Canadian Journal of Zoology
Revue canadienne de zoologie

BatScope manages acoustic recordings, analyses calls and classifies bat species automatically

Journal:	<i>Canadian Journal of Zoology</i>
Manuscript ID	cjz-2017-0103.R2
Manuscript Type:	Article
Date Submitted by the Author:	23-Nov-2017
Complete List of Authors:	Obrist, Martin; Swiss Federal Research Institute WSL, Biodiversity and Conservation Biology Boesch, Ruedi; Swiss Federal Research Institute WSL, Landscape Dynamics
Keyword:	bats, echolocation, pattern recognition, species identification, database

BatScope manages acoustic recordings, analyses calls and classifies bat species automatically³

Obrist, M.K. ¹, Boesch, R. ²

¹Swiss Federal Research Institute WSL, Biodiversity and Conservation Biology, Zürcherstrasse 111, CH-8903

Birmensdorf

Email: martin.obrist@wsl.ch

²Swiss Federal Research Institute WSL, Landscape Dynamics, Zürcherstrasse 111, CH-8903 Birmensdorf

Email: ruedi.boesch@wsl.ch

Corresponding author:

Martin K. Obrist

Swiss Federal Research Institute WSL, Biodiversity and Conservation Biology, Zürcherstrasse 111, CH-8903

Birmensdorf

Phone: +41 44 739 2466

Fax: +41 44 739 2215

Email: martin.obrist@wsl.ch

³This article is one of a series of papers arising from “*Learning to Listen* — Second International Symposium on Bat Echolocation Research: Tools, Techniques, and Analysis” that was held in Tucson, Arizona, USA, 26 March – 1 April 2017. Invited speakers were encouraged to submit manuscripts based on their talks, which then went through the normal *Canadian Journal of Zoology* peer-review process.

BatScope manages acoustic recordings, analyses calls and classifies bat species automatically

Obrist, M.K., Boesch, R.

Abstract

BatScope* is a free application for processing acoustic high-frequency recordings of bats. It can import data from recorders such as BatLogger** including associated meta-data information. The resulting content can be filtered visually as spectrograms or according to data fields, and displayed. Automated processing includes detecting and extracting of echolocation calls, filtering noise and measuring statistical parameters. Calls are classified to species by statistically matching to a reference database. A weighted list of classifiers helps to assign the most likely species per call. Classifiers were trained on 19'636 echolocation calls of 27 European bat species. When classifiers all agree on a species (76.4% of all cases), average correct classification rate reaches 95.7%. A sequence's summary statistic indicates the most likely species occurring therein. Classifications can be verified visually, by filtering and acoustic comparison with reference calls. Procedures are available for e.g. excluding dubious cutouts from the statistics, and for accepting or overriding the proposed species assignment. Acoustic recordings can be exported and exchanged with other users. Finally, the verified results can be exported to spreadsheets for further analyses and reporting. We currently reprogram BatScope using Java, PostgreSQL and R, to reach a unified and portable software architecture.

* <http://www.batscope.ch>

** <http://www.batlogger.com>

Keywords: bats, echolocation, pattern recognition, species identification, database

Introduction

Acoustic bat identification: Advent and evolution

Our understanding of bat echolocation has come a long way since this fascinating sensory modality was first detected and later described (Griffin 1958). Eavesdropping on bats has become an established mode of observation, although it is not equally suitable for all species, nor for all types of scientific questions. Estimating a species' abundance or area of occupancy from its acoustic signals for example, still requires considerable effort. Great technological advances in acoustic equipment, survey approaches and processing possibilities have been made (Fenton et al. 1987), especially in the last two decades (Brigham et al. 2004; Parsons and Obrist 2004). Acoustic monitoring has become accessible to a much wider community of bat enthusiasts than could have been foreseen. It has been possible to distinguish many species since the 1980s (Ahlén 1981; Fenton et al. 1983; Zingg 1990), but humans and machines still struggle (Vaughan et al. 1997; Parsons and Jones 2000; Russo and Jones 2002; Redgwell et al. 2009) to differentiate many others (e.g. Genus *Myotis*)

Identifying individuals on the basis of their calls has been tackled repeatedly (Masters et al. 1995; Obrist 1995; Pearl and Fenton 1996; Burnett 2001; Siemers and Kerth 2006; Kazial et al. 2008; Yovel et al. 2009; Arnold and Wilkinson 2011) but still seems to only work in some circumstances or for some individuals.

Acoustic methods allow non-invasive surveys of bat activity and habitat use. They have become especially valuable for monitoring the effects of new threats to bats (O'Shea et al. 2016) such as diseases (WNS: Blehert et al. 2009; USFWS 2017) or the impact of energy change technologies such as wind farms. Consequently new hardware devices and a variety of commercial software solutions have come on the market. So far these solutions have not been reproducibly documented or tested in independent surveys, and are thus disputed in the scientific community (Russo and Voigt 2016).

Bat echolocation research has a long tradition at the Swiss Federal Research Institute WSL (Obrist 1995; Obrist et al. 2004; Obrist et al. 2008; Obrist et al. 2010; Bohnenstengel et al. 2014), and many applied questions concerning bat conservation have been addressed (Obrist et al. 2011; Frey-Ehrenbold et al. 2013; Froidevaux et al. 2014; Froidevaux et al. 2016). Different species recognition approaches have been evaluated and further developed, ranging from command line processes to a much more user-friendlier interface for data management, analysis and species classification. One such valuable research tool is WSL's software BatScope (Boesch and Obrist 2013), which is available free of charge for users to eavesdrop on bats. In this paper we explain the software, its structure and processing workflows.

Software specifications

Semi-automated software should allow users to manage their data in a transparent way. Data storage, import, processing, validation and export should all be available. Automated species recognition should be implemented as a standardized operation, and the user should not be able to tamper with the numeric results to ensure objective and reproducible results. Additionally, species classification should be regionally adaptable. Finally, the interface must allow users to verify species identifications on the basis of their expert knowledge.

A precursor of the software presented here (BatScope3) is freely available for Apple computers from <http://www.batscope.ch>. The BatScope4 version discussed here is still under development and will be freely available for Apple computers later in 2017, and for Linux and Windows in 2018. We report here design of both versions, but will concentrate on features already implemented or scheduled for version 4.

The software BatScope

Design and tools

BatScope was designed to meet very different development and user requirements, combining both exploratory analyses and automation of complete workflows. Automation with a Python based scripting interface, found in many scientific applications, was considered too error-prone for many users. Typically users with a biological background have little scripting experience, and very robust multi-threaded programming for scripting environments is still a challenge even for experienced programmers. Handling large amounts of audio data with modern laptops requires having robust multi-threaded environment built in.

In BatScope3, several processing tasks require an intermediate sqlite3-database, which is difficult to handle in a thread-safe manner. Therefore, built-in multi-threading for database and processing operations is now part of the core application and a major improvement in BatScope4 over BatScope3. The entity relationship diagram (ERD) of the database underlying BatScope4 is given in Fig. S1. in the Supplements.

PostgreSQL was chosen as the database (PostgreSQL 2017) and R as the classification engine (R Core Team 2016) to allow connection-oriented interfaces. The software R is used with the package *Rserve* (Urbanek 2013). The graphical user interface and all processing apart from the classification tasks are programmed in JavaFX 8. Several mainly computational tasks (importing, exporting, cutting, analyzing and classifying data) were implemented as plugins. This form of implementation makes customizing it easier, e.g. with other import formats and classification methods, and it also simplifies code maintenance and error tracking.

Logical structure

In BatScope4, data are hierarchically represented in a structure that maps the logic workflow of a standard task (Fig. 1). *Projects* represent the top hierarchy. They contain *collections* of an arbitrary unit, which again contain the recorded audio-*sequences*. These sequences consist of consecutive echolocation *calls*, whose signal parameters are finally processed for *classification*. Perpendicular to this hierarchy, *sequences* can be assigned to arbitrary *categories* to further structure the database content.

• Figure 1 approx. here •

Projects

Projects are the top hierarchical level and serve to integrate or merge all data for a given topic. This could be something like an environmental impact study, a research project, or any collection of associated surveys. On the file system level, *projects* represent the only type of granularity that a user can see. They contain the actual sound data, which can then be moved using the tools provided between data storage media, or detached from and re-attach to the database.

Collections

All recordings from single surveys are contained in *collections*. These might be for instance all the recordings from a single night and machine, or from one location. *Collections* mainly serve as additional layers for organizing data. In the most transparent way, they reflect the content of the data container used for a survey. *Collections* are represented only in the database but not physically in the file system (Fig. 2).

• Figure 2 approx. here •

Sequences

Sequences are successions of echolocation calls, as stored in the recording device. They are saved on the file system inside the *projects* using a hash-function to link them to the database. Usually, they are affiliated with meta-data, stored in corresponding fields in the database. A graphic representation of the spectrogram of the *sequence* is stored with the signal's wav-file too (Fig. 3).

• Figure 3 approx. here •

Calls

During processing, a detection algorithm (see below) searches the recorded *sequence* for tonal signals, and then cuts and stores them as *calls* on the file system in the hash structure of the stored *sequences*. A graphic representation of the spectrogram of the *call* is stored with the wav-file as well (Fig. 4).

• Figure 4 approx. here •

Classifications

Each single *call* is inspected for 59 temporal and spectral parameters, which are then subjected to classification algorithms. The results are stored in the *classifications* table and can be accessed through the graphical user interface (GUI). Classifications can be seen in the lower right part of Fig. 4.

Categories

To add another level of structuring, arbitrary *categories* can be attributed to *sequences*. These might be *sequences* of special interest containing interactions and social calls. These all concern a similar topic, such as a wind energy study, or a behavioural situation, e.g. 'leaving roost' in Fig. 3 (center bottom). The notes field provides a less formally structured option for storing such information.

Taxonomy

All species included in the training base are also included in the taxonomy section of the application (Fig. 5). The taxonomy section can be browsed to see what species it contains, or addressed directly from the statistics results panel (lower left list in Fig. 3).

• Figure 5 approx. here •

Processing workflow

In a typical example of general use, a project is first created. By default, the project generation will propose a save-path to a shared folder on the local file system. This allows different collaborators to share the data and save storage space. As projects can potentially grow to contain terabytes of data, the user may also specify an external volume for storing the data.

The Ultrasound audio data recorded are then imported into these projects from e.g. storage media or folders containing survey data. Different options allow data to be imported from hardware devices like Batlogger (Elekon AG 2017), Batcorder (ecoObs 2017), D500X (Pettersson 2017), SM3/4 (Wildlife Acoustics 2017) or some other systems. Finally, data can also, of course, be imported from previous versions of the BatScope software presented here.

After having imported the raw data, further processing is started by selecting all *sequences* contained in the project or specific *collection* and then starting the process *cutInspectClassify*. This is actually a predefined process flow, which detects calls, cuts them into sub-files, inspects them for call features and classifies them with all available classifiers.

The next step, which is the most critical step before exporting results to external analytical tools, is the user verification of the species classifications proposed by BatScope. Classifications based on biological signals (see section “Reference base”) inherently suffer from the selective variability of the reference data. We therefore do not recommend accepting BatScope’s species nominations without user verification, which may require just slight filtering for simple numeric values (e.g. drop faint signals), but could involve scrutinizing single sequences by comparing them acoustically (internally), or numerically (through external helper applications) to sound samples or values in the literature (Fig. 6).

• Figure 6 approx. here •

BatScope implementation

Data formats

Internally data are processed as WAV files sampled at 312.5 kHz with 16 bits. This allows for a frequency range of 0 - 156.25 kHz that can be analyzed, which is appropriate for covering all European and North American species, without using up too much disk space for oversampled data. The sampling rate is aligned with that of the Batlogger. Data collected at other rates will be automatically resampled to the internal rate. By this process, data sampled at higher rates will lose information contained above the Nyquist frequency of 156.25 kHz, and data sampled at lower frequency will basically be padded with zero values up to this frequency. It should be noted that data sampled at significantly lower rates (e.g. 192 kHz or even lower) might thus miss relevant information required for later species classification.

Meta-data can be provided as XML files (e.g. Batlogger) or suitably formatted CSV files, and are aligned with the *sequences*' data. Data importing, like most other tasks, is implemented through plugins (see below).

Appropriate vendor-specific plugins can be designed.

Reference base

Classifications rely on a reference base of 19'636 echolocation calls extracted from 633 echolocation call *sequences* recorded for 27 European bat species (Table 1). References thus include repeated measurements of single individuals so that the intra-individual variability of their calls can be accounted for. Recordings were mostly made with a Pettersson D980 detector (Pettersson Electronic AB, Sweden) connected to a PC-DAS 16/330 data acquisition card (Measurement Computing Corporation, USA) plugged into a laptop computer, or with a Batlogger (Elekon AG, Switzerland). The majority of the recordings resulted from hand-released animals captured with mist nets or harp traps. A few unidentified individuals were recorded in front of roosts inhabited by a single species after a few animals had been captured when leaving and identified. Recorded *sequences* ranged in duration from 3 to 20 s. The active recording was carefully monitored over headphones so that the whole variation in the *calls could be* captured, ranging from the short broadband calls shortly after takeoff, typically emitted in cluttered environment, to the long open-space echolocation calls, produced in free flight. Each individual was recorded for just one fly-by, resulting in one recorded *sequence*. All reference signals were recorded in Switzerland, southwestern Germany or northern Italy.

In general, the larger the reference base, the better the real world can be mapped. To illustrate this, we selected as a test only a quarter of the calls in our dataset as a subsample for training and verification in an experimental setup, and could show that in this case the classification quality dropped by almost 10% when compared to the training with the full dataset. The full dataset was used for training in the final version of the software. We have tried to improve quality by generating larger versioned reference bases, but do not update very frequently to ensure the comparability of the results is long-lasting and to avoid user frustration.

• Table 1 approx. here •

Analysis and classification methods

Data processing

Once data are imported into BatScope, the following processing steps are performed:

Detect and cut single echolocation calls

A detector algorithm searches for peaks of tonal signals. The start of a signal is detected when the following two characteristic measures within a moving window (256 data points) are fulfilled:

1. The 'standard deviation (SD) times the mean (MN)' of consecutive signal period durations (zero-crossings) is below a user configurable threshold (default = 8; $SD \times MN < 8$).
2. The 'standard deviation (SD) divided by the mean (MN)' of consecutive signal period durations (zero-crossing) is below a user configurable threshold (default = 0.2; $SD / MN < 0.2$).

Consecutive moving windows fulfilling both criteria define the raw signal. The peak position is determined by the highest energy value in the raw signal. The effective signal is cut 4096 data points before and after this peak in the candidate regions, leading to a cutout length of 8192 data points (equivalent to 26.214 ms duration), which are saved as single *call* WAV files on the file system. Only a part will be cut from signals of longer duration (e.g. Rhinolophidae) to avoid generating more calls than actually present in the recording. Cutting out only part of a call does not have negative consequences on later classification, as the constant frequency part of rhinolophid calls are very species specific.

Spectrogram generation

Spectrograms of cut out calls are calculated with 1024 point FFT and hop size of 50 data points (86% overlap). This results in 157 spectra per spectrogram, leading to a temporal resolution of 0.167 ms and a spectral resolution of 0.305 kHz.

Noise elimination – spectrogram filtering

Spectrogram data of calls must be filtered before feature extraction because recordings often contain noise like hissing, artefacts from bad microphones, nearby traffic or rivers, harmonic reflections, or other sources.

The first step is to estimate the average noise level in spectrograms of calls. All spectrogram values below the average are considered as noise. Searching for the local peak frequency starts in the temporal middle part of the spectrogram (Fig. 7 a). An additional high-pass filter prevents low frequency noise bands from being mistaken for the peak of a bat call. Using the peak identified as a first guess, the search is continued for the best peak in the neighbouring time slices. The stop criterion for finding the edge of a call in the spectrogram is distinctly configurable in time and frequency as well as going back or forth in time. If a peak value falls outside a

threshold band, a few time slices (holes) are ignored and searching continues. The highest energy peak is considered to be a local peak (target symbol in Fig. 7 A).

• Figure 7 approx. here •

Feature calculation

With the detection of the curved signal shape additional straightforward numerical features like “frequency of peak energy in filtered signal” can be created. Similarly, features with properties of directionality or shape (slope, curvature) and extent (duration, bandwidth) can be extracted.

Starting from the local peak position, the trajectory continues until the energy is below the average noise level (square dots in Fig. 7 B3). The extracted trajectory is further divided into discrete time bins to ensure that a constant number of features are generated (currently 5 bins are used, see the vertical dotted lines in Fig. 7 B3). For each time bin, advanced shape feature values such as average frequency or slope are retrieved (see Table 2).

• Table 2 approx. here •

Of the 61 signal features calculated, only 59 were considered for evaluation (Table 2). Because intervals to the previous and next *call* tended to be very inaccurate, e.g. if a weaker call was omitted, in bimodal distributions, or if a second bat was present in the recording, we ignored these values in the process. The remaining values contained temporal and spectral values, calculated in the raw and filtered spectrograms, as well as the values for bandwidth, slope and curvature. For these last values, we split each *call* into 5 time-equal bins and calculated central frequency, slope and curvature in each of the bins, in order to trace the *call* sweeping through frequency in time. Most of the temporal and spectral parameters are measured in a temporal or spectral energy sum curve, e.g. frequency, where 5%, 25%, 50%, 75% and 95% of the calls energy was reached. We proceeded similarly for duration and bandwidth features. Column ‘Explanation’ in Table 2 gives parameter measurement details. Fig. 8 represents these measurements graphically. In contrast to absolute values such as highest or lowest frequency, which can heavily fluctuate with S/N ratio or echo content, these energy-based values tend to be quite robust (Cortopassi 2006).

• Figure 8 approx. here •

Statistical approaches

We used the software R (R Core Team 2016) for the statistics and evaluated the classification quality of 59 parameters. The classifiers we concentrated on are those that seem, according to the literature, to be the most promising (Fernández-Delgado et al. 2014), namely Random Forest (RF), Support Vector Machines (SVM) and Neural Networks (NN). To enable reference to earlier versions of the software, we also included K Nearest Neighbours (KNN), Weighted K Nearest Neighbours (KKNN) as well as Quadratic Discriminant Analysis (QDA). Table 3 gives the R packages used for the classifications.

• Table 3 approx. here •

We first evaluated feature importance using a RF (column “Importance” in Table. 2), and then the performance of the other classifiers for the feature sets $F_6, \dots, F_{59}$, where F_n contains the n most important features. To assess classification accuracy, we computed confusion matrices using the function *confusionMatrix()* from the R-package “caret” (version 6.0-52). For the confusion matrix:

True positive (TP) False positive (FP; Type I error)

False negative (FN; Type II error) True negative (TN),

the following definitions apply:

Accuracy = (TP+TN) / (TP+TN+FP+FN). This is the percentage of all correct classifications relative to the total subjected to the test.

Sensitivity = TP / (TP+FN). This equals the proportion of correct predictions for a species, also termed True Positive Rate (TPR) or Correct Classification Rate (CCR).

The precision or positive predictive value (**PPV**) = $TP / (TP + FP)$ is thus the percentage of the positives that are truly correct.

(For an excellent explanation of the approach, see: https://en.wikipedia.org/wiki/Confusion_matrix)

In the following, "lowest sensitivity" and "lowest PPV" refer to the minimum sensitivity and PPV among all the species processed. Note that if a species is never predicted, $TP + FP = 0$ and its PPV is not defined. Only species for which the PPV is defined are used to compute the lowest PPV.

To compute confusion matrices on the level of sequences, predictions for the sequences are generated by majority vote among the predictions for the corresponding calls.

Feature importance varied considerably among the different species. We decided to use for all subsequent analyses the order of feature importance as determined for *Myotis blythii* (Tomes, 1857) (Myo_blyt; Fig. 9). For all trained algorithms this species was very difficult to identify.

• Figure 9 approx. here •

When we evaluated the effect of feature set size on the performance of KNN, RF and SVM, the accuracy, the lowest sensitivity and the lowest PPV invariably levelled out or even started to decrease when reaching around 40 parameters, which is why we decided to limit further evaluations to these 40 most influential variables (Fig. 10).

• Figure 10 approx. here •

Parameter tuning

The parameters tuned are listed in Table 4 and the values that were chosen for parameters not subjected to tuning are given. For QDA, we first transformed the data by principal component analysis (PCA) to have sufficient variance within species. We then treated the number of principal components retained (nPC) as a tuning parameter. Tuning curves for parameters of all 6 classifiers are given in Supplements Fig. S3.

• Table 4 approx. here •

Final classification performance

Averaging classification results over all species is only one facet of overall quality when assessing the quality of classifiers. We were additionally interested in the values for the species classified least successfully, which can be rather low for some groups, e.g. the genus *Myotis*. To compare the final classification accuracies for the six classifiers we were therefore most interested in obtaining the highest accuracy, but also tried the lowest PPV and lowest sensitivity. The larger the minimum PPV and sensitivities are, the higher are the classification rates reached in the worst classifications.

For single calls, averaged over five runs, accuracy ranged from 72% (QDA) to 82% (RF). On the level of sequences, again averaged over five runs, accuracy ranged from 77% (QDA) to 89% (SVM). RF and SVM,

closely followed by NN, generally outcompete the other classifiers (Table 5). We cross-validated on the level of sequences to take into account the inter-dependence among calls in a single sequence (= same individual bat). Neglecting this artificially improved the classification accuracy by increasing the estimates for the accuracy and lowest PPV.

• Table 5 approx. here •

Classifier consensus

We tried to reduce the processing time by reducing the number of classifiers required and comparing this version with older versions of BatScope, which only use QDA, KNN and SVM. We tested combinations of triplets of the six classifiers to see how they increased classification accuracy when combined (Table 6). The combinations only marginally increased the average classification success of the single best classifier (SVM, Table 5), at the cost of decreasing lowest PPV and sensitivity. By requiring all three classifiers to agree, the accuracy and lowest PPV can be slightly boosted, but the lowest sensitivity drops further and 3-6% of the data cannot be classified any longer (Table 6, assignable < 100%).

In preliminary trials, asking for agreement amongst all six classifiers boosts the successful classification rate to 96%, at considerable cost since 24% of all sequences are no longer classifiable.

• Table 6 approx. here •

Quality management

Probabilistic recognition

The six classifiers mentioned above are implemented with the corresponding tuned models in BatScope4. Each call is classified according to all classifiers selected by the user. Each classifier returns the probability of a species' match down to a user selectable threshold. The number of votes and the confidence of the classifications are summed up for every assumed species and summarized per call. Classifiers may not agree on the species (Fig. 4, lower right), but all contribute to the final vote.

Call classifier results are visible in BatScope's GUI in the tabular view of calls (Fig. 4), where general call properties, summary statistics and the detailed results of classifications are also listed.

As an additional control, the values measured for the call duration, frequency of peak energy and the bandwidth are tested to see if they fall within the 95% confidence interval of the respective values of the assumed species (95% CI-Test; failed for the highlighted call in Fig. 4). Finally, all the votes for single calls are summarized over the sequence and ranked for absolute number of calls assigned, relative frequency and confidence. If the *sequence* data contain meta-information regarding recording location, the species will also be tested against its known distribution, user selectable for Switzerland or Europe, and the result indicated as pass (within range; Fig. 3, lower left). Alternatively, the distance is indicated to the calculated range border (alpha-shaped hull of European or Swiss distribution; Supplements Fig. S2).

Expert verification of classifications

Users can now decide if they want to accept the species proposed by the software by double-clicking on the species' name. Verified species will be given a rank. If the users assign the same rank to two or more species, the result is considered as one entry of a species complex or aggregate, and not as two separate species present in the recording.

The verification process also allows a quick switch to the taxonomic view of the particular species so that the exemplary call spectrograms can be visually checked and compared with literature values and the reference calls be played back acoustically. The calls may even be played back in parallel with the sequence under verification, thus allowing the human ear to judge the degree of pattern matching.

The single results of classifiers cannot be edited. However, users may decide to only run specific classifiers over their data. Alternatively, they can exclude single calls from the summary sequence statistics (Fig. 4) by *deactivating* these calls, e.g. those that failed the CI-test, have a low S/N ratio or some other improbable feature such as a large bandwidth AND long duration.

User process automation

For larger surveys it is not realistic to go through each *sequence* for verification. Thus, most user tasks can be saved for later storage or exchange. Data filters containing any combination of search criteria on the level of *sequences* or/and *calls* combined can be graphically compiled and then saved as XML files (Fig. 11). Similarly, calls that are e.g. dubious or noisy can be deactivated, or the species assignment of clear sequences verified automatically using a process builder interface (Fig. 11).

- Figure 11 approx. here •

Discussion

Applicability of BatScope4

At the time of writing, over 200 users are registered for BatScope3. The program is highly esteemed, but further distribution is hindered by the limited availability of the platform (macOS). This, and the highly interdependent program structure, led us to tackle a rewrite of the code.

Comparability – how does BatScope4 compare with other programs

Several commercial products for classifying bat species according to their echolocation calls are available (for details see e.g. Echolocation Handbook 2017). In general, most of these do not fully explain the parameters they measure, let alone how they calculate them and the associated statistics. If calls from bats or sequences that contributed calls to the training base are also used for testing, the results will be biased by autocorrelation through the recording, or individual bat. In BatScope4 we avoided this error in our accuracy measurements. Checking the accuracy of available products for classifying bat calls independently is very important (Rydell et al. 2017) to enable better comparisons of published studies on species occurrence based on automatic bat species identification. Products may rely on very different species compositions, call selections, measured parameters and applied algorithms, for example. Testing based on a comparable standard dataset (e.g. EchoBank, Walters et al. 2012) would make it easier to compare different technical approaches better.

Comparing classification results from data based on different species compositions is only moderately useful as some species may be more similar to others and may or may not be contained in a specific set. This can then drastically affect the average rates for correct classification. A 'gold' standard for echolocation samples as calibration data should therefore be introduced for future tests.

Our results do agree with other authors across continents on the species groups that are difficult to distinguish: Some species of the genus *Myotis* remain tricky whatever we try. In Europe, species in the genus *Eptesicus*, *Nyctalus* and *Vespertilio* can be confusing, and the genus *Plecotus* is another difficult group to identify, especially as their faint calls are difficult to record in the first place. A quick overview of a few publications (Parsons and Jones 2000; Britzke et al. 2002; Russo and Jones 2002; Fukui et al. 2004; Obrist et al. 2004; Jennings et al. 2008; Britzke et al. 2011; Agranat 2012; Walters et al. 2012; Rodriguez-San Pedro and Simonetti 2013; Henriquez et al. 2014; Wordley et al. 2014) shows a general trend of slightly decreasing correct

classification rate (CCR = TPR = sensitivity) with increasing number of species in the pool. However, if the correct classification rate for the species most difficult to identify (Min % correct) is considered, the rate decreases much more sharply with increasing number of species in the pool (Table 7). The chance of confusing species obviously increases the more species there are. On the other hand, increasing sample sizes of both N calls and N sequences, does increase both the average and the minimum recognition rates. Additional relevant publications are summarized in Henríquez et al. (2014) in Table 1.

• Table 7 approx. here •

In agreement with other sources (Armitage and Ober 2010; Fernández-Delgado et al. 2014), We found that some classifiers *per se* (e.g. KNN, QDA) perform on average clearly less well than others. Interestingly, triplets of classifiers containing either or both of KNN and QDA ranged in the top six combinations in our study, indicating that KNN and QDA seem to improve classifications of different species much more than the remaining classifiers. Alternatively, hierarchies of classifiers could be used to improve classification rates (Redgwell et al. 2009), but this approach was not tested so far and is thus not implemented in BatScope.

Selecting calls for training libraries is tricky, as Clement et al. (2014) pointed out: any automatic or manual pre-filtering of calls previous to training a classifier can influence not only average parameter measurements but cast its shadow as far as affecting comparability between classification accuracies. We did not select or filter out any calls to avoid this problem completely. Nevertheless, their argument may still hold, as, tested on a completely independent training set (separated in space and time), overall performance can drop. We had hoped to be able to test BatScope4 on data from EchoBank (Walters et al. 2012), to which we had contributed our own data, as we do not have our own second comprehensive training set. Unfortunately however, we have until now been refused access!

We provide scientists and other enthusiasts with access to BatScope as an interactive tool allowing human-assisted control of machine-generated results, and have successfully used it in our own research (Frey-Ehrenbold et al. 2013; Bohnenstengel et al. 2014; Froidevaux et al. 2014; Ravessoud 2017). BatScope, we hope, resolves the dilemma of ‘human vs. machine’ (Jennings et al. 2008) and is more within a ‘human-controls machine’ paradigm.

Chances and pitfalls

Speed

The speed of data processing largely depends on the number of computer-processing cores available, and that of all file-based operations on the speed of the underlying data-storage medium, hard disk or solid-state drive (SSD). A survey on flight corridors (0.64 Mio *sequences*, 11 Mio *calls*) monopolized a 4 core Intel Xeon 5100 MacPro (2007) for over nine weeks to calculate classifications, thus processing one *sequence* per core every 35 seconds. Modern machines can benefit fully from their multi-core architecture, as BatScope multithreads all relevant processes, and faster storage systems further increase processing speed. With a recent system equipped with a SSD and a 4 core Intel Core i7 (processing 8 threads in parallel; Apple Model iMac16,2), we can process data in 70% of the time which the recording duration spanned, thus perform better than real time.

Data volume

A recording of 10 seconds duration sampled with 312.5 kHz at 16 bit results in a wav file of 6.25 MB. Each cut out call with associated spectrograms contains 100 KB of data. Extrapolated onto 100'000 recorded sequences containing on average 20 calls each results in a data space requirement of roughly 1 TB. The large amount of storage space for raw data, processed data and a backup thus quickly becomes a matter of concern that needs to be treated seriously in larger projects.

Quality

A reference library of bat echolocation calls can be compiled in several ways. Recording bats flying in a tent will, depending on tent size and bat species, generate recordings that are of limited value for recognizing free-flying bats, but calls emitted after hand-release have similar characteristics in the first take-off phase before slowly becoming more similar to recordings of free-flying bats. By controlling the recordings through headphones, we were able to avoid recording these very first, atypical calls and extend the recording until the bats had reached normal foraging heights. Releasing the bats a few hundred meters away from the capture site also helped as the bats stayed around reorienting for sufficiently long recording times.

We did not select single echolocation calls for training, but used all non-take-off calls of a given recording sequence to cover the full variability of individual echolocation. This can be seen as pseudo-replication in the variable differentiation of species or individuals, and should be taken into consideration in such analyses, e.g. by nesting correspondingly. However, it is less of a problem when training classifiers for optimal separation of

species, as long as individuals (whole recording sequences) are tested later completely independently of the training process. This was handled during the measurement of classifier accuracy.

Given a specific training base, the quality of classifications depends very much on the quality of the initial recordings. Over time, we included calls in our training base recorded with different digital devices that also differ with respect to e.g. type of microphone, amplification or S/N ratio. Although this increases the variability of recordings in the database, it also results in better coverage of the possible variability of data fed into the system by different user profiles. Classifiers will have less success in *sequences* with low S/N ratios, as fewer calls will be cut and measurements (e.g. high frequencies) will be truncated. Users may deal with some of this problem by e.g. excluding calls of low S/N ratio or low classification quality from summary statistics, or increasing the number of classifiers required to agree on the classification. However, this last approach will also mean excluding some of the recordings from assessment (see also: Classifier consensus, Table 6).

Many species occupy similar foraging niches and echolocate with very similar calls. In some cases, such species may be locally separated by their geographic distribution (e.g. *Myotis daubentonii* (Kuhl, 1819), *Myotis capaccinii* (Bonaparte, 1837) and *Myotis dasycneme* (Boie, 1825)). The matching of geographical distribution ranges with the GPS positioning of the recording location then helps in these cases to improve classification accuracy. Similar improvement could, of course, also be made by compiling regional reference bases. This would, however, require cumbersome checking of the reference bases and recording locations, which is more appropriate for computer than human processing.

The implementation of this location check in BatScope4 is only meant as additional information for the verifying person. Restricting classifications to the known extent of occurrence would potentially limit how well changes in the ranges of species distributions, e.g. due to climate change effects, could be detected.

Effort for verification

The verification of a *sequence* takes on average 20-30 seconds if identification to the best possible taxonomic level is pursued. This has been evaluated with qualified users on dataset containing standard surveys in the Swiss lowlands. The cost of verifying 10'000 recordings with a salary of \$50/h amounts to over \$4000, and is therefore of major concern if finances are tight. Several approaches can be taken to reduce these costs when using BatScope. Filtering *sequences* to include those with a high number of *calls*, high agreement of classifiers and high occurrence of a single species speeds up verification, but at the cost of filtering out some data. If only

certain types of echolocators are of interest, filtering for specific call properties or taxonomic filtering for genus will also reduce verifying efforts.

Calls may, however, often be classified correctly as originating from different species, as more than one bat may be recorded at the same time. The chances of recording several bats in a single *sequence* increase with the duration of the recording *sequences*. Greatly shortening recording time will necessarily lead to degraded summary statistics in BatScope, and result in more cases of ‘human vs. machine’ (Jennings et al. 2008). Thus, as a general rule of thumb, we recommend to setting the longest recording duration of *sequences* to 10-20 seconds. If a pre-trigger of 0.5 s and a post-trigger of 1 s can be kept, (e.g. on Batloggers), the chances of obtaining a minimum amount of useful data for analyses increase.

Future

Filters and processes

BatScope’s user-adaptable filter and process-building tools (Fig. 7) are welcome instruments to help users compile their own repeatable workflows. They can streamline the most tedious tasks to be performed automatically and concentrate on the most difficult recordings. Filters and processes can be stored as XML-files, which makes them easy to share. We can provide presets for standard tasks, such as cleaning out noisy data or verifying sequences. As the filters are self-documenting, they may become standards in working with bat recordings. We therefore also encourage users to share their filters and processes.

Potential for regionalization

Currently BatScope includes a training database from recordings made in Central Europe, which does not contain some species restricted to Southern Europe. Wordley et al. (2014) highlighted the need for regional training bases to cover within species variation. The variability of bat species across the continent has not yet been covered sufficiently in BatScope, although we do not know about the actual degree of variability of species’ call parameters across regions. Obtaining more references from other regions for training will be an essential point as will be testing the analyses against independent recordings.

A regionally diversified training base would also allow us to decide if regional classifiers should be trained, or if matching recording locations against species distribution ranges is sufficient for improving classification quality. Ideally, the locations of reference recordings should also be part of the training process, which would add additional dimensions to the parameter set.

Novel approaches

Algorithms applied to acoustic species identification have evolved from explicit decision trees implemented in analog keys, over discriminant functions and neural networks to random forest and support vector machines. The same species or phonotypic clusters still, however, remain difficult to classify. With the advent of novel pattern recognition approaches and ‘deep learning’ algorithms, we may see further improvements in classification accuracy. What invariably improves classification algorithms is including more data, particularly species-accurate data. This is why we are keen to support the call for shared libraries of reference calls for training algorithms, so that data can be shared in the spirit of ‘open data’ initiatives.

Availability, platform and versioning

BatScope3 is presently freely available for macOS from <http://www.BatScope.ch>. BatScope4 will be freely available for macOS later in 2018, and for Linux and Windows in 2019.

Acknowledgements

Both authors contributed equally to this work, MKO focused on the biological, RB on the technical aspects. We wish to express our thanks to the many roost owners who allowed us to access ‘their bats’. Peter F. Flückiger was a key person during the reference recordings. We are grateful to Vlad Trifa, Jonas Honegger, Stefan Frauenfelder, Roman Vetter, Andreas Pasternak, Robin Oster, Stefan Dietiker, Raffael Theiler, Daniel Hegglin and Thomas Debrunner for help with their programming, and Nicolas Blöchliger for evaluating the reference bases and optimizing the classifier settings. We thank Oliver Probst and Moritz Küttel for finalizing BatScope4, and Annie Frey-Ehrenbold, Fabio Bontadina, Hans-Peter Stutz, Hubert Krättli, Thomas Sattler, Raul Rodriguez, Peter Zingg, Emmanuel Rey, Elias Bader, Martin Decurtins, Carsten Braun, and the many other registered users of BatScope for testing it and critically commenting on the different versions. Finally, we thank Silvia Dingwall for revising the English. Distributional data were either derived from GBIF website or kindly provided by the Swiss Biological Records Center (CSCF/SZKF). This work was partly financed by an internal grant of WSL to MKO.

References

- Agranat, I. 2012. Bat Species Identification from Zero Crossing and Full Spectrum Echolocation Calls using HMMs, Fisher Scores, Unsupervised Clustering and Balanced Winnow Pairwise Classifiers. Available from <http://condor.wildlifeacoustics.com/batid.pdf> [accessed 20.07.].
- Ahlén, I. 1981. Identification of Scandinavian bats by their sounds. The Swedish University of Agricultural Sciences. Department of Wildlife Ecology. Report 6. p. 56.
- Armitage, D.W., and Ober, H.K. 2010. A comparison of supervised learning techniques in the classification of bat echolocation calls. *Ecol. Inform.* **5**(6): 465-473.
- Arnold, B.D., and Wilkinson, G.S. 2011. Individual specific contact calls of pallid bats (*Antrozous pallidus*) attract conspecifics at roosting sites. *Behav. Ecol. Sociobiol.* **65**(8): 1581-1593. doi: 10.1007/s00265-011-1168-4.
- Blehert, D.S., Hicks, A.C., Behr, M., Meteyer, C.U., Berlowski-Zier, B.M., Buckles, E.L., Coleman, J.T., Darling, S.R., Gargas, A., and Niver, R. 2009. Bat white-nose syndrome: an emerging fungal pathogen? *Science*, **323**(5911): 227-227.
- Boesch, R., and Obrist, M.K. 2013. BatScope - Implementation of a Bioacoustic Taxon Identification Tool. Available from <http://www.batscope.ch> [accessed 12.01.2018].
- Bohnenstengel, T., Krättli, H., Obrist, M.K., Bontadina, F., Jaberg, C., Ruedi, M., and Moeschler, P. 2014. Rote Liste Fledermäuse. Gefährdete Arten der Schweiz, Stand 2011. Bundesamt für Umwelt, Bern; Centre de Coordination Ouest pour l'étude et la protection des chauves-souris, Genève; Koordinationsstelle Ost für Fledermausschutz, Zürich; Schweizer Zentrum für die Kartografie der Fauna, Neuenburg; Eidgenössische Forschungsanstalt für Wald, Schnee und Landschaft, Birmensdorf. pp. 95.
- Brigham, R.M., Kalko, E.K.V., Jones, G., Parsons, S., and Limpens, H.J.G.A. (Editors). 2004. Bat Echolocation Research. Tools, Techniques and Analysis. Bat Conservation International, Austin TX.
- Britzke, E.R., Duchamp, J.E., Murray, K.L., Swihart, R.K., and Robbins, L.W. 2011. Acoustic identification of bats in the eastern United States: A comparison of parametric and nonparametric methods. *J. Wildl. Manage.* **75**(3): 660-667. doi: 10.1002/jwmg.68.

- Britzke, E.R., Murray, K.L., Heywood, J.S., Robbins, L.W., Kurta, A., and Kennedy, J. 2002. Acoustic identification. In *The Indiana bat: biology and management of an endangered species*. Bat Conservation International, Austin, TX. Edited by A. Kurta and J. Kennedy. pp. 221-225.
- Burnett, S.C. 2001. Individual variation in the echolocation calls of big brown bats (*Eptesicus fuscus*) and their potential for acoustic identification and censusing. Ph.D. Dissertation. The Ohio State University. pp. 168.
- Clement, M.J., Murray, K.L., Solick, D.I., and Gruver, J.C. 2014. The effect of call libraries and acoustic filters on the identification of bat echolocation. *Ecol. Evol.* **4**(17): 3482-3493. doi: 10.1002/ece3.1201.
- Cortopassi, K.A. 2006. Automated and Robust Measurement of Signal Features. Available from <http://www.birds.cornell.edu/brp/research/algorithm/automated-and-robust-measurement-of-signal-features/> [accessed 03.08.2015].
- ecoObs. 2017. Batcorder. Available from <http://www.ecoobs.com/cnt-batcorder.html> [accessed 12.01.2018].
- Elekon AG. 2017. BatLogger. Available from <http://www.batlogger.com/en> [accessed 12.01.2018].
- Fenton, M.B., Merriam, H.G., and Holroyd, G.L. 1983. Bats of Kootenay, Glacier, and Mount-Revelstoke National-Parks in Canada - Identification by Echolocation Calls, Distribution, and Biology. *Can. J. Zool.* **61**(11): 2503-2508.
- Fenton, M.B., Racey, P., and Rayner, J.M.V. 1987. Recent advances in the study of bats. University Press, Cambridge. pp. 470.
- Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. 2014. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **15**(1): 3133-3181.
- Frey-Ehrenbold, A., Bontadina, F., Arlettaz, R., and Obrist, M.K. 2013. Landscape connectivity, habitat structure and activity of bat guilds in farmland-dominated matrices. *J. Appl. Ecol.* **50**(1): 252-261.
- Froidevaux, J.S., Zellweger, F., Bollmann, K., and Obrist, M.K. 2014. Optimizing passive acoustic sampling of bats in forests. *Ecol. Evol.* **4**(24): 4690-4700.
- Froidevaux, J.S.P., Zellweger, F., Bollmann, K., Jones, G., and Obrist, M.K. 2016. From field surveys to LiDAR: Shining a light on how bats respond to forest structure. *Remote Sens. Environ.* **175**: 242-250. doi: 10.1016/j.rse.2015.12.038.

- Fukui, D., Agetsuma, N., and Hill, D.A. 2004. Acoustic identification of eight species of bat (Mammalia : Chiroptera) inhabiting forests of southern Hokkaido, Japan: Potential for conservation monitoring. *Zool. Sci.* **21**(9): 947-955.
- Griffin, D.R. 1958. Listening in the Dark. The Acoustic Orientation of Bats and Men. Yale University Press, New Haven. (1986 reprint by Cornell University Press, Ithaca, New York). pp. 415.
- Henríquez, A., Alonso, J.B., Travieso, C.M., Rodríguez-Herrera, B., Bolaños, F., Alpizar, P., López-de-Ipina, K., and Henríquez, P. 2014. An automatic acoustic bat identification system based on the audible spectrum. *Expert. Syst. Appl.* **41**(11): 5451-5465. doi: 10.1016/j.eswa.2014.02.021.
- Jennings, N., Parsons, S., and Pocock, M.J.O. 2008. Human vs. machine: identification of bat species from their echolocation calls by humans and by artificial neural networks. *Can. J. Zool.* **86**: 371-377.
- Kazial, K.A., Kenny, T.L., and Burnett, S.C. 2008. Little brown bats (*Myotis lucifugus*) recognize individual identity of conspecifics using sonar calls. *Ethology*, **114**(5): 469-478.
- Masters, W.M., Raver, K.A.S., and Kazial, K.A. 1995. Sonar signals of big brown bats, *Eptesicus fuscus*, contain information about individual identity, age and family affiliation. *Anim. Behav.* **50**(Part 5): 1243-1260.
- O'Shea, T.J., Cryan, P.M., Hayman, D.T.S., Plowright, R.K., and Streicker, D.G. 2016. Multiple mortality events in bats: a global review. *Mamm. Rev.* **46**(3): 175-190. doi: 10.1111/mam.12064.
- Obrist, M.K. 1995. Flexible bat echolocation: the influence of individual, habitat and conspecifics on sonar signal design. *Behav. Ecol. Sociobiol.* **36**(3): 207-219.
- Obrist, M.K., Boesch, R., and Flückiger, P.F. 2004. Variability in echolocation call design of 26 Swiss bat species: consequences, limits and options for automated field identification with a synergetic pattern recognition approach. *Mammalia*, **68**(4): 307-322.
- Obrist, M.K., Boesch, R., and Flückiger, P.F. 2008. Probabilistic evaluation of synergetic ultrasound pattern recognition for large scale bat surveys. *In* International Expert meeting on IT-based detection of bioacoustical pattern. 7.-12.12.2007. *Edited by* K.-H. Frommolt and R. Bardeli and M. Clausen. Federal Agency for Nature Conservation, International Academy for Nature Conservation (INA) Isle of Vilm, Germany. pp. 29-42.
- Obrist, M.K., Pavan, G., Sueur, J., Riede, K., Llusia, D., and Márquez, R. 2010. Bioacoustics approaches in biodiversity inventories. *In* Manual on field recording techniques and protocols for All

- Taxa Biodiversity Inventories and Monitoring. Edited by J. Eymann and J.r.m. Degreeef and C.L. Häuser and J.C. Monje and Y. Samyn and D. VandenSpiegel. pp. 68-99.
- Obrist, M.K., Rathey, E., Bontadina, F., Martinoli, A., Conedera, M., Christe, P., and Moretti, M. 2011. Response of bat species to sylvo-pastoral abandonment. *Forest Ecol. Manag.* **261**(3): 789-798. doi: 10.1016/j.foreco.2010.12.010.
- Parsons, S., and Jones, G. 2000. Acoustic identification of twelve species of echolocating bat by discriminant function analysis and artificial neural networks. *J. Exp. Biol.* **203**(17): 2641-2656.
- Parsons, S., and Obrist, M.K. 2004. Recent methodological advances in the recording and analysis of chiropteran biosonar signals in the field. In *Echolocation in Bats and Dolphins*, Proceedings of the Biosonar Conference 1998. Edited by J. Thomas and C. Moss and M. Vater. University of Chicago Press, Chicago. pp. 468-477.
- Pearl, D.L., and Fenton, M.B. 1996. Can echolocation calls provide information about group identity in the little brown bat (*Myotis lucifugus*). *Can. J. Zool.* **74**(12): 2184-2192.
- Pettersson. 2017. Pettersson Elektronik. Available from <http://www.batsound.com/?p=10> [accessed 12.01.2018].
- PostgreSQL. 2017. PostgreSQL Database Management System. PostgreSQL Global Development Group.
- R Core Team. 2016. R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria.
- Ravessoud, T. 2017. Finding a method to predict the commuting activity of bats. Masters Thesis. Ecology and Evolution Department, University of Lausanne. pp. 58.
- Redgwell, R.D., Szewczak, J.M., Jones, G., and Parsons, S. 2009. Classification of Echolocation Calls from 14 Species of Bat by Support Vector Machines and Ensembles of Neural Networks. *Algorithms* **2**(3): 907-924. doi: 10.3390/a2030907.
- Rodriguez-San Pedro, A., and Simonetti, J.A. 2013. Acoustic identification of four species of bats (Order Chiroptera) in central Chile. *Bioacoustics*, **22**(2): 165-172. doi: 10.1080/09524622.2013.763384.
- Russo, D., and Jones, G. 2002. Identification of twenty-two bat species (Mammalia: Chiroptera) from Italy by analysis of time-expanded recordings of echolocation calls. *J. Zool.* **258**: 91-103.

- Russo, D., and Voigt, C.C. 2016. The use of automated identification of bat echolocation calls in acoustic monitoring: A cautionary note for a sound analysis. *Ecol. Indic.* **66**: 598-602.
- Rydell, J., Nyman, S., Eklöf, J., Jones, G., and Russo, D. 2017. Testing the performances of automated identification of bat echolocation calls: A request for prudence. *Ecol. Indic.* **78**: 416-420. doi: 10.1016/j.ecolind.2017.03.023.
- Siemers, B.M., and Kerth, G. 2006. Do echolocation calls of wild colony-living Bechstein's bat (*Myotis bechsteinii*) provide individual-specific signatures? *Behav. Ecol. Sociobiol.* **59**(3): 443-454.
- Urbanek, S. 2013. Rserve: Binary R server. R package version 1.7-3., Vienna, Austria.
- USFWS, United States Fish and Wildlife S. 2017. White-nose Syndrome.org: North America's Response to the Devastating Bat Disease. Available from <http://whitenosesyndrome.org/> [accessed 08.03.2017].
- Vaughan, N., Jones, G., and Harris, S. 1997. Identification of British bat species by multivariate analysis of echolocation call parameters. *Bioacoustics*, **7**: 189-207.
- Walters, C.L., Freeman, R., Collen, A., Dietz, C., Brock Fenton, M., Jones, G., Obrist, M.K., Puechmaille, S.J., Sattler, T., Siemers, B.M., Parsons, S., and Jones, K.E. 2012. A continental-scale tool for acoustic identification of European bats. *J. Appl. Ecol.* **49**: 1064-1074. doi: 10.1111/j.1365-2664.2012.02182.x.
- Wildlife Acoustics. 2017. Bioacoustic monitoring systems. Available from <https://www.wildlifeacoustics.com/products/> [accessed 20.03.2017].
- Wordley, C.F.R., Foui, E.K., Mudappa, D., Sankaran, M., and Altringham, J.D. 2014. Acoustic Identification of Bats in the Southern Western Ghats, India. *Acta. Chiropterologica*. **16**(1): 213-222. doi: 10.3161/150811014x683408.
- Yovel, Y., Melcon, M.L., Franz, M.O., Denzinger, A., and Schnitzler, H.U. 2009. The Voice of Bats: How Greater Mouse-eared Bats Recognize Individuals Based on Their Echolocation Calls. *PLOS Comput. Biol.* **5**(6): e1000400. doi: 10.1371/journal.pcbi.1000400.
- Zingg, P.E. 1990. Akustische Artidentifikation von Fledermäusen (Mammalia: Chiroptera) in der Schweiz. *Rev. Suisse. Zool.* **97**(2): 263-294.

Figure captions

Figure 1: Structural design

Hierarchy of database tables and its contents. See text for details.

Figure 2: Projects and collections

First hierarchical organizational level for the user. Projects contain collections with properties such as name and creator.

Figure 3: Sequences

Relevant information for a list of sequences is shown here with filter criteria. More details on the selected sequence are then displayed in the lower panel. Alternatively, the lower panel could show the recording location and selected species' distribution boundaries (Fig. S2, Supplements). In the sequence shown here, a *Pipistrellus pipistrellus* (Schreber, 1774) was verified.

Figure 4: Calls

Calls can be displayed as lists and in detailed view, with a visual representation of a single call spectrogram, filtered call spectrogram, and a sketch of the binned parameters, the general call properties, summary statistics and the results of the classification also listed, down to the output of single classifiers. For the highlighted call shown in detail in the lower panel only two classifiers agree on species and the CI-test (see "Quality management") failed. Thus, the operator chose to deactivate the call, and is now reminded to recalculate the summary sequence statistics (button appearing on top).

Figure 5: Taxonomy

Taxonomy shows representative calls of the species, allows sequences to be played back and gives information about the literature and web-views relevant to the species.

Figure 6: Processing and data workflow

Framework of the processing workflow: Data are first imported into collections (1), single signals are then cut and stored as *calls* (2), the data are inspected for parameters (3), and classified to species in R (4). After user verification (e.g. using Raven; 5), the results can be exported (6).

Figure 7: Spectrogram filtering and bin creation

A) Process of filtering spectrograms from peak (target symbol) up left and down right along the spectrogram until break criteria have been reached (green arrows). The start and end of the call can be reliably found (black vertical bars) even in the presence of neighbouring calls. B) The original spectrogram is shown on the left (B1), the filtered spectrogram in the middle and (B2) and the splitting of a call into bins for parameter extraction is visualized on the right (B3).

Figure 8: Parameter measurements

Graphical representation of the parameters measured in each call. Abbreviations are given in Table 2.

Figure 9: Feature importance

All 59 features, sorted according to the importance for *Myotis blythii* (Tomes, 1857) (Myo_blyt). Three more species are superimposed on the plot, *Myotis mystacinus* (Kuhl, 1819), *Myotis bechsteinii* (Kuhl, 1818) and *Tadarida teniotis* Rafinesque, 1814, and the mean decrease in accuracy is given. Features within the red frame were considered further for classification.

Figure 10: Effects of feature-set size

Effect of feature-set size on the call classification performance of KNN (5 runs), RF (5 runs) and SVM (1 run) and on accuracy, lowest sensitivity and lowest PPV are shown. Only the 40 parameters in the non-shaded area were considered further (Importance > 0.01 in Table 2).

Figure 11: User adaptable filter and process building tools

For data post-processing workflows such as filtering (top) or processes such as verification (bottom), building tools allow presets to be compiled and stored for later re-use or exchange.

Figure 1: Structural design

<i>Database tables</i>				
<i>Projects</i>	<i>Collections</i>	<i>Sequences</i>	<i>Calls</i>	<i>Classifications</i>
Project 1	Collection 1	Sequence 1	Call 1	Classification 1
				Classification 2
				Classification n
		Sequence 2	Call 1	Classification 1
				Classification 2
				Classification n
	Collection 2	Sequence 1	Call 1	Classification 1
				Classification 2
				Classification n
		Sequence 2	Call 2	Classification 1
				Classification 2
				Classification n
Project 2	...	...	...	...

Figure 2: Projects and collections

Projects

Sequences

Calls

Taxonomy

Verifying as Martin Obrist

+ Create new Project

Delete

Edit Project

Detach Project

Snapshot Project

Attach from File

All	None	Name	Creator	Creation Date	Path	Range Base	Detac...
▶	☐	Europe	Martin Obrist	15. Mär 2017 10:31	/Volumes/SSDtresor/TestDaten_BS_3-4/BS4/Europe.bs4prj	CH Alphahull - WSL	false
▶	☐	Overseas	Martin Obrist	15. Mär 2017 10:31	/Volumes/SSDtresor/TestDaten_BS_3-4/BS4/Overseas.bs4prj	CH Alphahull - WSL	false
▼	☒	Switzerland FleFluKorr	Thomas Ravessoud, Martin O...	15. Mär 2017 10:31	/Volumes/SSDtresor/TestDaten_BS_3-4/BS4/Switzerland FleFl...	CH Alphahull - WSL	false
	☒	RHI_21_Gisw_Rep-1_B-14_2016-06-22_BATLOG-1004		15. Mär 2017 10:31			
	☒	RHI_21_Gisw_Rep-1_B-12_2016-06-22_BATLOG-1014		15. Mär 2017 10:31			
	☐	MYM_01_Eysi_Rep-1_A-29_2016-06-17_BATLOG-1007		15. Mär 2017 10:31			
	☐	MYM_01_Eysi_Rep-1_A-28_2016-06-17_BATLOG-1029		15. Mär 2017 10:31			
▶	☐	Switzerland FoBaDiv	Elias Bader, Martin Obrist	15. Mär 2017 10:31	/Volumes/SSDtresor/TestDaten_BS_3-4/BS4/Switzerland FoBa...	CH Alphahull - WSL	false
▶	☐	Switzerland FoBaIn	Jeremy Froidevaux, Martin Ob...	15. Mär 2017 10:31	/Volumes/SSDtresor/TestDaten_BS_3-4/BS4/Switzerland FoBa...	CH Alphahull - WSL	false
▼	☒	Switzerland LaCoBa	Annie Frey-Ehrenbold, Martin ...	15. Mär 2017 10:31	/Volumes/SSDtresor/TestDaten_BS_3-4/BS4/Switzerland LaCo...	CH Alphahull - WSL	false
	☒	SD_Karte s14_13_1010		15. Mär 2017 10:31			
	☒	SD_Karte s14_08_1014		15. Mär 2017 10:31			
	☒	SD_Karte s12_15_1005		15. Mär 2017 10:31			
	☒	SD_Karte s12_03_1019		15. Mär 2017 10:31			
	☒	SD_Karte s06_11_1005		15. Mär 2017 10:31			
	☒	SD_Karte s06_03_1003		15. Mär 2017 10:31			
▼	☒	Switzerland TdA	Martin Obrist	15. Mär 2017 10:31	/Volumes/SSDtresor/TestDaten_BS_3-4/BS4/Switzerland TdA....	CH Alphahull - WSL	false
	☐	TdA Wohlen 2011-05-20 BATLOG_1010 F Torfmoos Ost		15. Mär 2017 10:31			
	☐	...		15. Mär 2017 10:31			
	☒	TdA Suhr 2013-06-14 3 Weiher Waldrand BATLOG_1013		15. Mär 2017 10:31			
	☐	TdA Möhlin 2012-06-15 BATLOG_1010 A Sturmfläche Röti		15. Mär 2017 10:31			
▶	☐	BatCorder	Martin Obrist	15. Mär 2017 10:31	/Volumes/SSDtresor/TestDaten_BS_3-4/BS4/BatCorder.bs4prj	CH Alphahull - WSL	false

Figure 3: Sequences

Figure 4: Calls

Figure 5: Taxonomy

Figure 6: Processing and data workflow

Figure 7: Spectrogram filtering and bin creation

Figure 8: Parameter measurements

59 Parameters measured by BatScope in each call
(some in filtered AND unfiltered spectrogram)

Figure 9: Feature importance

Figure 10: Effects of feature-set size

Figure 11: User adaptable filter and process building tools

1 **Tables**

2 **Table 1: Species**

3 List of species contained in the reference base, as well as the number of recorded *sequences* and
4 *calls* that served for training the classifiers.

Family	Species	N	
		N sequences	calls
Rhinolophidae	<i>Rhinolophus ferrumequinum</i>	21	2642
	<i>Rhinolophus hipposideros</i>	13	530
Vespertilionidae	<i>Barbastella barbastellus</i>	20	434
	<i>Eptesicus nilssonii</i>	34	1685
	<i>Eptesicus serotinus</i>	22	953
	<i>Hypsugo savii</i>	15	438
	<i>Miniopterus schreibersii</i>	19	674
	<i>Myotis alcathoe</i>	11	277
	<i>Myotis bechsteinii</i>	18	335
	<i>Myotis blythii</i>	13	160
	<i>Myotis brandtii</i>	19	835
	<i>Myotis capaccinii</i>	24	1131
	<i>Myotis daubentonii</i>	20	815
	<i>Myotis emarginatus</i>	25	874
	<i>Myotis myotis</i>	63	634
	<i>Myotis mystacinus</i>	12	596
	<i>Myotis nattereri</i>	24	470
	<i>Nyctalus leisleri</i>	54	808
	<i>Nyctalus noctula</i>	16	717
	<i>Pipistrellus kuhlii</i>	39	683
	<i>Pipistrellus nathusii</i>	23	535
	<i>Pipistrellus pipistrellus</i>	44	1710
	<i>Pipistrellus pygmaeus</i>	20	477
	<i>Plecotus auritus</i>	20	165

BatScope

Obrist & Boesch

	<i>Plecotus austriacus</i>	21	570
	<i>Vespertilio murinus</i>	14	460
Molossidae	<i>Tadarida teniotis</i>	9	28
Total		633	19636

5

Draft

6 **Table 2: Signal parameters calculated**

7 List of all 59 call parameters calculated from each echolocation call for evaluation. The parameter

8 abbreviations, dimensions, parameter shortcuts (see Fig. 8), and explanations are given, as well as

9 the feature extraction content. Importance was extracted from Random Forest results. The list is

10 sorted by decreasing importance for a correct classification. Rank: the top 40 are used in

11 classifications.

12 **Note:** Calc details on the calculation method: 1) calculated in energy sum over frequency axis, 2)

13 calculated in energy sum over time axis, 3) first derivative of course of frequency over time, 4) second

14 derivative of course of frequency over time. Up to rank 20 only parameters deduced from the filtered

15 spectrogram occur.

16

Parameter	Dims	Abbr. in Fig. 8	Explanation	Context	Importance	Rank	Calc
TrajectoryAvg	kHz/	Slope	Mean slope of trajectory in	bins	0.09335	1	3
SlopeBin4	ms		time bin 4				
TrajectoryAvg	kHz/	Slope	Mean slope of trajectory in	bins	0.06279	2	3
SlopeBin3	ms		time bin 3				
TrajectoryAvg	kHz/	Slope	Mean slope of trajectory in	bins	0.05438	3	3
SlopeBin5	ms		time bin 5				
Trajectorycent	kHz	TraCenterFreq	Center frequency of	trajectory	0.05150	4	
erFreq			trajectory				
TrajectoryAvg	kHz	AvgFreqBin	Mean frequency of	bins	0.05058	5	
FreqBin3			trajectory in time bin 3				
BandwidthD9	kHz	BdwD90	Bandwidth containing 90%	filtered	0.04397	6	1
OFIL			of call energy in filtered	spectrogram			
			signal				
TrajectoryAvg	kHz	AvgFreqBin	Mean frequency of	bins	0.04332	7	
FreqBin2			trajectory in time bin 2				
TrajectoryAvg	kHz	AvgFreqBin	Mean frequency of	bins	0.03785	8	
FreqBin4			trajectory in time bin 4				
TrajectoryAvg	kHz	AvgFreqBin	Mean frequency of	bins	0.03230	9	
FreqBin5			trajectory in time bin 5				

3

BatScope				Obrist & Boesch			
TrajectoryAvg	kHz/	Slope	Mean slope of trajectory in	bins	0.03033	10	3
SlopeBin2	ms		time bin 2				
TrajectoryEnd	kHz	TraEndFreq	End frequency of trajectory	trajectory	0.03004	11	
Freq							
TrajectoryMin	kHz	TraMinFreq	Lowest frequency of	trajectory	0.02961	12	
Freq			trajectory				
FreqP05FIL	kHz	FreqP05	Frequency (starting at 0	filtered	0.02937	13	1
			kHz) at which 5% of total	spectrogram			
			call energy is reached in				
			filtered signal				
BandwidthIQ	kHz	BdwIQR	Bandwidth containing 50%	filtered	0.02686	14	1
RFIL			of call energy (inter quartil	spectrogram			
			range) in filtered signal				
DurationD90F	ms	DurD90	Duration containing 90% of	filtered	0.02453	15	2
IL			call energy in filtered signal	spectrogram			
TrajectoryBan	kHz	TraBandwidth	Bandwidth of trajectory	trajectory	0.02376	16	
dwidth							
FreqP95FIL	kHz	FreqP95	Frequency (starting at 0	filtered	0.02332	17	1
			kHz) at which 95% of total	spectrogram			
			call energy is reached in				
			filtered signal				
TimeP95P75	ms	*	Duration between 75% and	filtered	0.02228	18	2
			95% total call energy in	spectrogram			
			filtered signal				
TrajectoryAvg	kHz	AvgFreqBin	Mean frequency of	bins	0.02191	19	
FreqBin1			trajectory in time bin 1				
FreqP50FIL	kHz	FreqP50	Frequency (starting at 0	filtered	0.01983	20	1
			kHz) at which 50% of total	spectrogram			
			call energy is reached in				
			filtered signal				
FreqP75RAW	kHz	FreqP75	Frequency (starting at 0	raw	0.01955	21	1
			kHz) at which 75% of total	spectrogram			
			call energy is reached in				

BatScope

Obrist & Boesch

			raw signal					
FreqP05RAW	kHz	FreqP05	Frequency (starting at 0 kHz) at which 5% of total call energy is reached in raw signal	raw spectrogram	0.01951	22	1	
FreqP50RAW	kHz	FreqP50	Frequency (starting at 0 kHz) at which 50% of total call energy is reached in raw signal	raw spectrogram	0.01943	23	1	
FreqP75FIL	kHz	FreqP75	Frequency (starting at 0 kHz) at which 75% of total call energy is reached in filtered signal	filtered spectrogram	0.01926	24	1	
BandwidthIQ RRAW	kHz	BdwIQR	Bandwidth containing 50% of call energy (inter quartil range) in raw signal	raw spectrogram	0.01884	25	1	
TimeP50P25	ms	*	Duration between 25% and 50% total call energy in filtered signal	filtered spectrogram	0.01879	26	2	
TrajectoryDur ation	ms	TraDur	Duration of trajectory	trajectory	0.01861	27		
FreqP25FIL	kHz	FreqP25	Frequency (starting at 0 kHz) at which 25% of total call energy is reached in filtered signal	filtered spectrogram	0.01837	28	1	
FreqP95RAW	kHz	FreqP95	Frequency (starting at 0 kHz) at which 95% of total call energy is reached in raw signal	raw spectrogram	0.01811	29	1	
TimeP25P05	ms	*	Duration between 5% and 25% total call energy in filtered signal	filtered spectrogram	0.01772	30	2	

BatScope				Obrist & Boesch			
DurationIQRF IL	ms	DurIQRF	Duration containing 50% of call energy (inter quartil range) in filtered signal	filtered spectrogram	0.01697	31	2
FreqPeakFIL	kHz	FreqPeak	Frequency of peak energy in filtered signal	filtered spectrogram	0.01693	32	1
TrajectoryStar tFreq	kHz	TraStartFreq	Start frequency of trajectory	trajectory	0.01658	33	
TrajectoryMax Freq	kHz	TraMaxFreq	Highest frequency of trajectory	trajectory	0.01614	34	
DurationIQRR AW	ms	DurIQRF	Duration containing 50% of call energy (inter quartil range) in raw signal	raw spectrogram	0.01399	35	2
FreqPeakRA W	kHz	FreqPeak	Frequency of peak energy in raw signal	raw spectrogram	0.01391	36	1
BandwidthD9 ORAW	kHz	BdwD90	Bandwidth containing 90% of call energy in raw signal	raw spectrogram	0.01352	37	1
TimeP75P50	ms	*	Duration between 50% and 75% total call energy in filtered signal	filtered spectrogram	0.01331	38	2
FreqP25RAW	kHz	FreqP25	Frequency (starting at 0 kHz) at which 25% of total call energy is reached in raw signal	raw spectrogram	0.01122	39	1
TrajectoryAvg SlopeBin1	kHz/ ms	Slope	Mean slope of trajectory in time bin 1	bins	0.01039	40	3
DurationD90R AW	ms	DurD90	Duration containing 90% of call energy in raw signal	raw spectrogram	0.00983	41	2
TimeP05FIL	ms	TimeP05	Time (starting at 0 ms) at which 5% of total call energy is reached in filtered signal	filtered spectrogram	0.00854	42	2
TrajectoryStar tTime	ms	TraStartTime	Start time of trajectory	trajectory	0.00823	43	

BatScope			Obrist & Boesch					
TimeP05RAW	ms	TimeP05	Time (starting at 0 ms) at which 5% of total call energy is reached in raw signal	raw spectrogram	0.00742	44	2	
TimeP95FIL	ms	TimeP95	Time (starting at 0 ms) at which 95% of total call energy is reached in filtered signal	filtered spectrogram	0.00612	45	2	
TimeP25RAW	ms	TimeP25	Time (starting at 0 ms) at which 25% of total call energy is reached in raw signal	raw spectrogram	0.00550	46	2	
TimeP95RAW	ms	TimeP95	Time (starting at 0 ms) at which 95% of total call energy is reached in raw signal	raw spectrogram	0.00469	47	2	
TimeP25FIL	ms	TimeP25	Time (starting at 0 ms) at which 25% of total call energy is reached in filtered signal	filtered spectrogram	0.00321	48	2	
TrajectoryAvg CurvBin2	none	Curvature	Mean curvature of trajectory in time bin 2	bins	0.00270	49	4	
TrajectoryAvg CurvBin4	none	Curvature	Mean curvature of trajectory in time bin 4	bins	0.00255	50	4	
TimeP75RAW	ms	TimeP75	Time (starting at 0 ms) at which 75% of total call energy is reached in raw signal	raw spectrogram	0.00253	51	2	
TrajectoryAvg CurvBin3	none	Curvature	Mean curvature of trajectory in time bin 3	bins	0.00246	52	4	
TimeP50FIL	ms	TimeP50	Time (starting at 0 ms) at which 50% of total call energy is reached in filtered	filtered spectrogram	0.00226	53	2	

BatScope

Obrist & Boesch

			signal					
TimeP75FIL	ms	TimeP75	Time (starting at 0 ms) at which 75% of total call energy is reached in filtered signal	filtered spectrogram	0.00175	54	2	
TrajectoryAvg	none	Curvature	Mean curvature of trajectory in time bin 1	bins	0.00166	55	4	
CurvBin1								
TimePeakRA	ms	*	Time of peak energy in raw signal	raw spectrogram	0.00091	56	2	
W								
TimePeakFIL	ms	*	Time of peak energy in filtered signal	filtered spectrogram	0.00076	57	2	
TimeP50RAW	ms	TimeP50	Time (starting at 0 ms) at which 50% of total call energy is reached in raw signal	raw spectrogram	0.00065	58	2	
TrajectoryAvg	none	Curvature	Mean curvature of trajectory in time bin 5	bins	0.00005	59	4	
CurvBin5								

17

Table 3: R packages
R packages and the versions used for classifications.

Classifier	Abbreviation	R-package	Version	Function
k-nearest neighbours	KNN	class	7.3-14	knn()
weighted k-nearest neighbours	KKNN	kknn	1.3.0	kknn()
support vector machine	SVM	e1071	1.6-7	svm()
neural network	NN	nnet	7.3-11	nnet()
quadratic discriminant analysis	QDA	MASS	7.3-45	qda()
random forest	RF	randomForest	4.6-12	randomForest()

Draft

23 **Table 4: Parameter tuning**

24 Final parameter settings for the six classifiers.

25

Classifier	Preset parameters	Tuned parameters
KNN	-	k=7
KKNN	distance = 1, kernel = gaussian	k = 7
SVM	-	cost = 100'000, gamma = 0.01
NN	-	size = 80, decay = 0.01, maxit = 100
QDA	-	nPC=11
RF	-	mTry = 6, nTree = 1000

26

27

Draft

Table 5: Classification results
Mean (N=5) accuracy, lowest PPV, and lowest sensitivity of six classifiers in predicting *calls* and *sequences* to species. The lowest values are in *italics* and best performances underlined.

Classifi- cation of	by classifier	Measure		
		accuracy	lowest PPV	sensitivity
calls	KKNN	0.784	0.463	<i>0.195</i>
	KNN	0.762	0.440	0.225
	NN	0.806	0.353	0.298
	QDA	<i>0.716</i>	<i>0.280</i>	0.241
	RF	<u>0.815</u>	<u>0.512</u>	0.290
	SVM	0.810	0.332	<u>0.388</u>
	Mean	0.782	0.397	0.273
sequences	KKNN	0.858	0.599	0.215
	KNN	0.838	0.607	<i>0.185</i>
	NN	0.871	0.612	0.349
	QDA	<i>0.774</i>	<i>0.335</i>	0.293
	RF	0.859	0.561	0.355
	SVM	<u>0.885</u>	<u>0.651</u>	<u>0.507</u>
	Mean	0.848	0.561	0.317

Table 6: Classifier triplets

Sequence classification performance of six best triplets of classifiers measured as mean (N=5) accuracy, lowest PPV, and lowest sensitivity. Either all classifications were considered or only those where the three classifiers agreed, which lead to a reduction of assignable sequences. The lowest values are in *italics* and best performances underlined.

	Classifier triplets	accuracy	lowest		assignable
			PPV	sensitivity	
3 agree	NN - KNN - SVM	<u>89.8%</u>	<u>67.0%</u>	30.7%	96.7%
	NN - RF - SVM	<u>89.8%</u>	62.2%	39.1%	97.0%
	QDA - KNN - SVM	88.7%	55.2%	<u>48.2%</u>	94.4%
	QDA - KNN - SVM	88.5%	60.5%	41.7%	<u>94.0%</u>
	QDA - NN - SVM	88.7%	61.5%	44.5%	94.9%
	QDA - RF - SVM	89.1%	57.4%	45.9%	94.7%
	RF - KNN - KNN	87.2%	64.3%	33.1%	<u>97.9%</u>
all	NN - KNN - SVM	89.0%	<u>68.2%</u>	30.8%	<u>100.0%</u>
	NN - RF - SVM	89.0%	64.4%	39.7%	<u>100.0%</u>
	QDA - KNN - SVM	88.7%	59.6%	<u>43.3%</u>	<u>100.0%</u>
	QDA - KNN - SVM	88.3%	64.1%	41.1%	<u>100.0%</u>
	QDA - NN - SVM	<u>89.1%</u>	62.7%	<u>43.3%</u>	<u>100.0%</u>
	QDA - RF - SVM	88.8%	59.3%	38.9%	<u>100.0%</u>

Table 7: Correct Classification Rates compared

Performance of classifiers as reported in the literature. Numbers of species, recorded sequences and included calls are given on the left. The mean and minimum classifier performance is shown in the middle. Column ‘Measure’ indicates the measurement reported in the reference (CCR = Correct Classification Rate (=True Positive Rate or Sensitivity); ACC = Accuracy). Classifiers used include Hidden Markov Models (HMM), Neural Networks ((A)NN), Discriminant Function Analyses (DFA), Synergetics, and proprietary algorithms. The exact source of reference is given in the last column.

N species	N sequences	N calls	Mean % correct	Min % correct	Measure	Classifier	Reference
4	552	9698	96	93	ACC	DFA	(Britzke et al. 2002; Tabs. 1+2)
4	-	263	87	77	CCR	DFA	(Rodriguez-San Pedro & Simonetti 2013; Tab. 2)
7	300	300	97	92	CCR	proprietary	(Henríquez et al. 2014; Tab. 4)
8	171	171	92	81	CCR	DFA	(Fukui, Agetsuma & Hill 2004; Tab. 3)
8	-	158	89	83	CCR	DFA	(Wordley et al. 2014; Tab. 2)
12	48	698	85	60	CCR	ANN	(Parsons & Jones 2000; Fig. 4A)
12	1846	35979	94	40	ACC	NN	(Britzke et al. 2011; Tabs. 4+6)
14	45	-	61	0	CCR	ANN	(Jennings, Parsons & Pocock 2008; Tab. 1)
17	4386	111361	89	56	CCR	HMM	(Agranat 2012; Tab. 10)
18	950	-	82	38	CCR	DFA	(Russo & Jones 2002; Tab. 3)
26	643	14354	83	50	CCR	Synergetics	(Obrist, Boesch & Flückiger 2004; Tab. 3)
27	633	19636	89	48	CCR	DFA	this study
34	1350	1350	81	49	CCR	ANN	(Walters et al. 2012; Fig. 3)