Methods in Ecology and Evolution

BRITISH ECOLOGICAL SOCIETY

Methods in Ecology and Evolution 2017, **8,** 184–191 doi: 10.1111/2041-210X.12624

APPLICATION

warbleR: an R package to streamline analysis of animal acoustic signals

Marcelo Araya-Salas[1,2]* **and Grace Smith-Vidaurre**[3]

[1]Escuela de Biología, Universidad de Costa Rica, San Pedro, San Jose, UCR 11501-2060, Costa Rica; [2]Laboratory of Ornithology, Cornell University, 159 Sapsucker Woods Road, Ithaca, NY 14850, USA; and [3]Department of Biology, New Mexico State University, Las Cruces, NM 88003-8001, USA

Summary

1. Animal acoustic communication is one of the most fruitful research areas in behavioural and evolutionary biology. Work in this area depends largely on quantifying the structure of acoustic signals, which has often depended upon closed-source or graphical user interface (GUI)-based software.

2. Here, we describe the R package warbleR, a new package for the analysis of animal acoustic signal structure. The package offers functions for downloading avian vocalizations from the open-access online repository Xeno-Canto, displaying the geographic extent of the recordings, manipulating sound files, detecting acoustic signals or importing detected signals from other software, assessing performance of methods that measure acoustic similarity, conducting cross-correlations, measuring acoustic parameters and analysing interactive vocal signals, among others. Functions working iteratively allow parallelization to improve computational efficiency.

3. We present a case study showing how warbleR functions can be used in a workflow to evaluate the structure of acoustic signals. We analyse geographic variation in long-billed hermit hummingbirds (*Phaethornis longirostris*) songs obtained from Xeno-Canto.

4. The code in warbleR can be executed by less experienced R users, but has also been thoroughly commented, which will facilitate further customization by advanced users. The combination of the tools described here with other acoustic analysis packages in R should significantly expand the range of analytical approaches available.

Key-words: bioinformatics, evolutionary biology, software

Introduction

Animal acoustic communication has provided fruitful grounds for research in behaviour and evolutionary biology in the past five decades (Bradbury & Vehrencamp 2011). Studies of geographic variation (e.g. Keighley *et al*. 2016), female preferences (e.g. Lea & Ryan 2015) and variation among species (e.g. Medina-García, Araya-Salas & Wright 2015), among other topics, have brought critical insights regarding the evolution of signals and communication systems. This work largely relies on quantifying the structure of acoustic signals and comparison across behavioural contexts (e.g. Searcy, Anderson & Nowicki 2006), individuals (e.g. Araya-Salas 2012), groups (e.g. Gonzalez & Ornelas 2005) or species (e.g. Rodriguez *et al*. 2015). Hence, tools for measuring and comparing signal structure have been pivotal in the development of animal communication research.

Here, we describe the package warbleR, developed in the open-source R environment, which uses code and functions from the packages SEEWAVE (Sueur, Aubin & Simonis 2008), TUNER (Ligges *et al*. 2014) and MONITOR (Katz, Hafner & Donovan 2016), to increase the accessibility of bioacoustics analyses. warbleR combines the flexibility of R with the analysis of animal acoustic signal structure and open-access online resources, offering functions easily executed by less experienced R users and well-commented code for more innovation by advanced users. The package is available on CRAN (https://cran.r-project.org/web/packages/warbleR/index.html) and GitHub (https://github.com/maRce10/warbleR).

warbleR streamlines analyses of animal acoustic signal structure in R. Users can collect open-access avian recordings or input their own data into a workflow that facilitates spectrographic visualization and measurement of acoustic parameters. warbleR makes fundamental sound analysis tools from the R package SEEWAVE (Sueur, Aubin & Simonis 2008*)*, as well as new tools not yet offered in the R environment, readily available for batch process analysis of acoustic signals. The package offers functions for downloading recordings from Xeno-Canto (a community-sourced, open-access online repository with recordings for over 9300 avian species), creating maps of Xeno-Canto recordings, manipulating sound files, detecting signals or importing detected signals from other software, printing spectrograms of whole recordings or individual signals, conducting cross-correlations, assessing performance of methods that measure acoustic similarity, measuring acoustic parameters and analysing interactive vocal signals (e.g. duets; Table 1). Most functions allow task parallelization, which

*Correspondence author. E-mail: marceloa27@gmail.com distributes tasks across multiple processors for improving computational efficiency. Tools to evaluate analysis performance at each step are also available. All these tools are provided in a standardized workflow for the analysis of signal structure (Fig. 1), making them accessible to a wide range of users, including those without much knowledge of R. In addition, warbleR meets the need for rigorous, open-source bioacoustics analyses, facilitating research opportunities and innovation of further customized analyses.

warbleR workflow: a case study

We present a case study showing how warbleR functions can be used in a workflow to evaluate the structure of acoustic signals (Fig. 1). For brevity, not all warbleR functions are used, but see Table 1 for a complete list and the workflow diagram (Fig. 1) for alternative workflow paths.

This case study entails analysis of variation at a microgeographical scale (a few km range) in songs of long-billed hermit hummingbirds (*Phaethornis longirostris*) obtained from Xeno-Canto. This lekking hummingbird species exhibits a single song-type repertoire, characterized by vocal variation at small geographic scale (Araya-Salas & Wright 2013). The following workflow can be easily modified for evaluating acoustic structure in recordings from the Xeno-Canto repository or users' own recordings.

To install warbleR from CRAN and load it (the R code in this paper is also found in Appendix S1):

```
install.packages("warbleR")
library(warbleR)
```

Additional dependencies may be required on OSX and Linux operating systems. See the package vignette (https://cran.r-project.org/web/packages/warbleR/vignettes/warbleR_workflow.html) for instructions on how to install these dependencies.

DOWNLOAD XENO-CANTO RECORDINGS AND METADATA

Metadata and recordings can be downloaded from Xeno-Canto using a species or genus as a query.

```
Phae <- querxc(qword = "Phaethornis", download =
FALSE)
Phae.lon <- querxc(qword = "Phaethornis
longirostris", download = FALSE)
```

Xeno-Canto metadata is returned as a data frame and contains useful information for filtering recordings, like vocalization type, location, geographic coordinates and recording quality. Filtered metadata can be input back into *querxc* to download specific recordings. We recommend (i) downloading metadata, (ii) filtering based on desired attributes of the recordings (e.g. geographic range, quality) and then (iii) using filtered metadata to download the desired subset of recordings.

Maps can be created to visualize the geographic spread of Xeno-Canto recordings by species. The maps can be displayed in the graphics device or saved as image files in the working directory and are useful to double-check coordinate accuracy. Figure 2 shows the geographic spread of long-billed hermit recordings.

```
xcmaps(X = Phae.lon, img = FALSE)
```

Below, metadata is filtered by signal type, location and recording quality prior to downloading. High-quality songs (A

Table 1. Description of functions in the warbleR package, see functions' documentation for more detailed information

Function	Returned	Description
querxc	R data object; .mp3 files	Download recordings from Xeno-Canto
xcmaps	.jpeg or .tiff image files	Create maps of Xeno-Canto recordings
imp.raven	R data object	Import Raven selections
imp.syrinx	R data object	Import Syrinx selections
mp32.wav	.wav files	Convert .mp3 files to .wav
checkwavs	R console message	Check .wav files for compatibility
lspec	.jpeg or .tiff image files	Produce long spectrograms of whole recordings
autodetec	R data object; .jpeg or .tiff image files	Automatically detect signals within recordings
snrspecs	.jpeg or .tiff image files	Visually optimize margins used for signal-to-noise ratio analysis
sig2noise	R data object	Measure signal-to-noise ratio
manualoc	.csv file	Manually detect signals within recordings
seltailor	.csv file	Fix start/end coordinates of selections
specreator	.jpeg or .tiff image files	Create spectrograms of individual selections
compare.methods	.jpeg image file	Simultaneously compare two acoustic distance calculation methods
trackfreqs	.jpeg or .tiff image files	Visualize accuracy of frequency measurements
specan	R data object	Measure acoustic parameters across several files
x.corr	R data object	Perform spectrogram cross-correlation
xcorr.graph	Plot in graphics device	Pairwise plots of spectrogram cross-correlation scores
dfts	R data object; .jpeg or .tiff image files	Extract dominant frequency values as time series
dfDTW	R data object; .jpeg or .tiff image files	Calculate acoustic dissimilarity using dynamic time warping on dominant frequency contours
ffts	R data object; .jpeg or .tiff image files	Extract fundamental frequency values as time series
ffDTW	R data object; .jpeg or .tiff image files	Calculate acoustic dissimilarity using dynamic time warping on fundamental frequency contours
coor.test	R data object	Randomization test for coordinated singing
coor.graph	R data object; .jpeg or .tiff image files	Visualize coordination of acoustic signals

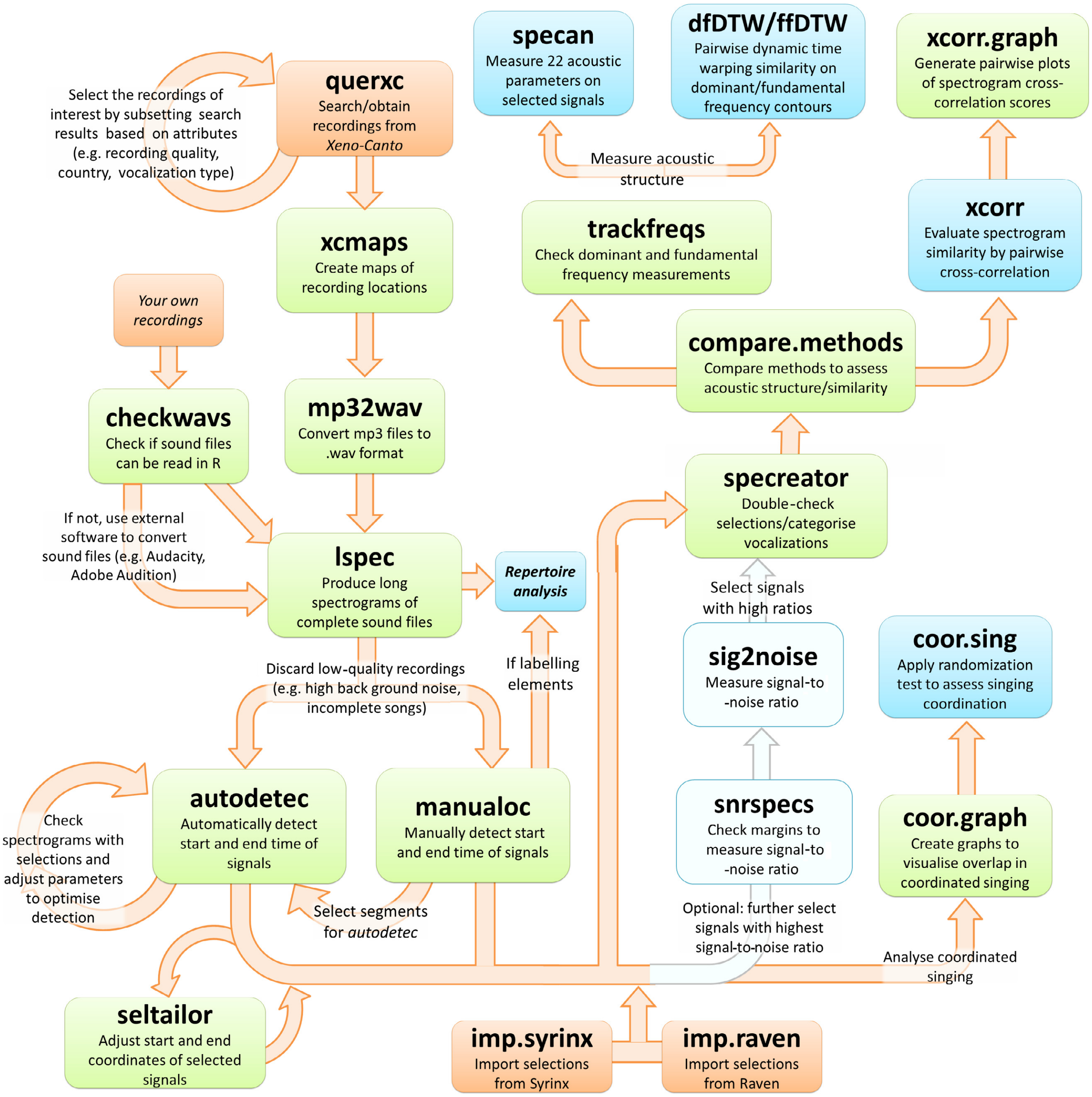
querxc
Search/obtain recordings from *Xeno-Canto*
Select the recordings of interest by subsetting search results based on attributes (e.g. recording quality, country, vocalization type)
xcmaps
Create maps of recording locations
mp32wav
Convert mp3 files to .wav format
Your own recordings
checkwavs
Check if sound files can be read in R
If not, use external software to convert sound files (e.g. Audacity, Adobe Audition)
lspec
Produce long spectrograms of complete sound files
Repertoire analysis
If labelling elements
Discard low-quality recordings (e.g. high back ground noise, incomplete songs)
autodetec
Automatically detect start and end time of signals
Check spectrograms with selections and adjust parameters to optimise detection
manualoc
Manually detect start and end time of signals
Select segments for *autodetec*
seltailor
Adjust start and end coordinates of selected signals
imp.syrinx
Import selections from Syrinx
imp.raven
Import selections from Raven
specan
Measure 22 acoustic parameters on selected signals
dfDTW/ffDTW
Pairwise dynamic time warping similarity on dominant/fundamental frequency contours
Measure acoustic structure
xcorr.graph
Generate pairwise plots of spectrogram cross-correlation scores
trackfreqs
Check dominant and fundamental frequency measurements
xcorr
Evaluate spectrogram similarity by pairwise cross-correlation
compare.methods
Compare methods to assess acoustic structure/similarity
specreator
Double-check selections/categorise vocalizations
Select signals with high ratios
sig2noise
Measure signal-to -noise ratio
snrspecs
Check margins to measure signal-to -noise ratio
Optional: further select signals with highest signal-to-noise ratio
coor.sing
Apply randomization test to assess singing coordination
coor.graph
Create graphs to visualise overlap in coordinated singing
Analyse coordinated singing

Fig. 1. The warbleR workflow provides functions for flexible and streamlined analysis of the structure of acoustic signals. Different functions provide alternative ways to import recordings, detect signals or analyse acoustic similarity, depending on the data and research question. Orange nodes are steps in which data or recordings are input in the workflow, while blue nodes represent functions for assessing or comparing acoustic structure. Green nodes depict manipulation of recordings or evaluation of analysis accuracy, and sky-blue nodes illustrate optional steps to filter recordings.

score) are selected from a single location (La Selva Biological Station, Sarapiqui, Heredia, Costa Rica).

```
Phae.lon.song<-Phae.lon[grep(“song”,Phae.lon
$Vocalization_type,ignore.case=TRUE),]
Phae.lon.song<-Phae.lon.song[grep(“Sarapiqui,
Heredia”,Phae.lon.song$Locality,ignore.case=
FALSE),]
Phae.lon.song<-Phae.lon.song[Phae.lon.song
$Quality==“A”,]
querxc(X=Phae.lon.song,download=TRUE)
```

Xeno-Canto files are maintained in .mp3 format and need to be converted to .wav format for further analyses. *mp32wav* converts all .mp3 files in the working directory and can be used to lower sampling rates.

```
mp32wav(samp.rate=22.05)
```

CREATE LONG SPECTROGRAMS FOR VISUAL INSPECTION AND SIGNAL CLASSIFICATION

lspec creates long spectrograms of whole recordings, facilitating visual examination of recording quality and overall signal

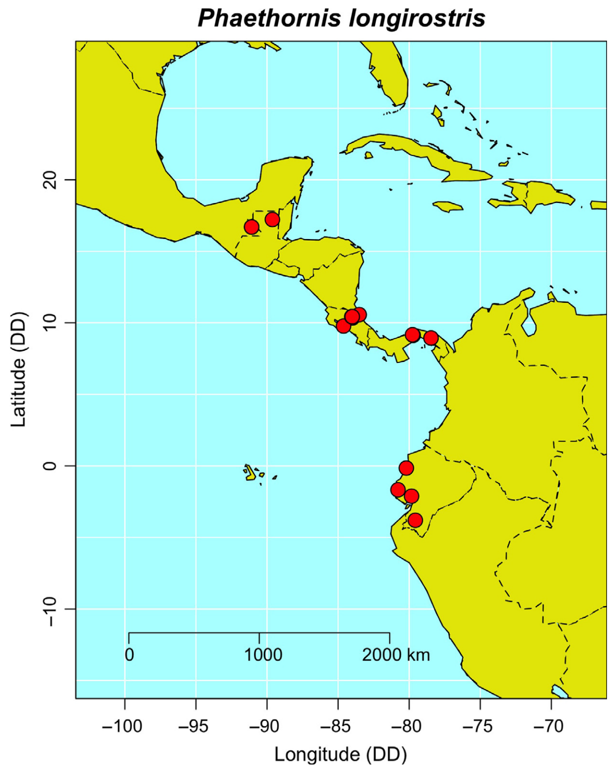

Fig. 2. Map of long-billed hermit recordings downloaded from Xeno-Canto, demonstrating the geographic spread and coordinate accuracy of available recordings.

structure. These spectrograms are displayed as rows in letter-size image files (8″ × 11·5″) saved in the working directory. The number of rows and seconds per row can be adjusted. Several pages will be generated for long recordings. *lspec* can also be used for visual classification of signals (e.g. repertoire analyses).

```
lspec(flim = c(1.5, 11), ovlp = 10, sxrow = 6, rows = 15)
```

DETECT OR SELECT SIGNALS WITHIN RECORDINGS

Unlike other acoustic analysis software (e.g. Raven), functions for signal structure analysis in warbleR require only the temporal coordinates of signals as input, not the frequency range. These temporal coordinates can be automatically determined using *autodetec*. Frequencies can be detected within a specified frequency band using the *bandpass* (*bp*) argument. If a bandpass filter is specified, *autodetec* will first detect signals in the given frequency band, then will subsequently select signals above the given amplitude threshold. The function can also detect signals within a range of duration if the *mindur* and *maxdur* arguments are specified.

autodetec produces two types of output: a data frame containing start and end coordinates of selected signals, and a spectrogram of the entire recording marked with the temporal coordinates of selected signals, which can be used to assess detection performance.

```
Phae.ad <- autodetec(bp = c(2, 9), threshold = 20,
mindur = 0.09, maxdur = 0.22, ssmooth = 900, ls = TRUE,
res = 100, flim = c(1, 12), wl = 300, set = TRUE, sxrow =
6, rows = 15)
```

Signals can also be manually selected with *manualoc*. This function produces an interactive display of the spectrograms, through which users can select the start and end of signals. All recordings in the working directory are displayed sequentially, and the temporal coordinates are saved in a single .csv file. Alternatively, warbleR can be used to import selections made in Syrinx (J. Burt, WA, USA) or Raven (Cornell Laboratory of Ornithology, Ithaca, NY, USA) with *imp.syrinx* and *imp.raven*.

```
manualoc(flim = c(1, 11), wl = 300)
```

Selections produced in warbleR or imported from other software can be further adjusted using the *seltailor* function.

FILTER BY SIGNAL-TO-NOISE RATIO

Selections can be filtered by signal-to-noise ratio using *sig2noise*. This step helps discard low-quality (e.g. high background noise) selections. In our example, only a few signals per recording are needed here, given the single song-type repertoire of long-billed hermits. The following code measures signal-to-noise ratio and then selects the signals with the highest signal-to-noise radio for each recording:

```
Phae.snr <- sig2noise(X = Phae.ad[seq(1, nrow
(Phae.ad), 2), ], mar = 0.04)
Phae.hisnr <- Phae.snr[ave(-Phae.snr$SNR, Phae.snr
$sound.files, FUN = rank) <= 5, ]
```

MEASURE ACOUSTIC PARAMETERS AND ACOUSTIC SIMILARITY

warbleR offers three approaches to assess the structure of acoustic signals: (i) *specan* measures frequency, time and amplitude parameters (Table S1, Supporting Information), (ii) *dfDTW* and *ffDTW* measure pairwise acoustic dissimilarity based on the dominant or fundamental frequency contours, respectively, and (iii) *xcorr* performs pairwise spectrogram cross-correlations. These functions take *manualoc* or *autodetec* data frames as input (or signals imported from external software) and can handle large batches of recordings. Importantly, users must carefully evaluate which of the above methods perform best for signals in their study system. The performance of the approaches can be directly compared using the *compare.methods* function.

specan measures 22 acoustic parameters (see Table S1) on signals listed in the input data frame, corresponding to sound files in the working directory.

```
params <- specan(Phae.hisnr, bp = c(1, 11), threshold
= 15)
```

dfDTW and *ffDTW* calculate pairwise acoustic dissimilarity by applying dynamic time warping (DTW) analysis on dominant or fundamental frequency contours, respectively (Fig. 3).

Briefly, dynamic time warping compares the alignment of two sequences (in this case dominant frequency values), returning a single dissimilarity score. This analysis has been successfully used in pairwise comparisons of bird songs (Kogan & Margoliash 1998). *dfDTW* and *ffDTW* also return image files of spectrograms overlaid with frequency contours.

```
tsLBH <- dfDTW(Phae.hisnr, length.out = 30, flim = c
(1, 12), picsize = 2, res = 100, bp = c(2, 9))
```

Figure 3 shows song-type grouping based on dynamic time warping dissimilarities. As long-billed hermit song types vary at a small geographic scale, it is common to find multiple distinct song types within a single lek (Araya-Salas & Wright 2013). Note that the frequency contours can be directly extracted as time series using the *dfts* and *ffts* functions. These contours can be used to calculate additional signal attributes (e.g. start and end frequencies, ‘frequency excursion index’ (Podos *et al.* 2016)).

Spectrographic cross-correlation is another powerful tool to evaluate acoustic signal similarity (Clark, Marler & Beaman 1987). This method ‘slides’ one spectrogram (matrix of amplitudes in each time-frequency bin) over the other, calculating a correlation of the amplitude values at each step. The *xcorr* function runs pairwise cross-correlations on several signals and returns a list of objects, including the correlation statistic for each ‘sliding’ step, as well as the maximum (peak) correlation for each pairwise comparison.

```
Phae.hisnr1 <- Phae.snr[ave(-Phae.snr$SNR,
Phae.snr$sound.files,
FUN = rank) <= 1, ]
xcor <- xcorr(X = Phae.hisnr1, wl = 300, frange = c(2,
9),
ovlp = 90, dens = 0.9, wn = “hanning”, cor.method =
“pearson”)
```

Pairwise signal similarity yielded by cross-correlation can be visualized as a matrix (Fig. 4).

```
xcorr.graph(X = xcor, cex.cor = 1, cex.lab = 1, rel.cex
= FALSE, labs = gsub(“[^0-9]”, “”, Phae.hisnr1
$sound.files))
```

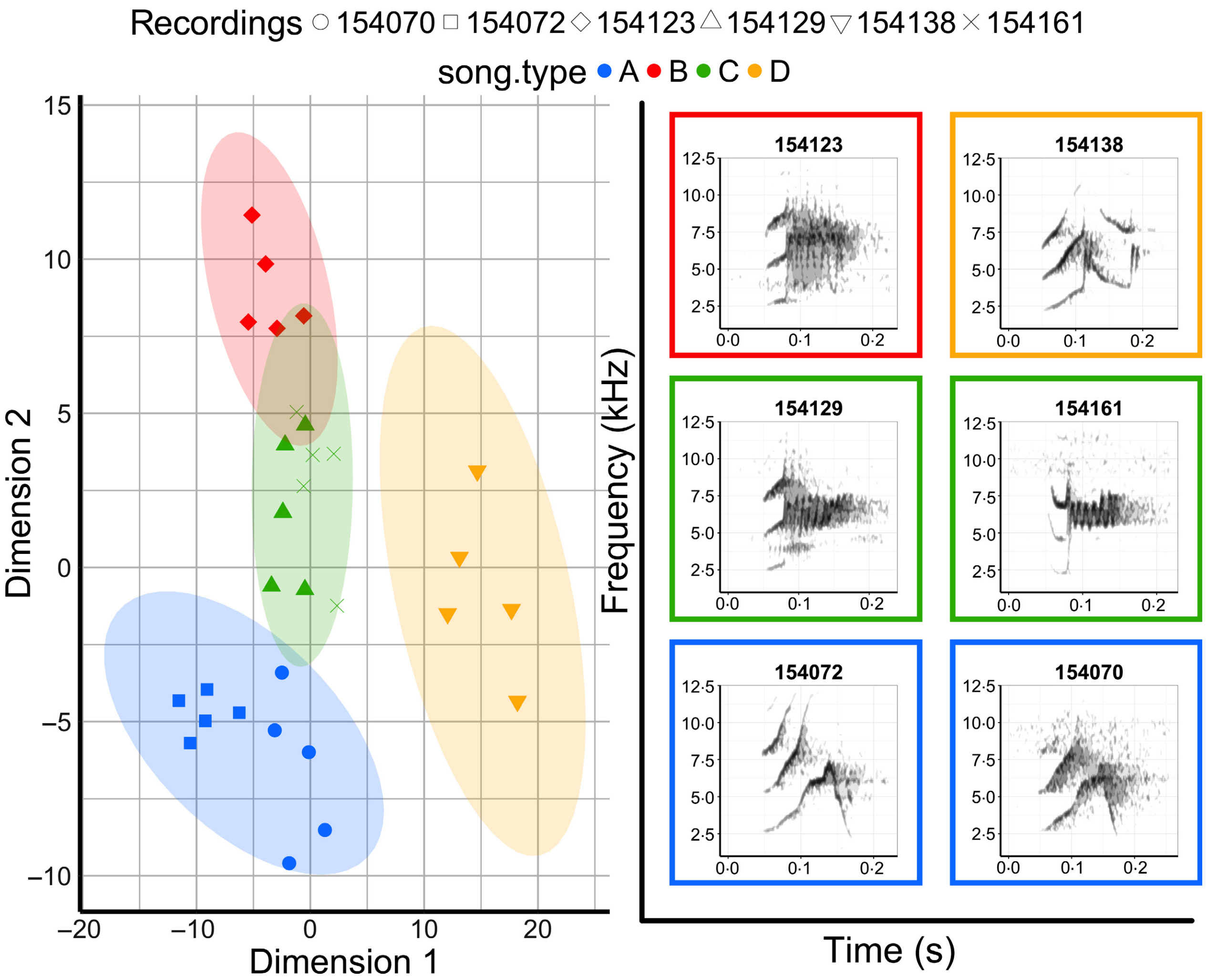

Fig. 3. Grouping of long-billed hermit songs by similarity of dominant frequency contours. Similarity was assessed using dynamic time warping. The scatterplot is based on the two axes from a classic multidimensional scaling. Spectrograms were produced with the R package SEEWAVE (Sueur, Aubin & Simonis 2008).

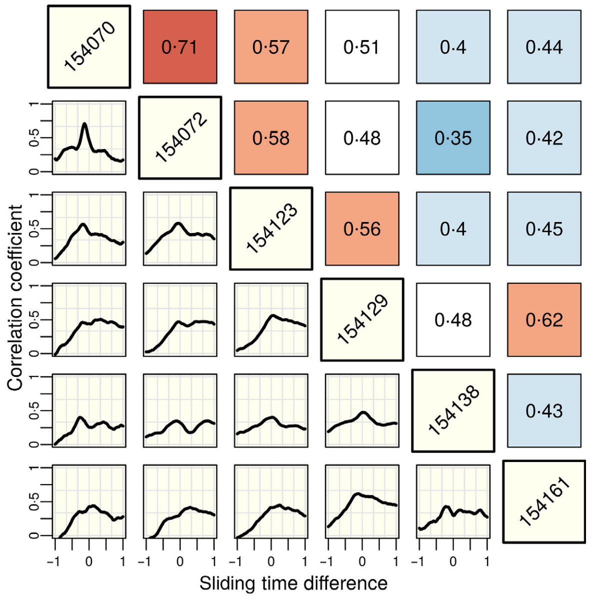

Fig. 4. Matrix of pairwise correlation coefficients plotted against time-sliding steps in cross-correlation analysis, produced by *xcorr.-graph*. The top triangle of the matrix illustrates strength and direction of correlations between spectrograms (high positive correlations in red). The lower triangle of the matrix demonstrates the time difference between signals for each sliding step, in which 0 represents two perfectly centred signals.

The three methods described above require the same input data, facilitating the application and comparison of different analytical approaches on the same data sets.

Additional bioacoustics analyses

INTERACTIVE SINGING ANALYSIS

Acoustic signals are often delivered in an interactive fashion when two or more signalling individuals overlap their active space. However, tools for the analysis of these signals are scarce. Indeed, the statistical treatment of these signals has been the subject of some controversy (Searcy & Beecher 2009, 2011; Naguib & Mennill 2010). Randomization tests have been proposed as the sole reliable method (Searcy & Beecher 2011; Helfer & Osiejuk 2015). Few analytical tools, however, have been designed to study interactive signals (but see Masco *et al.* 2016). warbleR facilitates the visualization of interactive signalling (Fig. 5) and the evaluation of overlapping or alternating patterns.

```
data(sim.coor.sing)
coor.graph(X = sim.coor.sing, ovlp = TRUE, only.coor
= FALSE, xl = 2, res = 80)
```

The occurrence of interactive singing strategies can be evaluated using the function *coor.test*. This function calculates the probability of finding an equal or more extreme number of overlaps in an interactive singing event. The function shuffles sequences of signals and silent intervals of both individuals to produce a null distribution of number of overlaps expected by chance. The p-value is calculated as the proportion of random expected values equal to or more extreme than the observed value. The test is equivalent to the 'KeepGaps' method described in Masco *et al.* (2016). The function runs one test for each singing event in the input data frame. The example data *sim.coor.sing* included in warbleR contains simulated data sets with three possible patterns: uncoordinated, overlapping and alternating (Fig. 5).

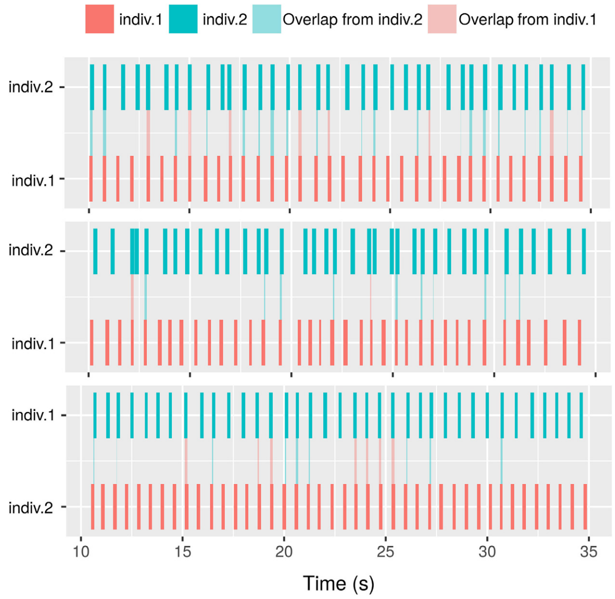

Fig. 5. Graphics created by the *coor.graph* function showing three types of simultaneous singing events: overlapping (top), alternating (middle) and uncoordinated (bottom).

```
coor.test(sim.coor.sing, iterations = 100,
less.than.chance = TRUE)
```

Classification of Signal Elements

Many bioacoustics analyses are targeted to the syntax of multi-element signals. warbleR functions provide tools to facilitate the classification of song elements. Segmenting songs into their basic elements can be done with *manualoc* and *autodetec,* which provide the start and end coordinates of the elements. Element types can be labelled while running the *manualoc* function with the *selcomm* argument. Alternatively, element types can be visually classified by looking at the output images from *lspec* and *specreator*. The output of *specan*, *dfDTW, ffDTW* or *xcorr* can be used as input for exploratory data analyses (e.g. principal components analysis); the clustering of signals detected by these techniques can be used to define syllable categories.

Assessing analysis reliability with warbleR

Many recordings include background noise and reverberation that compromise precision in signal detection and

Table 2. Performance testing of warbleR functions. Performance was measured as mean duration in seconds (from 50 iterations) in processing middle size data sets (100 recordings, 100 selections or 100 singing events) using warbleR example files. Functions showing results for only 1 core do not allow for parallel computing. Parallelization is not available for some functions in Windows (e.g. dfDTW) as it does not improve performance. Tests were performed on a 64-bit Linux or Windows operating system, using an 8-core 31GB RAM CPU

	1 core (no parallel)		2 cores		4 cores	
Function	Linux	Windows	Linux	Windows	Linux	Windows
autodetec	70·25	56·42	35·63	32·36	21·41	24·24
checkwavs	0·02	0·08				
compare.methods	1·36	1·74	1·11	8·76	1·17	13·78
coor.graph	11·93	13·83				
coor.test	87·01	85·03	42·71	44·10	26·99	31·13
dfDTW	8·74	10·42	7·06		6·59	
dfts	3·39	3·79	1·52		1·13	
ffDTW	6·62	7·17	5·59		5·28	
ffts	1·46	1·78	0·81		0·61	
imp.raven	0·15	0·26				
imp.syrinx	0·18	0·52				
lspec	27·63	30·34	13·72	22·81	9·00	18·24
sig2noise	0·31	0·28	0·25		0·31	
snrspecs	15·77	22·00	7·94	15·52	6·42	13·10
specan	11·21	11·57	5·27	8·24	3·26	8·16
specreator	11·21	17·11	5·71	12·82	3·67	11·67
trackfreqs	15·66	24·41	8·09	17·31	5·30	13·82
xcmaps	2·12	3·75				
xcorr	2·35	2·42	1·82		1·27	

measurement. warbleR functions are not immune to this issue. Nonetheless, the package offers quality control tools that can help detect inconsistencies in the analyses. These tools generate spectrograms superimposed with measurements or labels which can be used to visually inspect the reliability of the analyses. For instance, *lspec* and *specreator* can be used to evaluate consistency in element type classification (as element labels are displayed in the image title). *trackfreqs* can be used to inspect precision of frequency measurements. In addition, the performance of methods measuring acoustic similarity can be compared using *compare.methods*. Users can also make use of function options (i.e. arguments) to fine-tune their analyses (e.g. bandpass filters in most warbleR functions exclude background noise outside the frequency range of the signal, see *autodetec* example above). Examples demonstrating the application of these tools are available in the package vignette.

Computational performance

Most warbleR functions designed to run iteratively, or over batches of files, allow for task parallelization (using the argument *parallel*). This feature can significantly improve computational speed. In some cases, parallel computing runs more than three times faster than the non-parallel process (an average of 2·4 times faster when using 2 cores for parallelization), depending on computing capacity (see Table 2).

Conclusions

We have shown how warbleR functions can be used in a workflow, from obtaining recordings to assessing acoustic variation. For brevity, only the functions directly involved in manipulating recordings or obtaining acoustic measurements were included in the workflow description (but see Fig. 1), and only the most basic options of those functions were described. We would like to stress that an important feature of acoustic analysis in R (and any analysis in R) lies in allowing in-depth quality control throughout the process. Indeed, warbleR offers several functions that exploit this potential. We encourage users to routinely incorporate these tools in their analyses.

By providing direct access to the Xeno-Canto birdsong repository, warbleR can promote the use of this important resource in research. In addition, warbleR code has been well-commented, allowing advanced users to further customize their analyses in the flexible R environment. We hope that the bioacoustics tools described here, in combination with other R packages, will help to expand the range of analytical approaches, while promoting more cautious bioacoustics analyses and increasing the accessibility of these tools.

Acknowledgements

We thank the Wright laboratory and Hua Zhong for suggestions during warbleR development, William-Pier Vellinga for access to the Xeno-Canto collection (www.xeno-canto.org) and the developers of the packages seewave, monitor and tuner for their contributions. The authors declare no conflicts of interest.

Data accessibility

The *Phaethornis longirostris* recordings used in this paper are freely available in the Xeno-Canto repository (recordist: Marcelo Araya-Salas, recording URLs: http://www.xeno-canto.org/154161, http://www.xeno-canto.org/154138, http://www.xeno-canto.org/154129, http://www.xeno-canto.org/154123, http://www.xeno-canto.org/154072, http://www.xeno-canto.org/154070).

References

Araya-Salas, M. (2012) Is birdsong music? Evaluating harmonic intervals in songs of a Neotropical songbird. *Animal Behaviour*, **84**, 309–313.

Araya-Salas, M. & Wright, T. (2013) Open-ended song learning in a hummingbird. *Biology Letters*, **9**, 20130625.

Bradbury, J. & Vehrencamp, S. (2011) *Principles of Animal Communication*. Sinauer Associates Inc, Sunderland, MA.

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

Gonzalez, C. & Ornelas, J.F. (2005) Song structure and microgeographic song variation in the wedge-tailed sabrewings (*Campylopterus curvipennis*) in Veracruz, Mexico. *The Auk*, **122**, 593–607.

Helfer, B. & Osiejuk, T.S. (2015) It takes all kinds in acoustic communication: a new perspective on the song overlapping phenomenon. *Ethology*, **120**, 1–12.

Katz, J., Hafner, S.D. & Donovan, T. (2016) Tools for automated acoustic monitoring within the R package monitoR. *Bioacoustics*, **25**, 197–210.

Keighley, M.V., Langmore, N.E., Zdenek, C.N. & Heinsohn, R. (2016) Geographic variation in the vocalizations of Australian palm cockatoos (*Probosciger aterrimus*)*. *Bioacoustics*, **25**, 1–18.

Kogan, J. & Margoliash, D. (1998) Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: a comparative study. *The Journal of the Acoustical Society of America*, **103**, 2185–2196.

Lea, A. & Ryan, M.J. (2015) Irrationality in mate choice revealed by túngara frogs. *Science*, **349**, 964–966.

Ligges, U., Krey, S., Mersmann, O. & Schnackenberg, S. (2014) tuneR: analysis of music. R package, URL: https://cran.r-project.org/package=tuneR [accessed 15 July 2016].

Masco, C., Allesina, S., Mennill, D.J. & Pruett-Jones, S. (2016) The Song Overlap Null model Generator (SONG): a new tool for distinguishing between random and non-random song overlap. *Bioacoustics*, **25**, 29–40.

Medina-García, A., Araya-Salas, M. & Wright, T.F. (2015) Does vocal learning accelerate acoustic diversification? Evolution of contact calls in Neotropical parrots. *Journal of Evolutionary Biology*, **28**, 1782–1792.

Naguib, M. & Mennill, D.J. (2010) The signal value of birdsong: empirical evidence suggests song overlapping is a signal. *Animal Behaviour*, **80**, e11–e15.

Podos, J., Moseley, D.L., Goodwin, S.E., McClure, J., Taft, B.N., Strauss, A.V.H., Rega-Brodsky, C. & Lahti, D.C. (2016) A fine-scale, broadly applicable index of vocal performance: frequency excursion. *Animal Behaviour*, **116**, 203–212.

Rodriguez, R.L., Araya-Salas, M., Gray, D.A., Reichert, M.S., Symes, L.B., Wilkins, M.R., Safran, R.J. & Hobel, G. (2015) How acoustic signals scale with individual body size: common trends across diverse taxa. *Behavioral Ecology*, **26**, 168–177.

Searcy, W., Anderson, R. & Nowicki, S. (2006) Bird song as a signal of aggressive intent. *Behavioral Ecology and Sociobiology*, **60**, 234–241.

Searcy, W. & Beecher, M.D. (2009) Song as an aggressive signal in songbirds. *Animal Behaviour*, **78**, 1281–1292.

Searcy, W. & Beecher, M.D. (2011) Continued scepticism that song overlapping is a signal. *Animal Behaviour*, **81**, e1–e4.

Sueur, J., Aubin, T. & Simonis, C. (2008) Equipment review: seewave, a free modular tool for sound analysis and synthesis. *Bioacoustics*, **18**, 213–226.

Received 30 April 2016; accepted 20 July 2016
Handling Editor: Nick Golding

Supporting Information

Additional Supporting Information may be found online in the supporting information tab for this article:

Table S1. 22 acoustic parameters measured by *specan, which can be utilized to compare signal structure*.

Appendix S1. Code contained within warbleR*: an* R *package to streamline analysis of animal acoustic signals*.