tracked in the deployment table (f). Location-specific spatial (g) and temporal (h) information is also stored. Deployed equipment collects recordings and/or photos on a schedule (i). Collected files are delivered to and remain in external storage, and metadata about external-based files are stored in the recordings (j) and photos (k) tables. Team members can log annotations (l) for target signals that are linked to a signal library (m), and/or can create templates (n) that automatically search for target signals. When templates are run against incoming recordings, the scores table (o) stores metrics indicating the closeness of a signal to the template. Statistical learning classifiers can return the probability that a detected event is the target signal, stored in the classifications table (p). The soundscape table (q) is outlined with a dotted line to illustrate AMMonitor's flexible design to accommodate audio or image analysis from other R packages. Data sources can be used to analyze the state of the ecosystem with respect to research hypotheses or management objectives. The assessments table (r) can be used to store analysis metadata

(Figure 2f) table. Location-specific spatial and temporal information is stored in the **spatials** (Figure 2g) and **temporals** (Figure 2h) tables respectively. Deployed equipment collects recordings and/or photos on a **schedule** (Figure 2i) transmitted to each phone's Google calendar daily, if using smartphone-based monitoring. Files are delivered to external storage on disk or in the cloud; metadata about external-based files are stored in the **recordings** (Figure 2j) and **photos** (Figure 2k) tables. Team members can manually search files for target **species** or target signals and log **annotations** (Figure 2l) linked to a signal **library** (Figure 2m). To facilitate automated detection of target sounds, team members can create **templates** (Figure 2n) of target signals. Templates are run against incoming recordings; the **scores** table (Figure 2o) stores metrics indicating the closeness of a signal to the template. Statistical learning classifiers are used to return the probability that a detected event is the target signal, stored in the **classifications** table (Figure 2p).

The **soundscape** table (Figure 2q) is outlined with a dotted line to illustrate AMMonitor's flexible design to accommodate audio or image analysis from other R packages. For example, the R package soundecology (Villanueva-Rivera & Pijanowski, 2018) calculates soundscape ecology indices such as acoustic complexity and diversity. AMMonitor's *soundscape*() function loads soundecology to conduct these analyses, and stores outputs in the **soundscape** table where they can be analysed within the AMMonitor workflow. Similarly, the R package mlwic (Tabak et al., 2019) allows users to train convolutional neural network models to scan monitoring images for target species. MLWIC users can leverage the AMMonitor workflow by creating a function that inserts MLWIC outputs to a new AMMonitor table. If desired, new fields can be added to existing tables to extend a table's definition. For example, researchers who use AudioMoths (Hill et al., 2018) to collect acoustic data may customize the **recordings** table by adding fields that reflect AudioMoth-specific information such as the sample rate, gain and sleep duration. Modularity and flexible customization capacity ensure that AMMonitor can take advantage of a rich variety of existing solutions, while facilitating new approaches. **Classifications, scores**, **annotations** and other data sources can be used to analyse the state of the ecosystem with respect to research hypotheses or management **objectives**, which provide the primary motivation for the monitoring program. The **assessments** table (Figure 2r) can be used to store analysis metadata, linking to user-created R scripts.

Data can be inserted, deleted, updated or read with R functions, and many of the tables are automatically populated by R. If desired, users can download a Microsoft Access-based front end from the AMMonitor website to use as a customizable database interface.

3 | EXAMPLE WORKFLOW FOR MONITORING A TARGET SONGBIRD

We illustrate the AMMonitor workflow with a hypothetical example for a management agency. The management agency has three management objectives (Table 1), each concerning the occurrence of a focal species across the managed landscape. Progress towards each management objective can be monitored using an occupancy framework, which tracks the indicator psi (Ψ), the proportion of management area occupied by the species (MacKenzie et al., 2002). Objectives are to increase the occurrence values of Species 1, decrease occurrence values of the invasive Species 2 and maintain current occurrence values of Species 3. All three species have minimum, standard, and maximum values of Ψ deemed acceptable for meeting the management objective. In the AMMonitor approach, a dynamic occupancy modelling framework can be used to measure progress towards management objectives (Miller et al., 2013). This framework provides estimates of Ψ through time, in addition to insights about factors that may affect site colonization by desired species (i.e. Species 1) or local extinction of undesired species (i.e. Species 2). Some factors, such as invasive vegetation or key habitat features, may be directly influenced through management action enacted on the landscape. As monitoring progresses, understanding of how such variables affect the distribution of target species can be updated through time. Objectives are tracked in the **objectives** table of the AMMonitor database, and progress towards them is evaluated in the **assessments** table.

A research team tasked with these objectives will establish a robust study design, entering target species information into the **species** table, AMUs into the **equipment** table and sampling locations into the **locations** table. Spatial layers can be tracked in the **spatials** table. If using smartphone-based monitoring, the team will also track Google accounts in the **accounts** table.

3.1 | Temporally adaptive sampling

Once the study design has been chosen and monitoring locations are established, the next step is to schedule sampling periods for

Objective	Indicator	Direction	Minimum	Standard	Maximum
1. Maximize occurrence of species 1	Ψ	Maximize	0.35	0.40	0.45
2. Minimize occurrence of species 2	Ψ	Minimize	0.00	0.15	0.30
3. Maintain occurrence of species 3	Ψ	Maintain	0.30	0.35	0.40

TABLE 1 Example of hypothetical objectives, progress towards which can be monitored via acoustic recordings collected by autonomous monitoring units. The hypothetical objectives include a monitoring indicator and a direction, as well as minimum, standard and maximum values for the indicator, against which objective progress can be measured

each AMU at each location. In this section, we will assume that the monitoring team uses smartphones as AMUs, where units receive acoustic monitoring schedules via Google Calendar.

Several options exist for setting schedules with Google Calendar. First, users may send a monitoring schedule of their choice using the function *schedulesFixed*(), which enables direct specification of dates, times and durations of monitoring activity for recordings, motion capture or timed photographs. Alternatively, the function *scheduleSun*() facilitates expedient scheduling of monitoring activities based on sunrise and sunset times at monitoring locations.

Beyond these options, one of AMMonitor's novel methodological features, temporally adaptive sampling, was heavily inspired by smartphone-based monitoring systems (Balantic & Donovan, 2019a). In monitoring circumstances where cellular coverage is available, a key benefit of smartphone-based monitoring is the minimal lag between data collection and analysis. Data can be transmitted from the smartphone to external storage in near-real time, facilitating rapid analysis and minimizing cumbersome field trips. A drawback of using the cellular network is that smartphone data plans can be expensive. It thus behooves smartphone-based monitoring programs to avoid squandering sampling resources.

We developed a temporally adaptive sampling algorithm that allows acoustic monitoring events to be scheduled when target species are likely to be acoustically available for capture on an audio recording. The algorithm adapts by prioritizing monitoring activity based on previous target detections at each site. In the **priorities** table, users can set monitoring priority weights for target species at each monitoring location. User-specified monitoring prioritization of species (via *prioritySet*(), *priorityInit*(); Figure 3a) is combined with user-supplied target species activity models that predict when focal species will be acoustically available. Target species activity models might be built based on real data, literature values, expert opinion and/or simulation (via *simGlm*(); Figure 3b). These models can be stored in and retrieved from an AMModels library (Donovan & Katz, 2018), a vehicle for efficiently storing models and their metadata. The prescribed acoustic sampling schedule is generated via *scheduleOptim*(), which uses a combination of the species activity models, the weather forecast (acquired via the Dark Sky API (Dark Sky, 2017) with *temporalsGet*()), and the cumulative probability of acoustic capture based on audio sampling efforts thus far in the study period (see Balantic & Donovan, 2019a; Figure 3c). Sampling times and weather data are logged in the **schedule** and **temporals** tables respectively. The cellular network is used to send each smartphone a Google Calendar with its prescribed audio sampling schedule for the next day, to improve the chances of detecting a focal species, if present.

Temporally adaptive sampling is useful in some—but not all—research circumstances. Interested users may explore the utility of temporally adaptive sampling in a simulation framework, using *simGlm*() in combination with *scheduleOptim*() to investigate whether this approach would yield benefits for their specific monitoring circumstances. See https://github.com/cbalantic/temporally-adaptive-sampling for code to accompany Balantic and Donovan (2019a) that

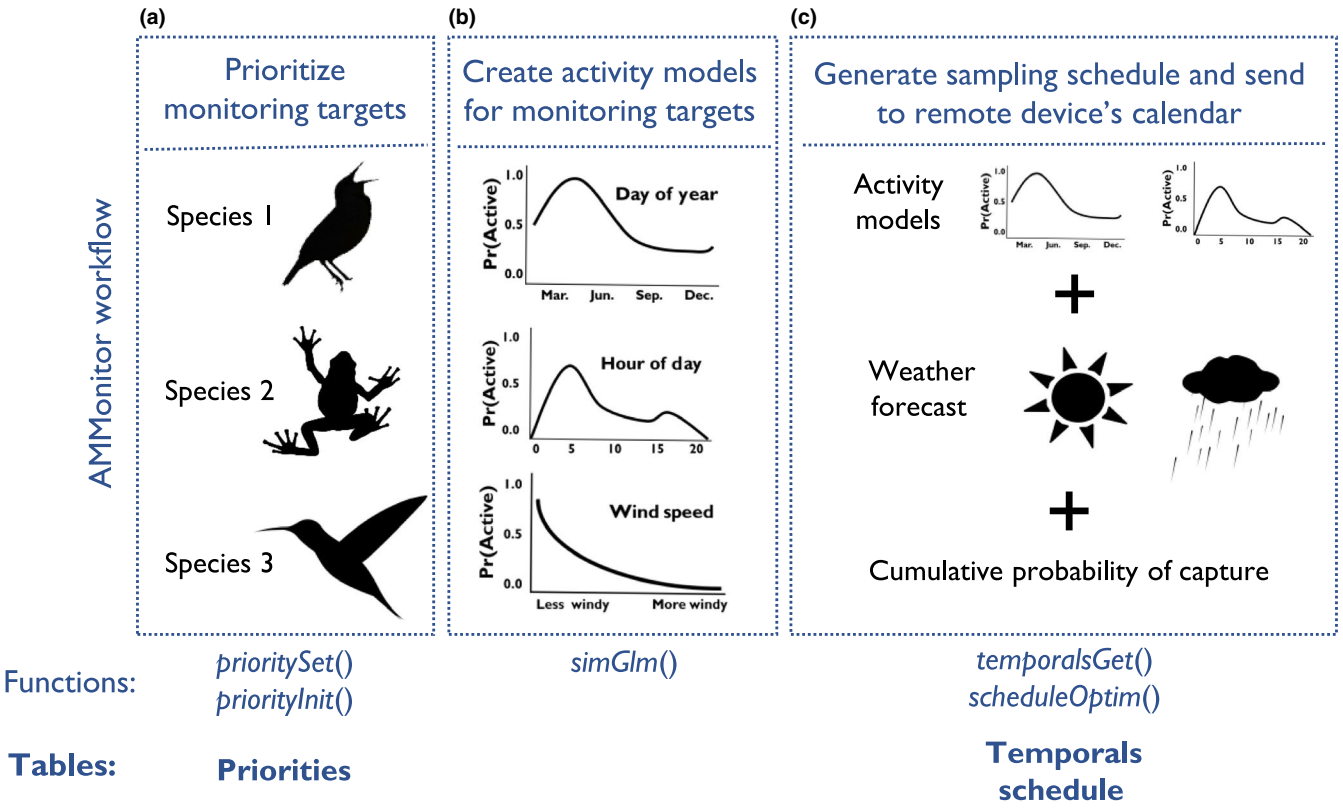

FIGURE 3 Temporally adaptive sampling workflow in AMMonitor, with some associated functions and tables

provides a comprehensive illustration of how such a simulation may be conducted.

3.2 | Semi-automated detection of target sounds

Regardless of whether monitoring is being conducted via smartphone or another AMU like an AudioMoth (Hill et al., 2018), remote monitoring programs may acquire large amounts of audio data. Humans may inspect files individually, labelling any observed targets with the *annotateRecording*() function, which logs records in the **annotations** table. However, intensive manual inspection of all files may not be possible. Researchers may minimize manual processing time of audio recordings by creating either binary point matching or spectrogram cross correlation templates to search for target signals (MONITOR R package: Hafner & Katz, 2018). With the *scoresDetect*() function, templates are used in a moving window analysis to detect similar signals in other audio recordings, with detections logged in the **scores** table (Figure 4a). Many signals detected via templates may be false alarms which do not contain the target signal. Users may label a subset of detections as target signals and false alarms with *scoresVerify*(), and use *plotVerifications*() to plot and check their work (Figure 4b). User-labelled verifications can then be used in the function *classifierModels*() to train a suite of statistical learning classifiers, including regularized logistic regression, radial support vector machines, linear support vector machines, random forests and *k*-nearest neighbours; performance of the models can be assessed using *classifierPerformance*() (Figure 4c). High-performing models can be used on incoming detections to predict the probability that an unknown signal is from the target species (Figure 4d), improving the quality of monitoring data (Balantic & Donovan, 2019b). The end result is a collection of target signal probabilities for each detected event (stored in the **classifications** table).

3.3 | Analysis and assessment example: Aggregating species detections into dynamic occupancy models

Remotely collected audio and photo data can be processed and analysed in a variety of ways. Any analytical methods undertaken should be directly informed by management objectives and associated indicators. In this section, we revisit the hypothetical management agency's first objective from Table 1, which is to maximize the occurrence of Species 1. The agency uses the indicator occupancy (Ψ) to track progress through time towards the goal of having Species 1 occur at between 0.35 and 0.45 of sites (Figure 5a).

The agency implements a remote monitoring program, deploying AMUs at monitoring locations to collect recordings and/or photos (Figure 5b). After creating an acoustic template to automatically detect potential Species 1 vocalizations, statistical learning classifiers are trained to distinguish between Species 1 target signals and false alarms (using the workflow in Figure 4). Now, each incoming detection for Species 1 is assigned a probability that it is a target signal. Using *shapeOccupancy*(), acoustic detections are aggregated into an encounter history for a dynamic occupancy model that accommodates false positives (Balantic & Donovan, 2019c), and the R package

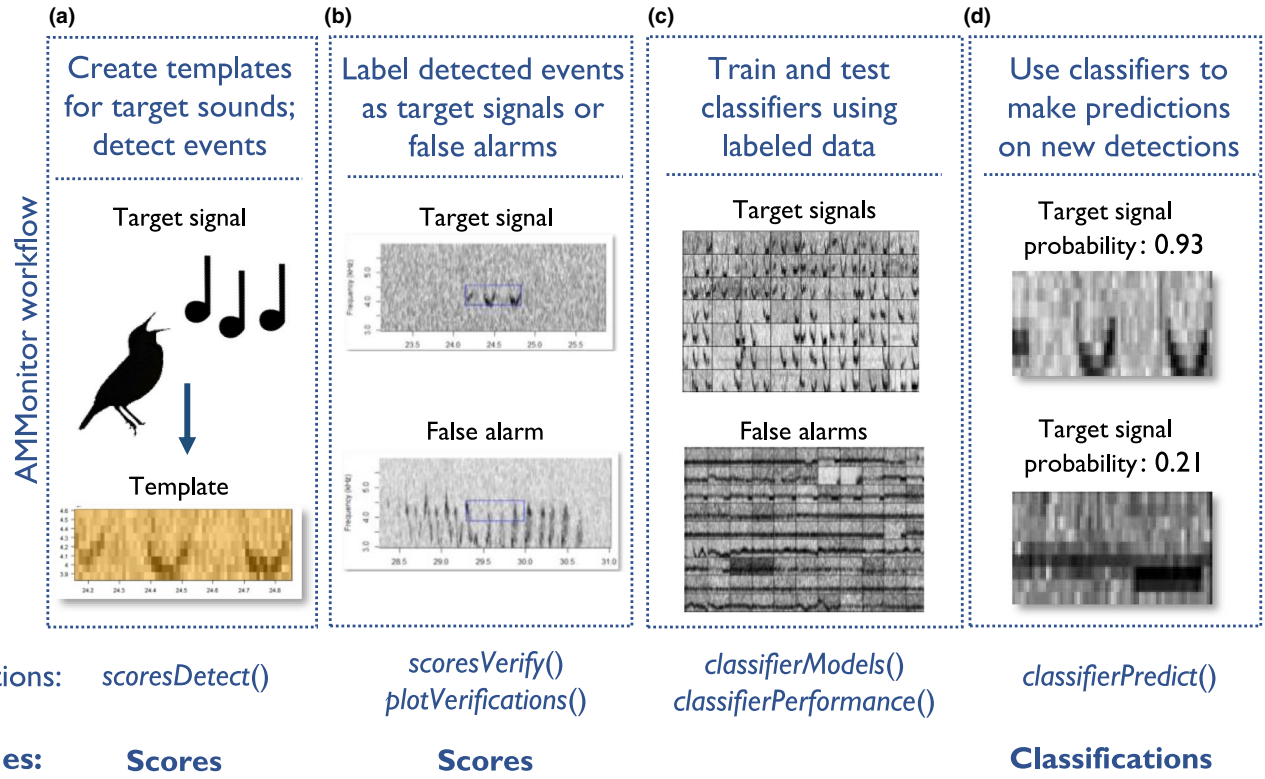

FIGURE 4 Workflow for semi-automated detection of target sounds in AMMONITOR, with some associated functions and tables

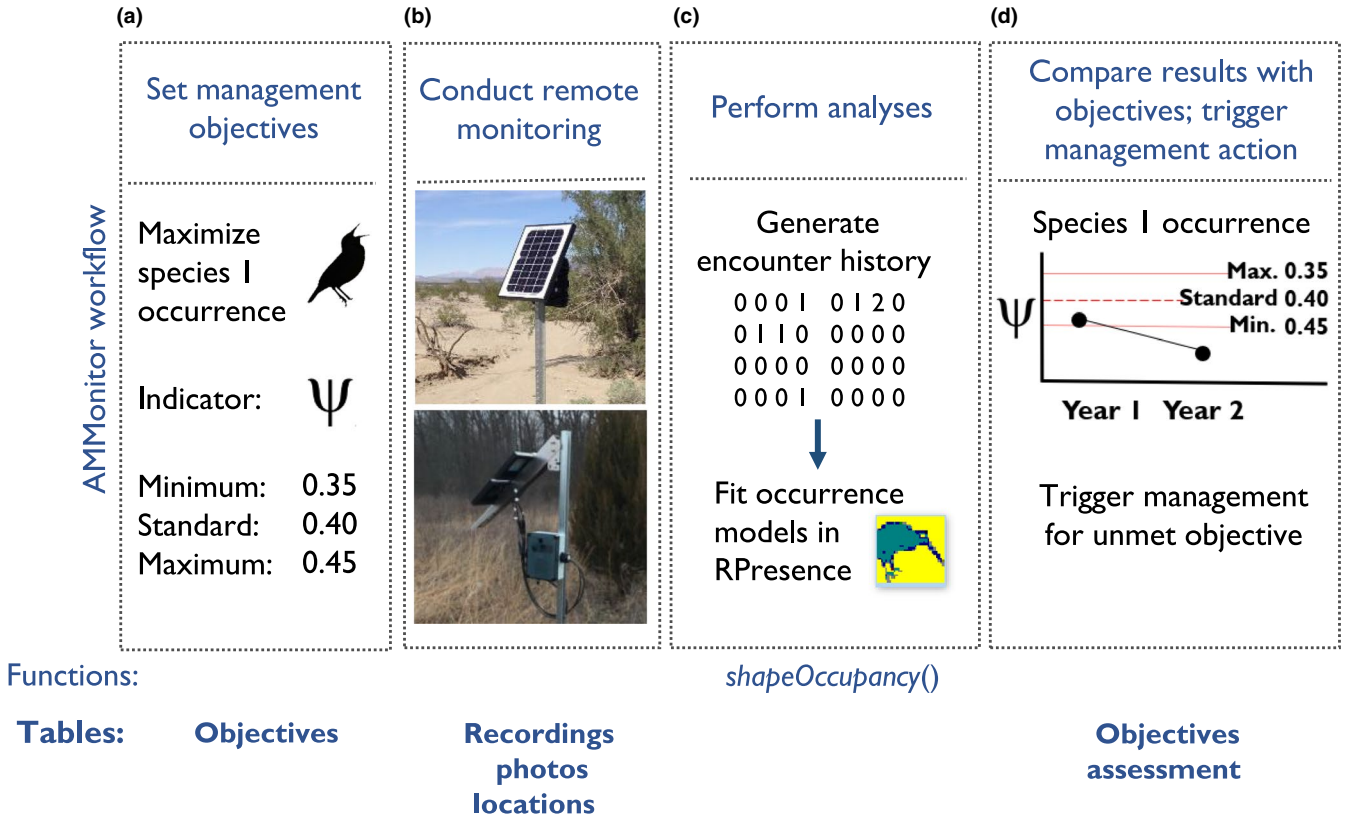

FIGURE 5 Sample analytical workflow in AMMonitor, with some associated functions and tables, for aggregating species detections into dynamic occupancy models

RPresence is used to fit models of occupancy, colonization and extinction patterns for Species 1 (Figure 5c). Results from the analysis are then compared with the agency's objectives. Figure 5d illustrates the possibility that Species 1 occurrence rates are not falling within the 0.35–0.45 target window, potentially triggering explicit management action to increase Species 1 occupancy rates. Thus, **objectives** are assessed by comparing the results of an analysis with the stated objective. The analysis can be logged in the **assessments** database table to provide a trace.

4 | SCRIPTS AND BATCH PROCESSING

Autonomous monitoring units can generate massive streams of information daily, and many aspects of the AMMonitor workflow can be automated. For example, the **temporals** table can be populated daily with the next day's weather forecast at monitoring locations. Other examples of recurring workflows in AMMonitor include:

1. Scheduling recordings via *scheduleOptim*().
2. Searching recordings for target signals with *scoresDetect*().
3. Predicting the probability that a detected event is a target signal using *classifierPredict*().

Many of these tasks may be written as R scripts, logged in the AMMonitor **scripts** table and sourced daily or run in batch mode.

5 | CONCLUSIONS

AMMonitor fills a gap in remote biodiversity monitoring by providing a solution for systematic data management, translation of raw data into results for analysis, and methodical tracking of progress towards objectives through time. It is an open-source monitoring system with utility for a variety of remote biodiversity monitoring projects. Here, we have demonstrated the workflow of translating raw acoustic data into occupancy models, but AMMonitor offers a flexible ecosystem customizable for a variety of analytical approaches. Although acoustic processing has been a focus to date, the package is designed to enable further development of nascent work supporting image-based and other monitoring efforts. Competence in R is required for user customization, but the package also offers a Microsoft Access Database front end for data entry, and future efforts will leverage RShiny (Chang, Cheng, Allaire, Xie, & McPherson, 2019) to enhance the package experience for non-R users. AMMonitor is part of the Adaptive Management Toolkit, a family of R packages developed at the Vermont Cooperative Fish and Wildlife Research Unit.

ACKNOWLEDGEMENTS

This project was supported by the Bureau of Land Management and the National Park Service. We thank Mark Massar (BLM) for providing the opportunity to work in the Riverside East Solar Energy Zone, Randy Knutson (NPS) for providing the opportunity to work at Indiana Dunes National Lakeshore, Jonathan Katz and Jim Hines for

their code contributions to the AMMonitor package and reviewers for their helpful comments. Any use of trade, firm or product names is for descriptive purposes only and does not imply endorsement by the US Government. The Vermont Cooperative Fish and Wildlife Research Unit is jointly supported by the US Geological Survey, University of Vermont, Vermont Department of Fish and Wildlife and Wildlife Management Institute.

AUTHORS' CONTRIBUTIONS

C.B. and T.D. developed the code and wrote the manuscript; C.B. wrote most of the code related to species detection, classification algorithms and adaptive sampling; T.D. designed the database and workflow and wrote most of the database code.

DATA AVAILABILITY STATEMENT

The latest version of AMMonitor can be downloaded as a zip or tar.gz file from Gitlab at https://code.usgs.gov/vtcfwru/ammonitor, or installed directly in R using *install_git*(). The AMMonitor Gitlab Wiki page contains comprehensive documentation and step-by-step examples using a sample database. Alternatively, users can clone AMMonitor to their own git account with the command: **git clone** https://code.usgs.gov/vtcfwru/ammonitor.git. To work with an exact copy of the code described at the time of this publication, use the command: **git checkout 7d740071cc8eb28d8473b4ab0e6802653a546b12**.

ORCID

Cathleen Balantic https://orcid.org/0000-0003-2043-0975
Therese Donovan https://orcid.org/0000-0001-8124-9251

REFERENCES

Allen, C. R., & Garmestani, A. S. (2015). Adaptive management. In C. R. Allen & A. S. Garmestani (Eds.), *Adaptive management of social-ecological systems* (pp. 1–10). Dordrecht, The Netherlands: Springer.

August, T., Harvey, M., Lightfoot, P., Kilbey, D., Papadopoulos, T., & Jepson, P. (2015). Emerging technologies for biological recording. *Biological Journal of the Linnean Society*, *115*(3), 731–749. https://doi.org/10.1111/bij.12534

Balantic, C. M., & Donovan, T. M. (2019a). Temporally adaptive acoustic sampling to optimize detection across a suite of focal species. *Ecology and Evolution*, *9*, 10582–10600. https://doi.org/10.1002/ece3.5579

Balantic, C. M., & Donovan, T. M. (2019b). Statistical learning mitigation of false positives from template-detected data in automated acoustic wildlife monitoring. *Bioacoustics*, *29*(3), 296–321. https://doi.org/10.1080/09524622.2019.1605309

Balantic, C. M., & Donovan, T. M. (2019c). Dynamic wildlife occupancy models using automated acoustic monitoring data. *Ecological Applications*, *20*. https://doi.org/10.1002/eap.1854

Burton, A. C., Neilson, E., Moreira, D., Ladle, A., Steenweg, R., Fisher, J. T., … Boutin, S. (2015). Wildlife camera trapping: A review and recommendations for linking surveys to ecological processes. *Journal of Applied Ecology*, *52*(3), 675–685.

Chang, W., Cheng, J., Allaire, J. J., Xie, Y., & McPherson, J. (2019). *shiny: Web application framework for R*. R package version 1.4.0. Retrieved from https://CRAN.R-project.org/package=shiny

Crimmins, M. A., & Crimmins, T. M. (2008). Monitoring plant phenology using digital repeat photography. *Environmental Management*, *41*(6), 949–958. https://doi.org/10.1007/s00267-008-9086-6

Dark Sky. (2017). *Dark sky API [Application Programming Interface]*. Retrieved from https://darksky.net

Donovan, T. M., & Katz, J. E. (2018). AMModels: An R package for storing models, data, and metadata to facilitate adaptive management. *PLoS ONE*, *13*(2), e0188966. https://doi.org/10.1371/journal.pone.0188966

Hafner, S., & Katz, J. (2018). monitoR: Acoustic template detection in R. R package version 1.0.7. Retrieved from https://cran.r-project.org/web/packages/monitoR/index.html

Hill, A. P., Prince, P., Piña Covarrubias, E., Doncaster, C. P., Snaddon, J. L., & Rogers, A. (2018). AudioMoth: Evaluation of a smart open acoustic device for monitoring biodiversity and the environment. *Methods in Ecology and Evolution*, *9*(5), 1199–1211. https://doi.org/10.1111/2041-210X.12955

Hobson, K. A., Rempel, R. S., Greenwood, H., Turnbull, B., & Van Wilgenburg, S. L. (2002). Acoustic surveys of birds using electronic recordings: New potential from an omnidirectional microphone system. *Wildlife Society Bulletin*, *30*(3), 709–720.

Holling, C. S. (1978). *Adaptive environmental assessment and management*. Chichester, UK: John Wiley & Sons.

Katz, J., & Donovan, T. (2018). *AMModels: Adaptive management model manager (version 0.1.4)*. Comprehensive R Archive Network. Retrieved from https://cran.r-project.org/web/packages/AMModels/

Katz, J., Hafner, S. D., & Donovan, T. M. (2016a). Assessment of error rates in acoustic monitoring with the R package monitoR. *Bioacoustics*, *25*, 177–196. https://doi.org/10.1080/09524622.2015.1133320

Katz, J., Hafner, S. D., & Donovan, T. M. (2016b). Tools for automated acoustic monitoring within the R package monitoR. *Bioacoustics*, *25*, 197–210. https://doi.org/10.1080/09524622.2016.1138415

Lee, K. N. (1994). *Compass and gyroscope: Integrating science and politics for the environment*. Washington, DC: Island Press.

Lynch, E., Joyce, D., & Fristrup, K. (2011). An assessment of noise audibility and sound levels in US National Parks. *Landscape Ecology*, *26*(9), 1297–1309. https://doi.org/10.1007/s10980-011-9643-x

MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, A., & Langtimm, C. A. (2002). Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, *83*, 2248–2255. https://doi.org/10.1890/0012-9658(2002)083[2248:ESORWD]2.0.CO;2

Miller, N. P. (2008). US National Parks and management of park soundscapes: A review. *Applied Acoustics*, *69*(2), 77–92. https://doi.org/10.1016/j.apacoust.2007.04.008

Miller, D. A., Nichols, J. D., Gude, J. A., Rich, L. N., Podruzny, K. M., Hines, J. E., & Mitchell, M. S. (2013). Determining occurrence dynamics when false positives occur: Estimating the range dynamics of wolves from public survey data. *PLoS one*, *8*(6), e65808. https://doi.org/10.1371/journal.pone.0065808

Miller-Rushing, A., Evenden, A., Gross, J., Mitchell, B., & Sachs, S. (2011). Parks use phenology to improve management and communicate climate change. *Park Science*, *28*, 61–67.

Müller, K., Wickham, H., James, D. A., & Falcon, S. (2018). *RSQLite: 'SQLite' interface for R (version 2.1.1)*. Retrieved from https://cran.r-project.org/web/packages/RSQLite/index.html

Pijanowski, B. C., Villanueva-Rivera, L. J., Dumyahn, S. L., Farina, A., Krause, B. L., Napoletano, B. M., … Pieretti, N. (2011). Soundscape ecology: The science of sound in the landscape. *BioScience*, *61*(3), 203–216. https://doi.org/10.1525/bio.2011.61.3.6

Pollock, K. H., Nichols, J. D., Simons, T. R., Farnsworth, G. L., Bailey, L. L., & Sauer, J. R. (2002). Large scale wildlife monitoring studies: Statistical methods for design and analysis. *Environmetrics: The Official Journal of the International Environmetrics Society*, *13*(2), 105–119. https://doi.org/10.1002/env.514

SQLite. (2019). *SQLite (version 3.29.0)* [Computer software]. Retrieved from https://www.sqlite.org

Tabak, M. A., Norouzzadeh, M. S., Wolfson, D. W., Sweeney, S. J., VerCauteren, K. C., Snow, N. P., … Teton, B. (2019). Machine learning

to classify animal species in camera trap images: Applications in ecology. *Methods in Ecology and Evolution*, *10*(4), 585–590. https://doi.org/10.1111/2041-210X.13120

Villanueva-Rivera, L. J., & Pijanowski, B. (2018). *Soundecology: Soundscape ecology*. R Package. Retrieved from https://cran.r-project.org/web/packages/soundecology/index.html

Walters, C. J. (1986). *Adaptive management of renewable resources*. New York, NY: Macmillan Publishers Ltd.

Willi, M., Pitman, R. T., Cardoso, A. W., Locke, C., Swanson, A., Boyer, A., … Fortson, L. (2019). Identifying animal species in camera trap images using deep learning and citizen science. *Methods in Ecology and Evolution*, *10*(1), 80–91. https://doi.org/10.1111/2041-210X.13099

Williams, B. K. (2011). Adaptive management of natural resources-framework and issues. *Journal of Environmental Management*, *92*, 1346–1353.

Williams, B. K., & Brown, E. D. (2016). Technical challenges in the application of adaptive management. *Biological Conservation*, *195*, 255–263.

How to cite this article: Balantic C, Donovan T. AMMonitor: Remote monitoring of biodiversity in an adaptive framework with R. *Methods Ecol Evol*. 2020;00:1–9. https://doi.org/10.1111/2041-210X.13397