Species-specific audio detection: A comparison of three template-based detection algorithms using random forests

Carlos J Corrada BravoCorresp., 1, 2, Rafael Álvarez Berrios 2, T. Mitchell Aide 2, 3

1 Department of Computer Science, University of Puerto Rico - Rio Piedras, San Juan, Puerto Rico
2 Sieve Analytics, Inc., San Juan, Puerto Rico
3 Department of Biology, University of Puerto Rico - Río Piedras, San Juan, Puerto Rico

Corresponding Author: Carlos J Corrada Bravo
Email address: carlos.corrada2@upr.edu

We developed a web-based cloud-hosted system that allow users to archive, listen, visualize, and annotate recordings. The system also provides tools to convert these annotations into datasets that can be used to train a computer to detect the presence or absence of a species. The algorithm used by the system was selected after comparing the accuracy and efficiency of three variants of a template-based detection. The algorithm computes a similarity vector by comparing a template of a species call with time increments across the spectrogram. Statistical features are extracted from this vector and used as input for a Random Forest classifier that predicts presence or absence of the species in the recording. The fastest algorithm variant had the highest average accuracy and specificity; therefore, it was implemented in the ARBIMON web-based system.
Species-specific audio detection: A comparison of three template-based detection algorithms using random forests

Carlos J Corrada Bravo\textsuperscript{1,2}, Rafael Álvarez Berrios\textsuperscript{2}, and T. Mitchell Aide\textsuperscript{2,3}

\textsuperscript{1}Department of Computer Science, University of Puerto Rico - Rio Piedras.
\textsuperscript{2}Sieve Analytics, Inc., San Juan, Puerto Rico.
\textsuperscript{3}Department of Biology, University of Puerto Rico - Rio Piedras.

ABSTRACT

We developed a web-based cloud-hosted system that allow users to archive, listen, visualize, and annotate recordings. The system also provides tools to convert these annotations into datasets that can be used to train a computer to detect the presence or absence of a species. The algorithm used by the system was selected after comparing the accuracy and efficiency of three variants of a template-based detection. The algorithm computes a similarity vector by comparing a template of a species call with time increments across the spectrogram. Statistical features are extracted from this vector and used as input for a Random Forest classifier that predicts presence or absence of the species in the recording. The fastest algorithm variant had the highest average accuracy and specificity; therefore, it was implemented in the ARBIMON web-based system.

INTRODUCTION

Monitoring fauna is an important task for ecologists, natural resource managers, and conservationists. Historically, most data were collected manually by scientists that went to the field and annotated their observations (Terborgh et al., 1990). This generally limited the spatial and temporal extent of the data. Furthermore, given that the data were based on an individual’s observations, the information was difficult to verify, reducing its utility for understanding long-term ecological processes (Acevedo and Villanueva-Rivera, 2006).

To understand the impacts of climate change and deforestation on the fauna, the scientific community needs long-term, wide-spread and frequent data (Walther et al., 2002). Passive acoustic monitoring (PAM) can contribute to this need because it facilitates the collection of large amounts of data from many sites simultaneously, and with virtually no impact to the fauna and environment (Brandes, 2008; Lammers et al., 2008; Tricas and Boyle, 2009; Celis-Murillo et al., 2012). In general, PAM systems include a microphone or a hydrophone connected to a self powered system and enough memory to store various weeks or months of recordings, but there are also permanent systems that use solar panels and an Internet connection to upload recordings in real time to a cloud based analytical platform (Aide et al., 2013).

Passive recorders can easily create a very large data set (e.g. 100,000s of recordings) that is overwhelming to manage and analyze. Although researchers often collect recordings twenty-four hours a day for weeks or months (Acevedo and Villanueva-Rivera, 2006; Brandes, 2008; Lammers et al., 2008; Sueur et al., 2008; Marques et al., 2013; Blumstein et al., 2011), in practice, most studies have only analyzed a small percentage of the total number of recordings.

Web-based applications have been developed to facilitate data management of these increasingly large datasets (Aide et al., 2013; Villanueva-Rivera and Pijanowski, 2012), but the biggest challenge is to develop efficient and accurate algorithms for detecting the presence or absence of a species in many recordings. Algorithms for species identification have been developed using spectrogram matched filtering (Clark et al., 1987; Chabot, 1988), statistical feature extraction (Taylor, 1995; Grigg et al., 1996), k-Nearest neighbor algorithm (Hana et al., 2011; Gunasekaran and Revathy, 2010), Support Vector Machine (Fagerlund, 2007; Acevedo et al., 2009), tree-based classifiers (Adams et al., 2010; Henderson and Hildebrand, 2011) and template based detection (Anderson et al., 1996; Mellinger and Clark, 2000),
but most of these algorithms are built for a specific species and there was no infrastructure provided for

the user to create models for other species.

In this study, we developed a method that detects the presence or absence of a species’ specific call
type in recordings with a response time that allows researchers to create, run, tune and re-run models
in real time as well as detect hundreds of thousands of recordings in a reasonable time. The main
objective of the study was to compare the performance (e.g. efficiency and accuracy) of three variants
of a template-based detection algorithm and incorporate the best into the ARBIMON II bioacoustics
platform. The first variant is the Structural Similarity Index described in Wang et al. (2004), a widely
use method to find how similar two images are (in our case the template with the tested recording). The
second method filters the recordings with the dynamic thresholding method described in Wang et al.
(2004) and then use the Frobenius norm to find similarities with the template. The final method uses the
Structural Similarity Index, but it is only applied to regions with high match probability determined by
the OpenCV’s matchTemplate procedure (Bradski, 2000).

MATERIALS AND METHODS

Passive acoustic data acquisition

We gathered recordings from five locations, four in Puerto Rico and one in Peru. Some of the record-
ings were acquired using the Automated Remote Biodiversity Monitoring Network (ARBIMON) data
acquisition system described in Aide et al. (2013), while others were acquired using the newest version of
ARBIMON permanent recording station, which uses an Android cell phone and transmits the recorded
data through a cellular network. All recordings have a sampling rate of 44.1kHz, a sampling depth of
16-bit and an approximate duration of 60 seconds (±.5s)

The locations in Puerto Rico were the Sabana Seca permanent station in Toa Baja, the Casa la Selva
station in Carite Mountains (Patillas), El Yunque National Forest in Rio Grande and Mona Island (see
Figure 1). The location in Peru was the Amarakaeri Communal Reserve in the Madre de Dios Region
(see Figure 2). In all the locations, the recorders were programmed to record one minute of audio every
10 minutes. The complete dataset has more than 100,000 1-minute recordings. We randomly chose 362
recordings from Puerto Rico and 547 recordings from Peru for comparing the three algorithm variants.

![Figure 1. Recording locations in Puerto Rico. Map data: Google, Image - Landsat / Copernicus and Data - SIO, NOAA, US Navy, NGA and GEBCO.](Image)

We used the ARBIMON II web application interface to annotate the presence or absence of 21 species
in all the recordings. Regions in the recording where a species emits a sound were also marked using
the web interface. Each region of interest (ROI) is a rectangle delimited by starting time, ending time,
lowest frequency and highest frequency along with a species and sound type. The species included in the
analysis are listed in Table 1, along with the number of total recordings and the number of recordings
where the species is present or absent.

Algorithm

The algorithm recognition process is divided into three phases: 1) Template Computation, 2) Model
Training and 3) Detection (see Figure 3). In Template computation, all ROIs submitted by the user in the
training set are aggregated into a template. In Model Training the template is used to compute recognition
functions from validated audio recordings and features from the resulting vector $V$ are computed. These features are used to train a random forest model. In the Detection phase the template is used to compute the features, but this time the features are fed to the trained random forest model to compute a prediction of presence or absence.

**Figure 2.** Recording location in Peru. Map data: Google, US Dept. of State Geographer, Image - Landsat / Copernicus and Data - SIO, NOAA, US Navy, NGA and GEBCO.

**Figure 3.** The three phases of the algorithm to create the species-specific models. In the Model Training phase $Rec_i$ is a recording, $V_i$ is the vector generated by the recognition function on $Rec_i$, and in the Detection phase $V$ is the vector generated by the recognition function on the incoming recording.
Table 1. Species, class, location and count of recordings with validated data.

| Species                                | Group         | Total | Presence | Absence | Location       |
|----------------------------------------|---------------|-------|----------|---------|----------------|
| Eleutherodactylus cooki                | Amphibian     | 38    | 19       | 19      | Carite         |
| Eleutherodactylus brittoni             | Amphibian     | 38    | 17       | 21      | Sabana Seca    |
| Eleutherodactylus cochranae            | Amphibian     | 54    | 30       | 24      | Sabana Seca    |
| Eleutherodactylus coqui               | Amphibian     | 53    | 41       | 12      | Sabana Seca    |
| Eleutherodactylus juanariveroi        | Amphibian     | 35    | 14       | 21      | Sabana Seca    |
| Unknown Insect                         | Insect        | 48    | 22       | 26      | Sabana Seca    |
| Epinephelus guttatus                  | Fish          | 152   | 76       | 76      | Mona Island    |
| Megascops nudipes                     | Bird          | 100   | 50       | 50      | El Yunque      |
| Microcerculus marginatus              | Bird          | 80    | 40       | 40      | Peru           |
| Basiluterus chrysogaster              | Bird          | 60    | 30       | 30      | Peru           |
| Myrmoborus leucophys                  | Bird          | 160   | 80       | 80      | Peru           |
| Basiluterus bivittatus                | Bird          | 100   | 50       | 50      | Peru           |
| Lioseles thoracicus                   | Bird          | 76    | 38       | 38      | Peru           |
| Chlorothraupis carmioli               | Bird          | 112   | 56       | 56      | Peru           |
| Megascops guatemalae                  | Bird          | 28    | 8        | 20      | Peru           |
| Saltator grossus                      | Bird          | 68    | 34       | 34      | Peru           |
| Myrmeciza hemimelaena                 | Bird          | 180   | 90       | 90      | Peru           |
| Thamnophilus schistaceus              | Bird          | 60    | 30       | 30      | Peru           |
| Hypocnemis subflava                   | Bird          | 140   | 70       | 70      | Peru           |
| Percnostola lophotes                  | Bird          | 100   | 50       | 50      | Peru           |
| Formicarius analis                    | Bird          | 80    | 40       | 40      | Peru           |

In the following sections the Template Computation process will be explained, then the process of using the Template to extract features from a recording is presented and finally, the procedures to use the features to train the model and to detect recordings are discussed.

Template Computation

The template refers to the combination of all ROIs in the training data. To create a template, we first start with the examples of the specific call of interest (i.e. ROIs) that were annotated from a set of recordings for a given species and a specific call type (e.g. common, alarm). Each ROI encompasses an example of the call, and is an instance of time between time $t_1$ and time $t_2$ of a given recording and low and high boundary frequencies of $f_1$ and $f_2$, where $t_1 < t_2$ and $f_1 < f_2$. In a general sense, we combine these examples to produce a template of a specific song type of a single species.

Specifically, for each recording that has an annotated ROI, a spectrogram matrix ($SM$) is computed using the Short Time Fourier Transform with a frame size of 1024 samples, 512 samples of overlap and a Hann analysis window, thus the matrices have 512 rows. For a recording with a sampling rate of 44,100 Hz, the matrix bin bandwidth is approximately 43.06 Hz. The $SM$ is arranged so that the row of index 0 represents the lowest frequency and the row with index 511 represents the highest frequency of the spectrum. Properly stated the columns $c_1$ to $c_2$ and the rows from $r_1$ to $r_2$ of $SM$ were extracted, where:

$$c_1 = \lfloor t_1 \times 44100 \rfloor, \quad c_2 = \lfloor t_2 \times 44100 \rfloor, \quad r_1 = \lfloor f_1/43.06 \rfloor \quad \text{and} \quad r_2 = \lfloor f_2/43.06 \rfloor.$$

The rows and columns that represent the ROI in the recording (between frequencies $f_1$ and $f_2$ and between times $t_1$ and $t_2$) are extracted. The submatrix of $SM$ that contains only the area bounded by the ROI is define as $SM_{ROI}$ and refer in the manuscript as the ROI matrix.

Since the ROI matrices can vary in size, to compute the aggregation from the ROI matrices we have to take into account the difference in the number of rows and columns of the matrices. All recordings have the same sampling rate, 44100Hz. Thus, the rows from different $SM$s, computed with the same parameters, will represent the same frequencies, i.e. rows with same indexes represent the same frequency. After the ROI matrix, $SM_{ROI}$, has been extracted from $SM$, the rows of $SM_{ROI}$ will also represent specific frequencies. Thus, if we were to perform an element-wise matrix sum between two ROI matrices with potentially different number of rows, we should only sum rows that represent the same frequency.
To take into account the difference in the number of columns of the ROI matrices, we use the Frobenius norm to optimize the alignment of the smaller ROI matrices and perform element-wise sums between rows that represent the same frequency. We present that algorithm in the following section and a flow chart of the process in Figure 4.

**Template Computation Algorithm:**

1. Generate the set of $SM_{ROI}$ matrices by computing the short time Fourier Transform of all the user generated ROIs.

2. Create matrix $SM_{max}$, a duplicate of the first created matrix among the matrices with the largest number of columns.

3. Set $c_{max}$ as the number of columns in $SM_{max}$.

4. Create matrix $T_{temp}$, with the same dimensions as $SM_{max}$ and all entries equal to 0. This matrix will contain the element-wise addition of all the extracted $SM_{ROI}$ matrices.

5. Create matrix $W$ with the same dimensions of $SM_{max}$ and all entries equal to 0. This matrix will hold the count on the number of $SM_{ROI}$ matrices that participate in the calculation of each element of $T_{temp}$.

6. For each one of the $SM_i$ ROI matrices in $SM_{ROI}$:

   (a) If $SM_i$ has the same number of columns as $T_{temp}$:
      
      i. Align the rows of $SM_i$ and $T_{temp}$ so they represent equivalent frequencies and perform an element-wise addition of the matrices and put the result in $T_{temp}$.
      
      ii. Add one to all the elements of the $W$ matrix where the previous addition participated.

   (b) If the number of columns differs between $SM_i$ and $T_{temp}$, then find the optimal alignment with $SM_{max}$ as follows:
      
      i. Set $c_i$ as the number of columns in $SM_i$.
      
      ii. Define $(SM_{max})_i$ as the set of all submatrices of $SM_{max}$ with the same dimensions as $SM_i$. Note that the cardinality of $(SM_{max})_i$ is $c_{max} - c_i$.
      
      iii. For each $Sub_k \in (SM_{max})_i$:
          
          A. Compute $d_k = NORM(Sub_k - SM_i)$ where $NORM$ is the Frobenius norm defined as:

          $NORM(A) = \sqrt{\sum_{(i,j)} |a_{i,j}|^2}$

          where $a_{i,j}$ are the elements of matrix $A$.

      iv. Define $Sub_{min}(d_i)$ as the $Sub_k$ matrix with the minimum $d_k$. This is the optimal alignment of $SM_i$ with $SM_{max}$.

      v. Align the rows of $Sub_{min}(d_i)$ and $T_{temp}$ so they represent equivalent frequencies, perform an element-wise addition of the matrices and put the result in $T_{temp}$.

      vi. Add one to all the elements of the $W$ matrix where the previous addition participated.

7. Define the matrix $T_{template}$ as the element-wise division between the $T_{temp}$ matrix and the $W$ matrix.

The resulting $T_{template}$ matrix summarizes the information available in the ROI matrices submitted by the user and it will be used to extract information from the audio recordings that are to be analyzed. In this article each species $T_{template}$ was created using five ROIs.

In Figure 5a a training set for the *Eleutherodactylus coqui* is presented and in Figure 5b the resulting template can be seen. This tool is very useful because the user can see immediately the effect of adding or subtracting a specific sample to the training set.
Figure 4. Flowchart of the algorithm to generate the template of each species.

**Model Training**

The goal of this phase is to train a random forest model. The input to train the random forest are a series of statistical features extracted from vectors $V_i$ that are created by computing a recognition function (similarity measure) between the computed $T_{template}$ and submatrices of the spectrogram matrices of a series of recordings.

In the following section we present the details of the algorithm that processes a recording to create the recognition function vector and in Figure 6, we present a flowchart of the process.

**Algorithm to Create the Similarity Vector:**

1. Compute matrix $SPEC$, the submatrix of the spectrogram matrix that contains the frequencies in $T_{template}$. Note that we are dealing with recordings that have the same sample rate as the recordings used to compute the $T_{template}$.

2. Define $c_{SPEC}$, the number of columns of $SPEC$.

3. Define $c_{template}$, the number of columns of $T_{template}$. Note that $c_{SPEC} >> c_{template}$ since the $SPEC$ matrix have the same number of columns as the whole spectrogram and that the $T_{template}$ matrix fits $c = c_{SPEC} - c_{template} + 1$ times inside the $SPEC$ matrix. There are $c$ submatrices of $SPEC$ with the same dimensions as $T_{template}$.
4. Define \( \text{step} \), the step factor by which \( T_{\text{template}} \) will progressed over the \( \text{SPEC} \) matrix.

5. Define \( n = \left\lceil \frac{c_{\text{SPEC}} - c_{\text{template}}}{\text{step}} \right\rceil + 1 \). Note that if \( \text{step} = 1 \) then \( n = c \). In this work, however, this parameter was selected as \( \text{step} = 16 \) as a trade-off for speed\(^{1}\).

6. Define \( \text{SPEC}_i \) as the submatrix of \( \text{SPEC} \) that spans the columns from \( i \times \text{step} \) to \( i \times \text{step} + c_{\text{template}} \).

7. Set \( i = 1 \)

8. While \( i <= n \)

   (a) Compute the similarity measure \( \text{meas}_i \) for \( \text{SPEC}_i \) (the definition of \( \text{meas}_i \) for each of the three variants is provided in the following section).

   (b) Increase \( i \) by 1. Note that this is equivalent to progressing \( \text{step} \) columns in the \( \text{SPEC} \) matrix.

9. Define the vector \( V \) as the vector containing the \( n \) similarity measures resulting from the previous steps. That is, \( V = [\text{meas}_1, \text{meas}_2, \text{meas}_3, \cdots, \text{meas}_n] \).

**Recognition Function**

We used three variations of a pattern match procedure to define the similarity measure vector \( V \). First, the Structural Similarity Index described in Wang et al. (2004) and implemented in van der Walt et al. (2014) as \text{compare/ssim} with the default window size of seven unless the generated pattern is smaller. It will be referred in the rest of the manuscript as the SSIM variant. For the SSIM variant we define \( \text{meas}_i \) as:

\[
\text{meas}_i = \text{SSI}(T_{\text{template}}, \text{SPEC}_i),
\]

where \( \text{SPEC}_i \) is the submatrix of \( \text{SPEC} \) that spans the columns from \( i \times \text{step} \) to \( i \times \text{step} + c_{\text{template}} \) and the same number of rows as \( T_{\text{template}} \), and \( V = [\text{meas}_1, \text{meas}_2, \text{meas}_3, \cdots, \text{meas}_n] \) with

\[
n = \left\lceil \frac{c_{\text{SPEC}} - c_{\text{template}}}{\text{step}} \right\rceil + 1.
\]

\(^{1}\)Note that for recordings with a sample rate of 44100 when we calculate the STFT with a window of size 512 and a 50% overlap, one step is equivalent to 5.8 milliseconds, therefore, 16 steps is less than 100 milliseconds. Although this procedure may miss the strongest match, the length of the calls are much longer than the step interval; therefore, there is a high probability of detecting the species-specific call.
Second, the dynamic thresholding method (threshold_adaptive) described in Wang et al. (2004) with a block size of 127 and an arithmetic mean filter is used over both $T_{\text{template}}$ and $SPEC_i$ before multiplying them and applying the Frobenius norm and normalized by the norm of a matrix with same dimensions as $T_{\text{template}}$ and all elements equal to one. Therefore, $meas_i$ for the NORM variant is defined as:

$$meas_i = FN(\text{DTM}(T_{\text{template}}) \cdot DTM(SPEC_i))/FN(U),$$

where again $SPEC_i$ is the submatrix of $SPEC$ that spans the columns from $i \times \text{step}$ to $i \times \text{step} + c_{\text{template}}$. $FN$ is the Frobenius norm, DTM is the dynamic thresholding method, $U$ is a matrix with same dimensions as $T_{\text{template}}$ with all elements equal to one and $\cdot$ performs an element-wise multiplication of the matrices. Again, $V = [meas_1, meas_2, meas_3, \cdots, meas_n]$ with

$$n = \left\lfloor \frac{c_{SPEC} - c_{\text{template}}}{\text{step}} \right\rfloor + 1.$$

Finally, for the CORR variation we first apply the OpenCV’s matchTemplate procedure (Bradski, 2000) with the Normalized Correlation Coefficient option to $SPEC_i$, the submatrix of $SPEC$ that spans the columns from $i \times \text{step}$ to $i \times \text{step} + c_{\text{template}}$. However, for this variant, $SPEC_i$ includes two additions rows above and below, thus it is slightly larger than the $T_{\text{template}}$. With these we can define:

$$meas_{j,i} = \text{CORR}(T_{\text{template}}, SPEC_{j,i})$$

where $SPEC_{j,i}$ is the submatrix of $SPEC_i$ that starts at row $j$ (note that there are 5 such $SPEC_{j,i}$ matrices).

Now, we select 5 points at random from all the points above the 98.5 percentile of $meas_{j,i}$ and apply the Structural Similarity Index 5 strongly-matching regions. The size of these regions is eight thirds ($8/3$) of the length of $T_{\text{template}}$, $4/3$ before and $4/3$ after the strongly-matched point and was selected. Then, define $\text{FilterSPEC}$ as the matrix that contains these 5 strongly-matching regions and $\text{FilterSPEC}_i$ as the submatrix of $\text{FilterSPEC}$ that spans the columns from $i$ to $i + c_{\text{template}}$ then, the similarity measure for this variant is define as:

$$meas_i = \text{SSI}(T_{\text{template}}, \text{FilterSPEC}_i)$$

and the resulting vector $V = [meas_1, meas_2, meas_3, \cdots, meas_n]$ but this time with

$$n = 5 \times \left(\left\lfloor \frac{8}{3} \times c_{\text{template}} \right\rfloor + 1 \right).$$

---

Figure 6. Flowchart of the algorithm to generate the similarity vector of each recording.
It is important to note that no matter which variant is used to calculate the similarity measures, the result will always be a vector of measurements \( V \). The idea is that the statistical properties of these computed recognition functions have enough information to distinguish between a recording that has the target species present and a recording that does not have the target species present. However, notice that since \( c_{SPEC} \), the length of \( SPEC \), is much larger than \( c_{template} \), the length of the vector \( V \) for the \( CORR \) variant is much smaller than the other two.

**Random Forest Model Creation**

After calculating \( V \) for many recording we can train a random forest model. First, we need a set of validated recordings with the specific species vocalization present in some recordings and absent in others. Then for each recording we compute a vector \( V_i \) as described in the previous section and extract the statistical features presented in Table 2. These statistical features represent the dataset used to train the random forest model, which will be used to detect recordings for presence or absence of a species call event. These 12 features along with the species presence information are used as input to a random forest classifier with a 1000 trees.

**Recording Detection**

Now that we have a trained model to detect a recording, we have to compute the statistical features from the similarity vector \( V \) of the selected recording. This is performed in the same way as it was described in the previous section. These features are then used as the input dataset to the previously trained random forest classifier and a label indicating presence or absence of the species in the recording is given as output.

**The Experiment**

To decide which of the three variants should be incorporated into the ARBIMON web-based system, we performed the algorithm explained in the previous section with each of the similarity measures. We computed 10-fold validations on each of the variants to obtained measurements of the performance of the algorithm. In each validation 90% of the data is used as training and 10% of the data is used as validation data. Each algorithm variant used the same 10-fold validation partition for each species. The measures calculated were the area under the receiver operating characteristic (ROC) curve (\( AUC \)), accuracy or correct detection rate (\( Ac \)), negative predictive value (\( Npv \)), precision or positive predictive value (\( Pr \)), sensitivity, recall or true positive rate (\( Se \)) and specificity or true negative rate (\( Sp \)). To calculate the \( AUC \), the ROC curve is created by plotting the false positive rate (which can be calculated as \( 1 - Sp \)) against the true positive rate (sensitivity), then, the \( AUC \) is created by calculating the area under that curve. Notice that the further the \( AUC \) is from 0.5 the better. The rest of the measures are defined as follows:

\[
Ac = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}, \quad Npv = \frac{t_n}{t_n + f_n}, \quad Pr = \frac{t_p}{t_p + f_p}, \quad Se = \frac{t_p}{t_p + f_n} \quad \text{and} \quad Sp = \frac{t_n}{t_n + f_p}
\]

with \( t_p \) the number of true positives (number of times both the expert and the algorithm agree that the species is present), \( t_n \) the number of true negatives (number of times both the expert and the algorithm agree that the species is absent), \( f_p \) the number of false positives (number of times the algorithm but not the expert agree that the species is present) and \( f_n \) the number of false negatives (number of times the expert but not the algorithm agree that the species is absent).
agree that the species is not present), \(f_p\) the number of false positives (number of times the algorithm states that the species is present while the expert states it is absent) and \(f_n\) the number of false negatives (number of times the algorithm states that the species is not present while the expert states it is present).

Note that accuracy is a weighted average of the sensitivity and the specificity.

Although we present and discuss all measures, we gave accuracy and the AUC more importance because they include information on the true positive and true negative rates. Specifically, AUC is important when the number of positives is different than the number of negatives as is the case with some of the species.

The experiment was performed in a computer with an Intel i7 4790K 4 cores processor at 4.00 GHz with 32GB of RAM and running Ubuntu Linux. The execution time needed to detect each recording was registered and the mean and standard deviation of the execution times were calculated for each variant of the algorithm. We also computed the quantity of pixels on all the \(T_{\text{template}}\) matrices and correlated with the execution time of each of the variants.

A global one-way analysis of variance (ANOVA) was performed on the five calculated measures across all of the 10-fold validations to identify if there was a significant difference between the variants of the algorithm. Then a post-hoc Tukey HSD comparison test was performed to identify which one of the variants was significantly different at the 95% confidence level. Additionally, an ANOVA was performed locally between the 10-fold validation of each species and on the mean execution time for each species across the algorithm variants to identify if there was any significant execution time difference at the 95% confidence level. Similarly, a post-hoc Tukey HSD comparison test was performed on the execution times.

RESULTS

The six measurements (area under the ROC curve - AUC, accuracy, negative predictive value, precision, sensitivity and specificity) computed to compared the model across the three variants varied greatly among the 21 species. The lowest scores were among bird species while most of the highest scores came from amphibian species. Table 3 presents a summary of the results of the measurements comparing the three variants of the algorithm (for a detail presentation see Appendix 1). The NORM variant did not have the highest value for any of the measures summarized in Table 3, while the CORR variant had a greater number of species with 80% or greater for all the measures and an overall median accuracy of 81%. We considered these two facts fundamental for a general-purpose species detection system.

The local species ANOVA suggested that there are significant accuracy differences at the 95% significance level for 6 of the 21 species studied as well as 4 in terms of precision and 3 in terms of specificity (see supplemental materials). The algorithm variant CORR had a higher mean and median AUC at 78% and 81% respectively, but the SSIM variant seems to be more stable with a standard deviation of 20%. In terms of accuracy, both the SSIM and CORR have higher mean accuracy than the NORM variant. Nevertheless, variant CORR had the highest median accuracy of 81%, which is slightly higher than the median accuracy of the SSIM variant at 76%. In addition, variant CORR had more species with an accuracy of 80% or greater.

In terms of median precision, the three variants had similar values, although in terms of mean precision variants SSIM and CORR have greater values than the NORM variant. Moreover, the median and mean precision of the SSIM variant were only 1% higher than the median and mean precision of the CORR variant. In terms of sensitivity, variants SSIM and CORR had greater values than the NORM variant. It is only in terms of specificity that the CORR variant has greater values than all other variants. Figures 7 and 8 present a summary of these results with whisker graphs.

In terms of execution times, an ANOVA analysis on the mean execution times suggests a difference between the variants \(F = 9.9341e+30, df = 3, p < 2.2e − 16\). The CORR variant had the lowest mean execution time at 0.255s followed closely by the NORM variant with 0.271s, while the SSIM variant had the smallest mean execution time of 2.269s (Figure 9). The Tukey HSD test suggests that there was no statistical significant difference between the mean execution times of the NORM and CORR variants \(p = 0.999\). However, there was a statistical significant difference at the 95% confidence level between the mean execution times of all other pairs of variants, specifically variants SSIM and CORR \(p < 2.2e − 16\).

Moreover, the mean execution time of the SSIM variant increased as the number of pixels in the \(T_{\text{template}}\) matrix increases (Figure 9b). There was no statistically significant relationship between the \(T_{\text{template}}\) pixel size and the execution time for the other two variants (Table 4).
| Summary of measures                                             | SSIM | NORM | CORR |
|----------------------------------------------------------------|------|------|------|
| Number of species with an **Area under the curve** of 80% or greater | 8    | 7    | 12   |
| Number of species with statistically significant **Area under the curve** | 0    | 0    | 0    |
| Mean **Area under the curve**                                      | 0.76 | 0.71 | **0.78** |
| Median **Area under the curve**                                     | 0.75 | 0.72 | **0.81** |
| Standard Deviation of **Area under the curve**                    | **0.20** | 0.21 | 0.21 |
| Number of species with an **Accuracy** of 80% or greater           | 8    | 7    | 12   |
| Number of species with statistically significant **Accuracy**       | 0    | 0    | 0    |
| Mean **Accuracy**                                                  | **0.77** | 0.73 | **0.77** |
| Median **Accuracy**                                                | 0.76 | 0.75 | **0.81** |
| Standard Deviation of **Accuracy**                                 | **0.12** | 0.14 | 0.14 |
| Number of species with an **Negative predictive value** of 80% or greater | 7    | 5    | **10** |
| Number of species with statistically significant **Negative predictive value** | 0    | 0    | 0    |
| Mean **Negative predictive value**                                 | 0.73 | 0.71 | **0.74** |
| Median **Negative predictive value**                               | 0.71 | 0.75 | **0.79** |
| Standard Deviation of **Negative predictive value**               | **0.08** | 0.12 | 0.13 |
| Number of species with an **Precision** of 80% or greater          | 5    | 5    | 9    |
| Number of species with statistically significant **Precision**      | 2    | 0    | 2    |
| Mean **Precision**                                                 | **0.73** | 0.68 | 0.72 |
| Median **Precision**                                               | **0.75** | 0.73 | 0.74 |
| Standard Deviation of **Precision**                                | **0.12** | 0.13 | 0.16 |
| Number of species with an **Sensitivity** of 80% or greater         | 8    | 6    | **11** |
| Number of species with statistically significant **Sensitivity**     | 0    | 0    | 0    |
| Mean **Sensitivity**                                               | **0.77** | 0.70 | 0.74 |
| Median **Sensitivity**                                             | 0.79 | 0.73 | **0.80** |
| Standard Deviation of **Sensitivity**                              | **0.12** | 0.16 | 0.17 |
| Number of species with an **Specificity** of 80% or greater         | 4    | 6    | 7    |
| Number of species with statistically significant **Specificity**     | 3    | 0    | 0    |
| Mean **Specificity**                                               | 0.69 | 0.68 | **0.72** |
| Median **Specificity**                                             | 0.67 | 0.70 | **0.75** |
| Standard Deviation of **Specificity**                              | **0.13** | 0.15 | 0.16 |
| Ratio of False positive to True positive                           | **0.37** | 0.47 | 0.39 |
| Ratio of False negative to True positive                           | 0.45 | 0.47 | **0.39** |
| Ratio of False positive to True negative                           | **0.3** | 0.43 | 0.35 |
| Ratio of False negative to True negative                           | **0.37** | 0.43 | **0.35** |

**Table 3.** Summary of the measures of the three variants of the algorithm. Best values are in bold.

In summary, variants SSIM and CORR outperform the NORM variant in most of the statistical measures computed having statistically significant high accuracy for three species each. In terms of execution time, the CORR variant was faster than the SSIM variant (Table 3), and the mean execution time of CORR variant did not increase with increasing \( T_{\text{template}} \) size (Table 4).

**DISCUSSION**

The algorithm used by the ARBIMON system was selected by comparing three variants of a template-based method for the detection of presence or absence of a species vocalization in recordings. The most important features for selecting the algorithm were that it works well for many types of species calls and that it can process hundreds of thousands of recordings in a reasonable amount of time. The CORR algorithm was selected because of its speed and its comparable performance in terms of detection efficiency with the SSIM variant. It achieved AUC and accuracy of 0.80 or better in 12 of the 21 species and sensitivity of 0.80 or more in 11 of the 21 species and the average execution time of 0.26s per minute per recording means that it can process around 14,000 minutes of recordings per hour.

The difference in execution time between the SSIM variant and the other two was due to a memory...
Table 4. Summary of the execution times of the three variants of the algorithm. Best values are in bold. PPMCC is the Pearson product-moment correlation coefficient.

| Summary of execution times          | SSIM | NORM | CORR |
|-------------------------------------|------|------|------|
| Mean Execution Time                | 2.27 | 0.27 | **0.26** |
| Standard Deviation of Execution Time | 3.04 | **0.06** | 0.07  |
| PPMCC between Execution Time and size of template | 0.96 | 0.33 | **0.11** |

An interesting comparison is the method described in the work by (Fodor, 2013) and adapted and tested by (Lasseck, 2013). This method was designed for the Neural Information Processing Scaled for Bioacoustics (NIPS4B) competition and although the results are very good they do not report on time of execution. As we have mentioned it is very important to us to have a method that provides good response times and the execution time of Lasseck’s method seems to be greater than ours given the extensive pre-processing that method performs.

CONCLUSIONS AND FUTURE WORK

Now that passive autonomous acoustic recorders are readily available the amount of data is growing exponentially. For example, one permanent station recording one minute of every 10 minutes every day of the year generates 52,560 one minute recordings. If this is multiplied by the need to monitor thousands of locations across the planet, one can understand the magnitude of the task at hand.
We have shown how the algorithm used in the ARBIMON II web-based cloud-hosted system was selected. We compared the performance in terms of the ability to detect and the efficiency in terms of time execution of three variants of a template-based detection algorithm. The result was a method that uses the power of a widely use method to determine the similarity between two images (Structural Similarity Index (Wang et al., 2004)), but to accelerate the detection process, the analysis was only done in regions where there was a strong-match determined by the OpenCV’s matchTemplate procedure (Bradski, 2000). The results show that this method performed better both in terms of ability to detect as well as in terms of execution time.
A fast and accurate general-purpose algorithm for detecting presence or absence of a species complements the other tools of the ARBIMON system, such as options for creating playlists based on many different parameters including user-created tags (see Table 5). For example, the system currently has 1,749,551 1-minute recordings uploaded by 453 users and 659 species specific models have been created and run over 3,780,552 minutes of recordings of which 723,054 are distinct recordings.

While this research was a prove of concept, we provide the tools and encourage users to increase the size of the training data set as this should improve the performance of the algorithm. In addition, we will pursue other approaches, such as multi-label learning (Xie et al., 2016; Zhang et al., 2016; Briggs et al., 2012).

| Number of users in the system | 453 |
| Number of recordings in the system | 1,749,551 |
| Number of models created by users | 659 |
| Total number of detected recordings | 3,780,552 |
| Number of distinct detected recordings | 723,054 |
| Average times a recording is detected | 5.22 |
| Standard deviation of the number of times a recording is detected | 7.78 |
| Maximum number of times a recordings has been detected | 58 |

**Table 5.** Summary of the usage of the ARBIMON2 system and its model creation feature.

As a society, it is fundamental that we study the effects of climate change and deforestation on the fauna and we have to do it with the best possible tools. We are collecting a lot of data, but until recently there was not an intuitive and user-friendly system that allowed scientists to manage and analyze large number of recordings. Here we presented a web-based cloud-hosted system that provides a simple way to manage large quantities of recordings with a general-purpose method to detect their presence in recordings.
ACKNOWLEDGMENTS

The authors want to thank Marconi Campos-Cerqueira for his helpful comments on the manuscript.

REFERENCES

Acevedo, M. and Villanueva-Rivera, L. (2006). Using automated digital recording systems as effective tools for the monitoring of birds and amphibians. *Wildlife Society Bulletin*, 34:211–214.

Acevedo, M. A., Corrada-Bravo, C. J., Corrada-Bravo, H., Villanueva-Rivera, L. J., and Aide, T. M. (2009). Automated classification of bird and amphibian calls using machine learning: a comparison of methods. *Ecological Informatics*, 4:206–214.

Adams, M. D., Law, B. S., and Gibson, M. S. (2010). Reliable automation of bat call identification for eastern new south wales, australia, using classification trees and anascheme software. *Acta Chiropterologica*.

Aide, T. M., Corrada-Bravo, C., Campos-Cerqueira, M., Milan, C., Vega, G., and Alvarez, R. (2013). Real-time bioacoustics monitoring and automated species identification. *PeerJ*, 1:e103.

Anderson, S., Dave, A. S., and Margoliash, D. (1996). Template-based automatic recognition of birdsong syllables from continuous recordings. *The Journal of the Acoustical Society of America*, 100:1209–1219.

Blumstein, D. T., Mennill, D. J., Clemins, P., Girod, L., Yao, K., Patricelli, G., Deppe, J. L., Krakauer, A. H., Clark, C., Cortopassi, K. A., Hanser, S. F., McCowan, B., Ali, A. M., and Kirschel, A. N. G. (2011). Acoustic monitoring in terrestrial environments using microphone arrays: applications, technological considerations and prospectus. *Journal of Applied Ecology*, 48(3):758–767.

Bradski, G. (2000). The OpenCV library. *Dr. Dobb’s Journal of Software Tools*.

Brandes, S. (2008). Automated sound recording and analysis techniques for bird surveys and conservation. *Bird Conservation International*, 18:163–173.

Briggs, F., Lakshminarayanan, B., Neal, L., Fern, X. Z., Raich, R., Hadley, S. J., Hadley, A. S., and Betts, M. G. (2012). Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. *The Journal of the Acoustical Society of America*, 131(6):4640–4650.

Celis-Murillo, A., Deppe, J. L., and Ward, M. P. (2012). Effectiveness and utility of acoustic recordings for surveying tropical birds. *Journal of Field Ornithology*, 83(2):166–179.

Chabot, D. (1988). A quantitative technique to compare and classify humpback whale (megaptera novaeangliae) sounds. *Ethology*, 77(2):89–102.

Clark, C. W., Marler, P., and Beeman, K. (1987). Quantitative analysis of animal vocal phonology: an application to swamp sparrow song. *Ethology*, 76(2):101–115.

Fagerlund, S. (2007). Bird species recognition using support vector machines. *EURASIP Journal on Advances in Signal Processing*.

Fodor, G. (2013). The ninth annual mlsp competition: first place. In *Machine Learning for Signal Processing (MLSP)*, 2013 IEEE International Workshop on, pages 1–2. IEEE.

Grigg, G., Taylor, A., Mc Callum, H., and Watson, G. (1996). Monitoring frog communities: an application of machine learning. In *Proceedings of Eighth Innovative Applications of Artificial Intelligence Conference, Portland Oregon*, pages 1564–1569.

Gunasekaran, S. and Revathy, K. (2010). Content-based classification and retrieval of wild animal sounds using feature selection algorithm. *Second International Conference on Machine Learning and Computing*.

Hana, N. C., Muniandyb, S. V., and Dayoua, J. (2011). Acoustic classification of australian anurans based on hybrid spectral-entropy approach. *Applied Acoustics*, 72.

Henderson, E. and Hildebrand, J. A. (2011). Classification of behavior using vocalizations of pacific white-sided dolphins (lagenorhynchus obliquidens). *Acoustical Society of America*.

Lammers, M., Brainard, R., Au, W., Mooney, T. A., and Wong, K. (2008). An ecological acoustic recorder (ear) for long-term monitoring of biological and anthropogenic sounds on coral reefs and other marine habitats. *The Journal of the Acoustical Society of America*, 123:1720–1728.

Lasseck, M. (2013). Bird song classification in field recordings: winning solution for nips4b 2013 competition. In *Proc. of int. symp. Neural Information Scaled for Bioacoustics, sabiod. org/nips4b, joint to NIPS, Nevada*, pages 176–181.

Marques, T. A., Thomas, L., Martin, S. W., Mellinger, D. K., Ward, J. A., Moretti, D. J., Harris, D., and
Tyack, P. L. (2013). Estimating animal population density using passive acoustics. *Biological Reviews, 88*(2):287–309.

Mellinger, D. and Clark, C. (2000). Recognizing transient low-frequency whale sounds by spectrogram correlation. *The Journal of the Acoustical Society of America, 107*:3518–3529.

Sueur, J., Pavoine, S., Hamerlynck, O., and Duvail, S. (2008). Rapid acoustic survey for biodiversity appraisal. *PLoS ONE 3*:e4065.

Taylor, A. (1995). Bird flight call discrimination using machine learning. *The Journal of the Acoustical Society of America, 97*(5):3370–3370.

Terborgh, J., Robinson, S. K., III, T. A. P., Munn, C. A., and Pierpont, N. (1990). Structure and organization of an amazonian forest bird community. *Ecological Monographs, 60*:213–238.

Tricas, T. C. and Boyle, K. (2009). Validated reef fish sound scans of passive acoustic monitors on hawaiian coral reefs. *The Journal of the Acoustical Society of America, 125*:2589.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors (2014). scikit-image: image processing in Python. *PeerJ, 2*:e453.

Villanueva-Rivera, L. J. and Pijanowski, B. C. (2012). Pumilio: a web-based management system for ecological recordings. *Emerging Technologies, 93*:71–81.

Walther, G.-R., Post, E., Convey, P., Menzel, A., Parmesan, C., Beebee, T. J., Fromentin, J.-M., Hoegh-Guldberg, O., and Bairlein, F. (2002). Ecological responses to recent climate change. *Nature, 416*(6879):389–395.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on, 13*(4):600–612.

Xie, J., Michael, T., Zhang, J., and Roe, P. (2016). Detecting frog calling activity based on acoustic event detection and multi-label learning. *Procedia Computer Science, 80*:627–638.

Zhang, L., Towsey, M., Xie, J., Zhang, J., and Roe, P. (2016). Using multi-label classification for acoustic pattern detection and assisting bird species surveys. *Applied Acoustics, 110*:91–98.
Table 6. Area Under the ROC Curve (AUC), Accuracy (Ac), negative predictive value (Npv), precision (Pr), sensitivity (Se) and specificity (Sp) of the 21 species and three variants of the algorithm. Best values are shaded and the cases where the ANOVA suggested a significant difference between the algorithm variants at the 95% confidence level are in bold.
APPENDIX 2

In this Appendix we present the templates created by the training sets of each species. We classified them by the algorithm that presented a better accuracy for that species. Figure 10 presents the templates of the species where the SSIM variant presented better accuracy, Figure 11, those where the NORM variant presented better accuracy and, Figure 12, the species where the CORR variant presented better accuracy.

Templates of species that presented a better accuracy for the SSIM variant.

**Figure 10.** Sample of species that the SSIM variant presented better accuracy. a) *F. analis*, b) *M. hemimelaena*, c) *E. cooki*, d) *P. lophotes*, e) *E. coqui*, f) *T. schistaceus* and g) *H. subflava*. Species a, b and c are statistically significant.
Templates of species that presented a better accuracy for the NORM variant.

Figure 11. Sample of species that the NORM variant presented better accuracy. a) Unknown Insect, b) B. chrysogaster, c) S. grossus, d) M. guatemalae and e) E. juanariveroi. Neither is statistically significant.
Templates of species that presented a better accuracy for the CORR variant.

Figure 12. Sample of species that the CORR variant presented better accuracy. a) *E. cochranae*, b) *M. leucophrys*, c) *B. bivittatus*, d) *C. carmioli*, e) *M. marginatus*, f) *M. nudipes*, g) *E. brittoni*, h) *E. guttatus* and i) *L. thoracicus*. Species a, b and c are statistically significant.