Received: 28 April 2021 | Accepted: 23 August 2021

DOI: 10.1111/2041-210X.13711

Methods in Ecology and Evolution BRITISH ECOLOGICAL SOCIETY

APPLICATION

scikit-maad: An open-source and modular toolbox for quantitative soundscape analysis in Python

Juan Sebastián Ulloa[1,2,3] | Sylvain Haupert[2] | Juan Felipe Latorre[3] | Thierry Aubin[4] | Jérôme Sueur[2]

[1]Instituto de Investigación de Recursos Biológicos Alexander von Humboldt, Bogotá, Colombia

[2]Institut Systématique Évolution Biodiversité (ISYEB), Muséum National d'Histoire Naturelle, CNRS, Sorbonne Université, EPHE, Paris, France

[3]Departamento de Física, Universidad Nacional de Colombia, Bogotá, Colombia

[4]Équipe Communications Acoustiques, UMR 9197, Neuro-PSI-CNRS, Université Paris-Saclay, Orsay, France

Correspondence
Juan Sebastián Ulloa
Email: julloa@humboldt.org.co

Funding information
JSU was funded by the Colombian Ministry of Environment and Sustainable Development (resolución 0051, 2021) and by Minciencias Postdoctoral Grant (811-2018).

Handling Editor: Giovanni Strona

Abstract

1. Passive acoustic monitoring is increasingly being applied to terrestrial, marine and freshwater environments, providing cost-efficient methods for surveying biodiversity. However, processing the avalanche of audio recordings remains challenging, and represents nowadays a major bottleneck that slows down its application in research and conservation.
2. We present scikit-maad, an open-source Python package dedicated to the analysis of environmental audio recordings. This package was designed to (a) load and process digital audio, (b) segment and find regions of interest, (c) compute acoustic features and (d) estimate sound pressure levels. The package also provides field recordings and a comprehensive online documentation that includes practical examples with step-by-step instructions for beginners and advanced users.
3. scikit-maad opens the possibility to efficiently scan large audio datasets and easily integrate additional machine learning Python packages into the analysis, allowing to measure acoustic properties and identify key patterns in all kinds of soundscapes. To support reproducible research, the package is released under the BSD open-source licence, which allows unrestricted redistribution for commercial and private use.
4. This development will create synergies between the community of ecoacousticians, such as engineers, data scientists, ecologists, biologists and conservation practitioners, to explore and understand the processes underlying the acoustic diversity of ecological systems.

KEYWORDS

acoustic indices, bioacoustics, ecoacoustics, pattern recognition, sound pressure level

1 | INTRODUCTION

New sensor technology is changing the way we monitor ecological systems, collecting vast amounts of environmental data that need to be processed to support research and decision-making (Turner, 2014). In particular, passive acoustic monitoring used in ecoacoustics is being applied more and more widely in terrestrial, marine and freshwater environments. Such monitoring provides cost-efficient and non-invasive methods for surveying biodiversity, opening new scientific pathways for studies in ecology and evolution (Sugai et al., 2019). While passive acoustic sensors facilitate data collection, processing the avalanche of audio recordings is still challenging, and represents nowadays a bottleneck that slows down its application in research and conservation (Gibb et al., 2018).

Proprietary software solutions, such as Kaleidoscope (Wildlife Acoustics), Raven Pro (Bioacoustics Research Program, The Cornell Lab of Ornithology) and Avisoft-SASLab Pro (Avisoft Bioacoustics), have been developed to facilitate the analysis of passive acoustic recordings. However, the development of open-source solutions is essential for the advancement of reproducible methods. Open-source code, like the ecoacoustic software released by Queensland University of Technology (Towsey et al., 2020), removes an important barrier imposed by paid proprietary software, democratizing access to state-of-the-art developments and promoting collaboration between researchers. Because of their application in ecological research, most of the dedicated packages for environmental sound analysis have been developed in R (http://www.r-project.org/), such as seewave (Sueur, 2018; Sueur et al., 2008), monitoR (Katz et al., 2016) and warbleR (Araya-Salas & Smith-Vidaurre, 2017). While these are all valuable tools, the ecoacoustic community could find additional benefits from alternatives in other open-source languages, such as Python (http://www.python.org/), which are growing rapidly and provide a suitable environment for scientific computing.

Thanks to a large developer community effort, Python provides fundamental packages for scientific computing and digital signal processing. Numpy (Oliphant, 2007) is the core library that allows efficient numerical computation between matrices and high-level mathematical functions. The SciPy ecosystem (Virtanen et al., 2020) with SciKits add-on packages provides high-level functionalities for digital signal processing. Finally, the Matplotlib library offers an object-oriented application programming interface to plot and visualize multidimensional arrays. These libraries cover the basic functionalities of MATLAB (The Mathworks Inc.) but remain open source, unfolding vast possibilities for audio analysis in Python.

Previous open-source projects have developed toolboxes for audio analysis, such as essentia (Bogdanov et al., 2013), Librosa (McFee et al., 2015) and pyAudioAnalysis (Giannakopoulos, 2015), offering robust and efficient solutions for analysing speech and music from digital recordings. However, soundscapes, which encompass acoustic signals produced by humans, but also by other living organisms and geophysical processes, have unique acoustic characteristics that require custom-made tools. Notable examples of such tools developed by the community of bio- and ecoacousticians include (a) mathematical summaries of energy distribution (i.e. acoustic indices) aimed at characterizing animal acoustic communities and reflect different components of soundscapes (Buxton et al., 2018; Sueur et al., 2014; Towsey et al., 2014), (b) methods for spectro-temporal signal segmentation and characterization for multi-species signal classification (Ulloa et al., 2018) and (c) calibrated spectrograms and statistical analyses of sound levels to link digital signals with physical measurements (Merchant et al., 2015). Such functionalities are essentials to build efficient soundscape analysis pipelines for biodiversity assessment, but to date, there is no open-source package that integrates these analytical tools.

Here, we introduce scikit-maad (acronym of Multiresolution Analysis of Acoustic Diversity), an open-source Python package dedicated to the analysis of terrestrial or aquatic soundscapes. This package was designed to analyse environmental audio recordings and bring flexibility to (a) load and process digital audio, (b) segment and find regions of interest, (c) compute acoustic features and (d) estimate the sound pressure level of acoustic events. This workflow opens the possibility to efficiently scan large audio datasets and easily integrate additional machine learning Python packages into the analysis, allowing to measure acoustic properties and identify key patterns in all kinds of soundscapes.

2 | PACKAGE STRUCTURE

2.1 | Design principles

To provide an accessible and scalable toolbox, we prioritized five principles. First, to make the package readily available for researchers familiar with R and MATLAB, we opted for a flat package layout, with few abstract classes. For array computing and data visualization, we relied on the popular package trilogy for scientific computing in Python, namely Numpy, Scipy and Matplotlib. We opted to use DataFrames from the Pandas library (McKinney, 2010) to deliver outputs from heterogeneous tabular data and ease data exchange with R, which is more popular in the ecoacoustic and bioacoustic community. For image processing functionalities, we relied on scikit-image, a comprehensive library which performs efficient bidimensional segmentation and filtering (van der Walt et al., 2014). Second, to simplify new developments, we strived to follow the style guide for Python programming (PEP8) and use standardized variable names across the package. Third, to simplify the search, use and maintenance of the code, we worked on a comprehensive documentation of the functions included in the package and made it available online (https://scikit-maad.github.io). This documentation has an intuitive web design to navigate through the modules and includes an example gallery for both beginners and advanced users. Fourth, to stay abreast in a rapidly growing field of study, we worked on a modular implementation that separates the functionalities of the analytical workflow into interchangeable modules allowing rapid prototyping. Finally, to support reproducible research, we released this package under the BSD 3-clause licence, which allows unrestricted redistribution for commercial and private use as long as copyright notices and disclaimers of warranty are maintained.

Following an intuitive workflow, the package has four main modules: *sound*, *rois*, *features* and *spl* (Figure 1). The module *sound* allows to load and pre-process audio files. Once loaded, the signal can be segmented into discrete regions with high-energy density, or regions of interest (ROIs), with the module *rois*. Then, the signals can be characterized in time and frequency using the module *features*. The complementary module *spl* estimates sound pressure levels based on the signal amplitude and the recording parameters of the acoustic sensor. Furthermore, a module designed to model sound propagation and estimate the detection distance of a recorder is currently under

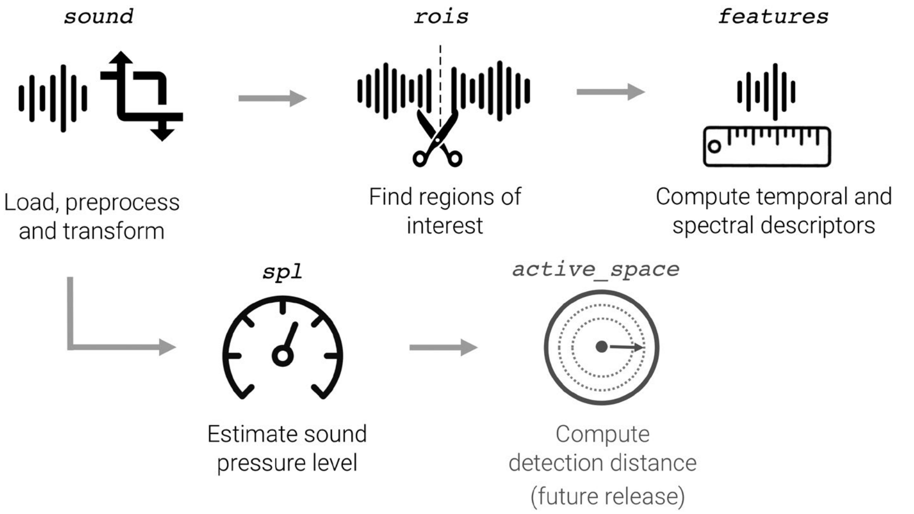

FIGURE 1 Soundscape analysis workflow and scikit-maad modules. The package is composed of four main modules (*sound*, *rois*, *features* and *spl*), each dedicated to a particular step in the workflow. A fifth module, *active_space*, will be released soon. All modules are fully interoperable, allowing researchers to perform advanced audio analyses

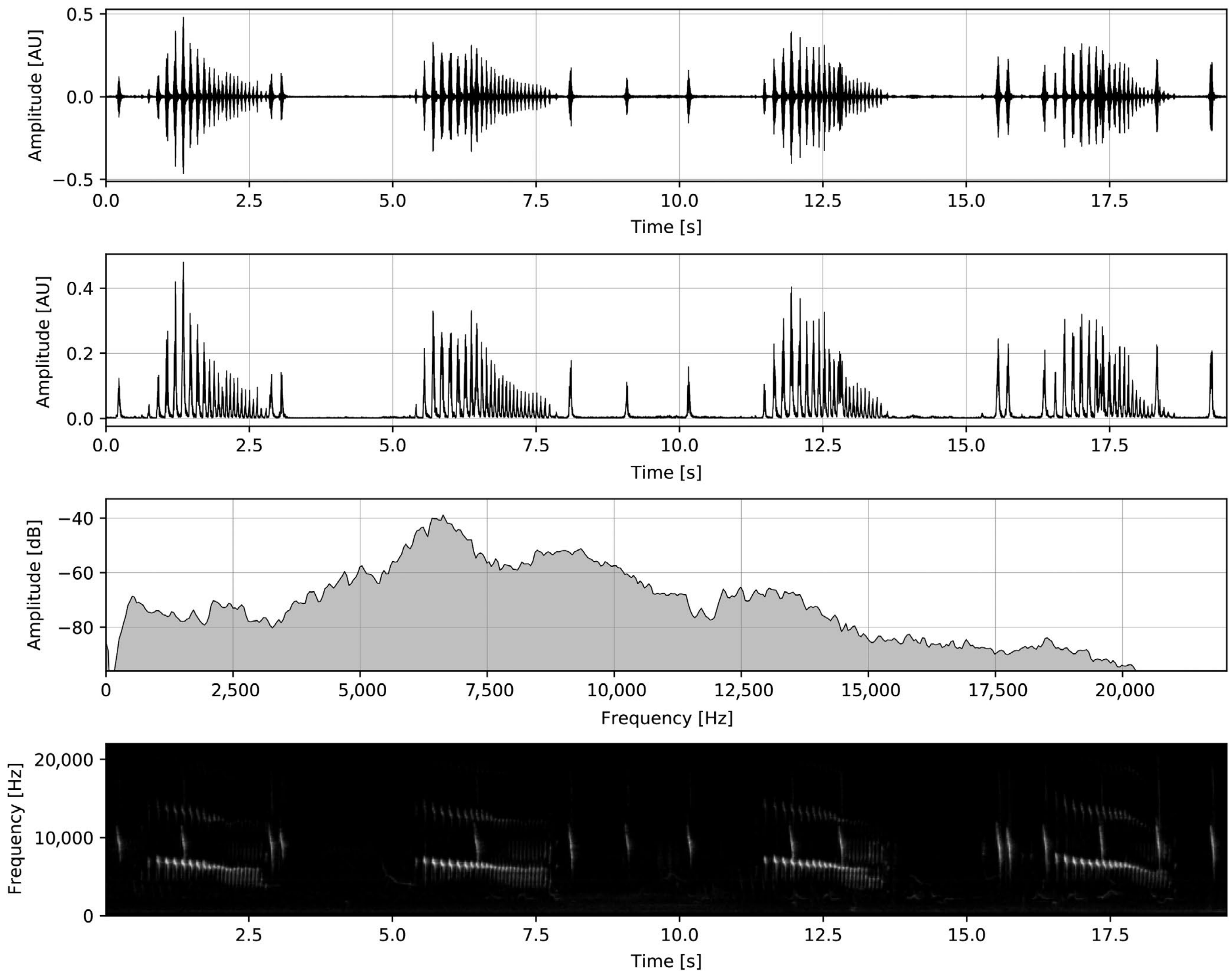

FIGURE 2 Multiple representations of the audio that can be computed with the module *sound*. From top to bottom, raw waveform, amplitude envelope (Hilbert transform), power spectrum and spectrogram. The 44.1 kHz sampled audio corresponds to the songs of the red-faced spinetail *Cranioleuca erythrops* overlapped by rapid calls of an unknown bird species. Recorded in scrub vegetation near a cloud forest patch in Valle del Cauca (Colombia), at 11:00 hr on 19 September 2018

development. All modules are fully interoperable so that advanced analysis on audio recordings can be performed in a flexible way.

2.2 | Module *sound*: Signal pre-processing

This module provides a simple way to load, transform and pre-process audio signals. More precisely, this module allows the reading of mono or stereo files in Waveform Audio File format (WAV) as a NumPy array. Once loaded, the signal can be adjusted and standardized with high-level functions. For instance, the signal can be downsampled or upsampled at a desired sampling rate and trimmed in the temporal domain according to specified limits. In addition, the signal can be transformed into multiple useful representations, such as the envelope, the power spectral density and the spectrogram, with a single line of code. In the temporal domain, it is possible to extract the envelope of the waveform based on the Hilbert transform or on a computationally efficient method proposed by Towsey et al. (2014). In the spectral domain, the module provides a function to compute the Fourier transform and return the power spectrum or the power spectral density. Finally, there is also a function to compute the short-time Fourier transform (STFT) to return a spectrogram (Figure 2). We paid particular attention to write a spectrogram function that fulfils Parseval's theorem, ensuring energy conservation of the signal between its time and frequency domain representations, to provide reliable sound pressure-level estimation from audio recordings.

The module also proposes more specific tools dedicated to soundscape denoising and animal vocalization enhancement. Natural soundscapes can be separated in three main categories of sounds: biophony (i.e. biotic sounds produced by animals such as birds and insects), geophony (i.e. abiotic sounds, such as wind and rain sounds) and anthropophony (i.e. sounds made by human activities, such as motorized transportation and industrial noises). During soundscapes analysis, a challenging task consists in removing geophony and anthropophony from the audio recordings in order to focus only on biophony (Towsey et al., 2014). For this, we implemented traditional frequency filters, such as low-pass, high-pass, band-pass and band-stop filters. In addition, we adapted several functions to remove stationary noises (i.e. wind or airplanes) based on spectral subtraction techniques (Boll, 1979), such as adaptive-level equalization (Towsey, 2013) and median equalizer. As a complement, the module provides the novel per-channel energy normalization (PCEN), a computationally efficient way to enhance transient sounds (e.g. bird vocalization) while discarding stationary noise (Lostanlen et al., 2019). These latter methods, which work on the spectrogram, are fast and reliable (Xie et al., 2020).

2.3 | Module *rois*: Audio segmentation

For many applications, it is important to decompose audio recordings by identifying and analysing specific ROI. The *rois* module provides multiple audio segmentation methods that identify high-energy density sections in the signal.

Temporal segmentation is the simplest segmentation method, which can be performed with an adapted peak detection algorithm based on wavelet filters (Du et al., 2006). This method is fast and suitable when the duration and frequency band of the signal of interest are known. However, natural soundscapes are usually composed with overlapping signals in time such that a spectro-temporal segmentation is needed to precisely identify signal boundaries. We adapted a region growing segmentation method from computer vision, called double threshold hysteresis binarization (Canny, 1986). The technique was used to automatically segment regions in spectrograms which encompass acoustic signatures with variable duration and spectral properties. Using the higher threshold, the technique first identifies salient STFT coefficients in the spectrogram. Then, these coefficients become seeds from which regions grow to adjacent regions as long as the coefficients are continuously connected to each other, and their value is above the lower threshold. As a result, a collection of ROIs is obtained after converting regions into time–frequency bounding boxes delimiting each segmented acoustic signature found in the spectrogram. Note that ROIs' segmentation performs best after removing stationary noises. In preliminary tests with few overlapping sounds, we have observed that similarity between the proposed automatic ROI segmentation is high when compared to human annotation (Figure 3). The observed limitations are the presence of rain (i.e. vertical spikes in the spectrogram) and continuous sounds that fill up the spectrogram without standing out clearly in the foreground.

2.4 | Module *features*: Audio characterization

The module *features* provides several methods to compute acoustic descriptors in the temporal, spectral and spectro-temporal domains, including widely used acoustic indices (Sueur et al., 2014). Temporal descriptors include zero-crossing rate, and the first four moments of the signal (mean, variance, kurtosis and skewness). Likewise, spectral descriptors include the first four moments of the power spectral density of the signal.

To characterize complex time–frequency covariations, we implemented bidimensional wavelet analysis. In particular, the module builds a custom bidimensional bank of Gabor filters, and then decomposes the spectrogram into its multiple components. Adapted from texture image recognition (Sifre & Mallat, 2013), this analysis is performed over a spectrogram and allows to capture in a few coefficients the spectro-temporal shape of sounds at multiple resolutions. Because it is performed at multiple scales, the analysis captures details and context of the sounds, giving valuable information about the signal for automated classification (Figure 4). This approach has been tested and proved effective to characterize sounds in complex and noisy environments, allowing to generate soundscape embeddings that are interpretable (Ulloa,

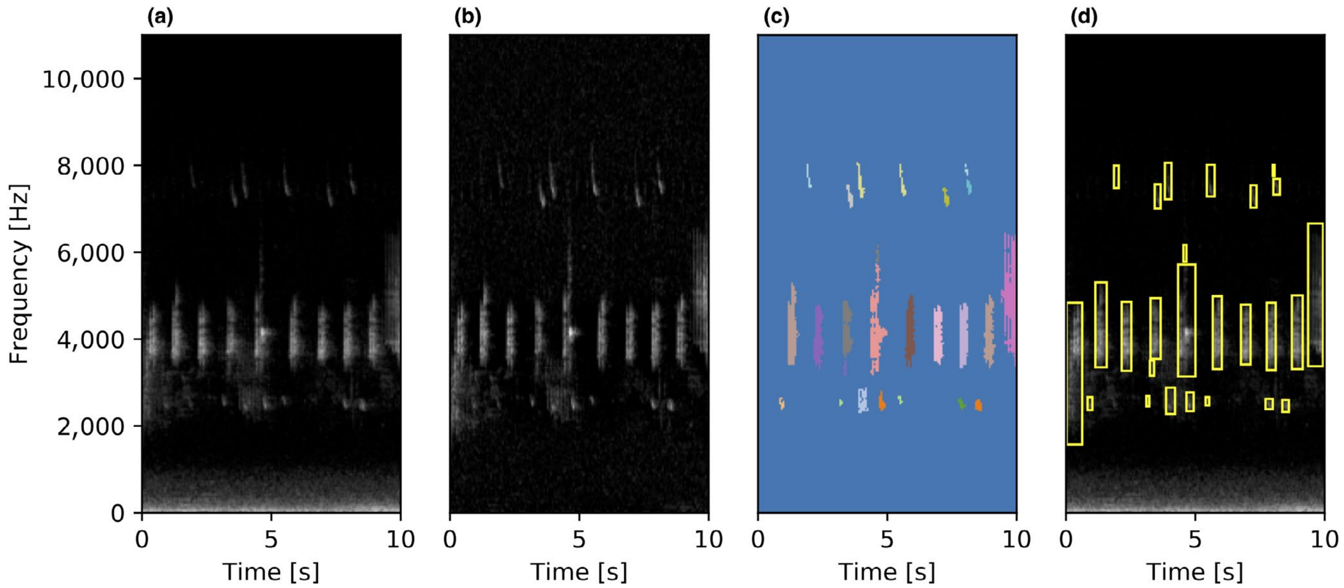

FIGURE 3 Automated time–frequency audio segmentation example using the *rois* module on a 10-s audio sample with multiple bird tweets: (a) original spectrogram, (b) spectrogram after signal filtering so that signal-to-noise ratio is increased, (c) spectrogram binarization using double threshold hysteresis binarization and (d) determination of time–frequency boundaries. Recorded in a temperate cold mountain forest covered with spruce in Jura (France), at 11:00 hr on 12 April 2019

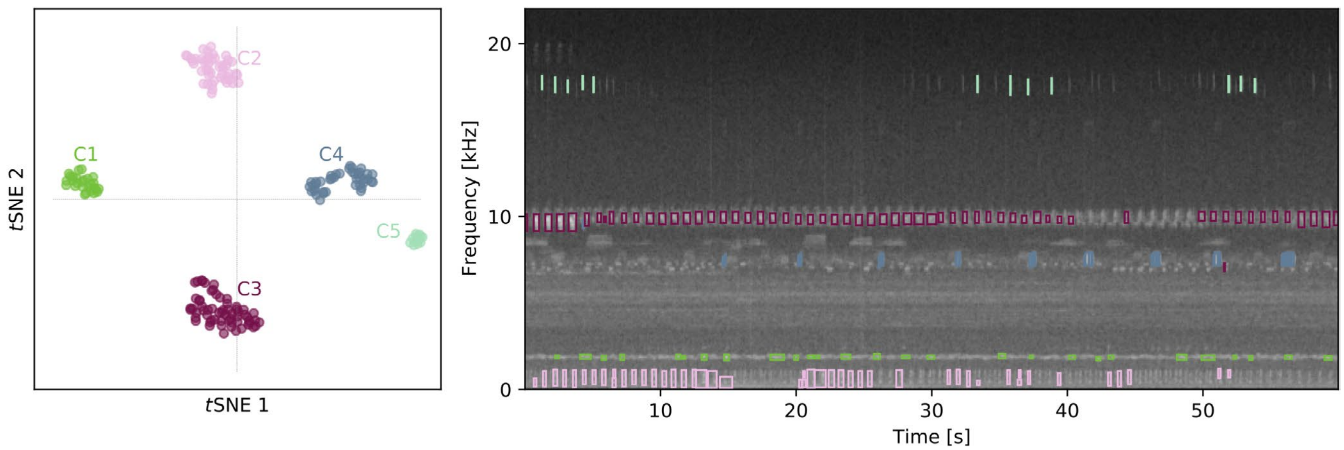

FIGURE 4 Automatic clustering of acoustic signals of tropical insects using spectro-temporal descriptors. Left, the signals were characterized using bidimensional wavelets and centroid frequency (49 descriptors in total), and then projected into a bidimensional space using the *t*-distributed stochastic neighbour embedding (*t*-SNE). On this projected space, the signals were clustered with a density-based spatial clustering of applications with noise (DBSCAN) and labelled according to their cluster. Right, the labelled signals were overlaid on the audio recording spectrogram. The audio was recorded in a rock savanna environment in the Nouragues Reserve (French Guiana) at 22:17 hr on 4 December 2014

Hernández-Palma, et al., 2021), and partition automatically the sounds of an acoustic environment based on their acoustic characteristics (Ulloa et al., 2018).

For its simplicity and interpretability, acoustic indices have become a popular way to characterize soundscapes (Sueur et al., 2014). To facilitate the use of acoustic indices, we grouped them in a single submodule that includes 28 functions which give the opportunity to compute over 50 of the most indicative and common indices (Buxton et al., 2018), such as the acoustic complexity index (ACI), the acoustic diversity index (ADI), the bioacoustic index (BI), the number of peaks (NP), the normalized difference soundscape index (NDSI) and the temporal and spectral entropy (Ht, Hf). In addition, we provide high-level functions to compute with a single line of code multiple acoustic indices based on temporal or spectral features.

2.5 | Module *spl*: Sound pressure-level estimations

The module *spl* provides functions to turn any audio recorder into a pseudo sound level meter. The module's set of functions enables the conversion of raw audio data (samples) into sound pressure levels in decibels (dB SPL) or into equivalent continuous sound pressure level (Leq), in time and frequency domains (Merchant et al., 2015). The process requires few parameters that are either provided by the manufacturer of the audio recorder or set by the user before data acquisition. These parameters are as follows: microphone sensitivity at 1 kHz in dB/V, total gain applied to amplify the input signal, maximum voltage (peak-to-peak) range supported by the analogue-to-digital (AD) converter and number of quantization levels.

Although useful, the ability to compute sound pressure levels from acoustic recorders does not replace the measurement made by a sound level meter. Indeed, the frequency response of the audio recorders is not flat, meaning that the sensitivity provided at 1 kHz is not constant along the frequency bandwidth. Despite this limitation, the estimation of sound pressure level is reliable in the frequencies of interest for most acoustic studies (i.e. 1–10 kHz), allowing comparison between datasets from research groups using different recording devices. Moreover, as the sound pressure level is a quantitative and standardized measurement, it is also possible to perform sound propagation experiments to derive a reliable habitat attenuation coefficient, a prerequisite for active distance estimation. This latter functionally will be released as a new module called *active_space*.

2.6 | Audio dataset

One of the main challenges in the automated analysis of environmental recordings is the low signal-to-noise ratio typical of audio files recorded with unattended acoustic sensors. To provide real examples of such signals and to facilitate testing of the presented tools or new approaches, we provide a diversified audio dataset of 101 audio recordings with a total time length of 20 min and 20 s. This dataset includes samples from tropical (French Guiana and Colombia) and temperate habitats (France) recorded using passive acoustic sensors (Supporting Information, Table S1). While this selection is still small, we plan to incorporate more sounds from other ecosystems in the near future.

3 | PERSPECTIVES

scikit-maad is the first open-source Python package dedicated to soundscape analysis that integrates signal denoising algorithms, automated spectro-temporal segmentation and characterization tools for signal classification, more than 50 acoustic indices and methods to get calibrated sound levels and spectrograms from passive acoustic recorders. The project is in active development, seeking additional functionalities for audio processing and measurement calibration. scikit-maad built distributions are regularly uploaded to the Python Package Index, making it easy to install and update the package on any operating system with *pip*, the main Python package management system. The source code is hosted on GitHub (https://github.com/scikit-maad/scikit-maad/), where researchers are invited to collaborate, submit new features or report issues.

This development opens new links to other related projects in Python. For example, at the end of scikit-maad's workflow, the computed ROIs and its characteristics could be classified automatically with supervised or unsupervised learning models using scikit-learn (Pedregosa et al., 2011), a comprehensive library to train, tune and test machine learning models. Alternatively, after preparing and preprocessing the audio with scikit-maad, advanced classification models could also be trained with Python Deep Learning interfaces, such as Tensorflow and PyTorch. Even the most advanced analysis heavily requires pre-processing, filtering or characterization as preliminary steps, so we anticipate that scikit-maad will be useful for a large audience.

Scientific packages for signal processing and classification are continuously being developed in Python, scikit-maad is part of this movement that will change the way we analyse environmental recordings. This development will create synergies between the community of ecoacousticians, such as engineers, data scientists, ecologists, biologists and practitioners to explore and understand the processes underlying the acoustic diversity of ecological systems.

ACKNOWLEDGEMENTS

The authors thank Chloé Huetz for the multiple ideas provided during the development of the package, and Angélica Hernández-Palma for language corrections and comments on the manuscript. J.S.U. was supported by Colciencias with a PhD scholarship (646-2014) and a Postdoctoral Grant (811-2018).

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHORS' CONTRIBUTIONS

J.S.U. and S.H. conceived the ideas and designed the package; J.S.U., S.H. and J.F.L. contributed to the development of the Python functions and the online documentation of the package; J.S.U. and S.H. led the writing of the manuscript; J.S. and T.A. initiated and supervised the research. All authors contributed critically to the drafts and gave final approval for publication.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/2041-210X.13711.

DATA AVAILABILITY STATEMENT

scikit-maad's source code and audio recordings used as illustrative examples in this article are hosted on GitHub (https://github.com/scikit-maad/scikit-maad/) and at the open-access repository Zenodo: https://zenodo.org/record/4752390 (Ulloa, Haupert, et al., 2021).

ORCID

Juan Sebastián Ulloa https://orcid.org/0000-0001-5666-6942
Sylvain Haupert https://orcid.org/0000-0003-4705-4527
Juan Felipe Latorre https://orcid.org/0000-0003-1115-2969
Thierry Aubin https://orcid.org/0000-0002-7190-8978

REFERENCES

Araya-Salas, M., & Smith-Vidaurre, G. (2017). warbleR: An r package to streamline analysis of animal acoustic signals. *Methods in Ecology and Evolution*, *8*(2), 184–191. https://doi.org/10.1111/2041-210X.12624

Bogdanov, D., Serra, X., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., & Zapata, J. (2013). Essentia: An open-source library for sound and music analysis. *Proceedings of the 21st ACM International Conference on Multimedia*, 855–858. https://doi.org/10.1145/2502081.2502229

Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *27*(2), 113–120. https://doi.org/10.1109/TASSP.1979.1163209

Buxton, R. T., McKenna, M. F., Clapp, M., Meyer, E., Stabenau, E., Angeloni, L. M., Crooks, K., & Wittemyer, G. (2018). Efficacy of extracting indices from large-scale acoustic recordings to monitor biodiversity: Acoustical Monitoring. *Conservation Biology*, *32*(5), 1174–1184. https://doi.org/10.1111/cobi.13119

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-8*(6), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Du, P., Kibbe, W. A., & Lin, S. M. (2006). Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, *22*(17), 2059–2065. https://doi.org/10.1093/bioinformatics/btl355

Giannakopoulos, T. (2015). pyAudioAnalysis: An open-source Python library for audio signal analysis. *PLoS ONE*, *10*(12), e0144610. https://doi.org/10.1371/journal.pone.0144610

Gibb, R., Browning, E., Glover-Kapfer, P., & Jones, K. E. (2018). Emerging opportunities and challenges for passive acoustics in ecological assessment and monitoring. *Methods in Ecology and Evolution*, 1–17. https://doi.org/10.1111/2041-210X.13101

Katz, J., Hafner, S. D., & Donovan, T. (2016). Tools for automated acoustic monitoring within the R package monitoR. *Bioacoustics*, *25*(2), 197–210. https://doi.org/10.1080/09524622.2016.1138415

Lostanlen, V., Salamon, J., Farnsworth, A., Kelling, S., & Bello, J. P. (2019). Robust sound event detection in bioacoustic sensor networks. *PLoS ONE*, *14*(10), e0214168. https://doi.org/10.1371/journal.pone.0214168

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and music signal analysis in Python. *Proceedings of the 14th Python in Science Conference*, *8*, 18–25.

McKinney, W. (2010). *Data structures for statistical computing in Python* (pp 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Merchant, N. D., Fristrup, K. M., Johnson, M. P., Tyack, P. L., Witt, M. J., Blondel, P., & Parks, S. E. (2015). Measuring acoustic habitats. *Methods in Ecology and Evolution*, *6*(3), 257–265. https://doi.org/10.1111/2041-210X.12330

Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science Engineering*, *9*(3), 10–20. https://doi.org/10.1109/MCSE.2007.58

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Sifre, L., & Mallat, S. (2013). Rotation, scaling and deformation invariant scattering for texture discrimination. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference On* (pp. 1233–1240). http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6619007

Sueur, J. (2018). *Sound analysis and synthesis with R*. Springer International Publishing. https://doi.org/10.1007/978-3-319-77647-7

Sueur, J., Aubin, T., & Simonis, C. (2008). Seewave: A free modular tool for sound analysis and synthesis. *Bioacoustics*, *18*, 213–226. https://doi.org/10.1080/09524622.2008.9753600

Sueur, J., Farina, A., Gasc, A., Pieretti, N., & Pavoine, S. (2014). Acoustic indices for biodiversity assessment and landscape investigation. *Acta Acustica United with Acustica*, *100*(4), 772–781. https://doi.org/10.3813/AAA.918757

Sugai, L. S. M., Silva, T. S. F., Ribeiro, J. W. Jr, & Llusia, D. (2019). Terrestrial passive acoustic monitoring: Review and perspectives. *BioScience*, *69*(1), 15–25. https://doi.org/10.1093/biosci/biy147

Towsey, M. (2013). *Noise removal from wave-forms and spectrograms derived from natural recordings of the environment|QUT ePrints*. https://eprints.qut.edu.au/61399/

Towsey, M., Truskinger, A., Cottman-Fields, M., & Roe, P. (2020). *QutEcoacoustics/audio-analysis: Ecoacoustics Audio Analysis Software (v20.11.2.0)*. Queensland University of Technology. https://doi.org/10.5281/ZENODO.4274299

Towsey, M., Wimmer, J., Williamson, I., & Roe, P. (2014). The use of acoustic indices to determine avian species richness in audio-recordings of the environment. *Ecological Informatics*, *21*, 110–119. https://doi.org/10.1016/j.ecoinf.2013.11.007

Turner, W. (2014). Sensing biodiversity. *Science*, *346*(6207), 301–302. https://doi.org/10.1126/science.1256014

Ulloa, J. S., Aubin, T., Llusia, D., Bouveyron, C., & Sueur, J. (2018). Estimating animal acoustic diversity in tropical environments using unsupervised multiresolution analysis. *Ecological Indicators*, *90*, 346–355. https://doi.org/10.1016/j.ecolind.2018.03.026

Ulloa, J. S., Haupert, S., & Latorre, J. F. (2021). Data from: scikit-maad/scikit-maad: latest stable release (v1.2). *Zenodo*. https://doi.org/10.5281/zenodo.4752390

Ulloa, J. S., Hernández-Palma, A., Acevedo-Charry, O., Gómez-Valencia, B., Cruz-Rodríguez, C., Herrera-Varón, Y., Roa, M., Rodríguez-Buriticá, S., & Ochoa-Quintero, J. M. (2021). Listening to cities during the COVID-19 lockdown: How do human activities and urbanization impact soundscapes in Colombia? *Biological Conservation*, *255*, 108996. https://doi.org/10.1016/j.biocon.2021.108996

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). scikit-image: Image processing in Python. *PeerJ*, *2*, e453. https://doi.org/10.7717/peerj.453

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Vázquez-Baeza, Y. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Xie, J., Colonna, J. G., & Zhang, J. (2020). Bioacoustic signal denoising: A review. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-020-09932-4

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

How to cite this article: Ulloa, J. S., Haupert, S., Latorre, J. F., Aubin, T., & Sueur, J. (2021). scikit-maad: An open-source and modular toolbox for quantitative soundscape analysis in Python. *Methods in Ecology and Evolution*, 12, 2334–2340. https://doi.org/10.1111/2041-210X.13711