Pyrenote: a Web-based, Manual Annotation Tool for Passive Acoustic Monitoring

Sean Perry
dept. of Mathematics
University of California, San Diego
San Diego, United States of America
shperry@ucsd.edu

Vaibhav Tiwari
dept. of Computer Science and Engineering
University of California, San Diego
San Diego, United States of America
vktiwari@ucsd.edu

Nishant Balaji
dept. of Electrical and Computer Engineering
University of California, San Diego
San Diego, United States of America
nibalaji@ucsd.edu

Erika Joun
dept. of Cognitive Science
University of California, San Diego
San Diego, United States of America
hjoun@ucsd.edu

Jacob Ayers
dept. of Electrical and Computer Engineering
University of California, San Diego
San Diego, United States of America
jgayers@ucsd.edu

Mathias Tobler
Conservation Science and Wildlife Health
San Diego Zoo Wildlife Alliance
San Diego, United States of America
matobler@gmail.com

Ian Ingram
Conservation Science and Wildlife Health
San Diego Zoo Wildlife Alliance
San Diego, United States of America
iingram@sandiegozoo.org

Ryan Kastner
dept. of Computer Science and *Engineering*
University of California, San Diego
San Diego, United States of America
kastner@ucsd.edu

Curt Schurgers
dept. of Electrical and Computer Engineering
University of California, San Diego
San Diego, United States of America
cschurgers@ucsd.edu

***Abstract*—Passive acoustic monitoring (PAM) involves deploying audio recorders across a natural environment over a long period of time to collect large quantities of audio data. To parse through this data, researchers have worked with automated annotation techniques stemming from Digital Signal Processing and Machine Learning to identify key species calls and judge a region's biodiversity. To apply and evaluate those techniques, one must acquire strongly labeled data that marks the exact temporal location of audio events in the data, as opposed to weakly labeled data which only labels the presence of an audio event across a clip.**

Pyrenote was designed to fit the demand for strong manual labels in PAM data. Based on Audino, an open-source, web-based, and easy-to-deploy audio annotation tool, Pyrenote displays a spectrogram for audio annotation, stores labels in a database, and optimizes the labeling process through simplifying the user interface to produce high-quality annotations in a short time frame. This paper documents Pyrenote's functionality, how the challenge informed the design of the system, and how it compares to other labeling systems.

***Index Terms*—passive acoustic monitoring, audio annotation tool, annotation, strong labels, open-source**

I. Introduction

Passive acoustic monitoring (PAM), the systematic collection of audio recordings in the field, has been on the rise with the growth of machine learning audio techniques and a wider range of acoustic databases, such as the popular Xeno-canto audio dataset for species calls [6, 10, 9]. With the ability to judge the biodiversity of the nearby environment by identifying and counting the number of keynote species in audio data, passive acoustic monitoring poses a great opportunity to track the effects of deforestation, habitat loss, and climate change [14, 9, 19]. Additionally, PAM system can produce several hundred hours of audio recording making it infeasible for humans to manually annotate the data in a timely fashion [3]. Thus it is important to develop machine learning to measure and segment the large amounts of acoustic data created from passive acoustics monitoring systems.

Despite the wide variety of data available, acoustic machine learning techniques have been limited by a lack of strongly labeled data, labels that note the exact start and end times in an audio clip that an audio event occurred at. Instead, many datasets, like the Xeno-canto dataset, contain weakly labeled data where labels only indicate the presence of a given audio event in the clip. While research is being done to use weakly labeled data to train automated audio segmentation, strongly labeled data is still required by those models to evaluate

This paper was funded by the REU Engineers for Exploration which is funded by the National Science Foundation Grant Number 1852403. Additional support for some students on the project was provided by the Summer Research Internship Program and Undergraduate Research Scholarships Program at the University of California, San Diego

their accuracy in identifying audio events [18]. Thus, to apply these machine learning techniques to the data collected by PAM systems, it is necessary to obtain more strongly labeled annotations from species calls in that region.

Therefore, manual work is needed to create more strongly labeled data to further develop these bioacoustic machine learning models. Such a manual labeling system would ideally be web-based for easy access and team-wide distribution compared to downloadable applications as well as having an audio visualization ideal for playing and labeling sound. It must have a secure backend capable of storing audio clips and labels for an entire team to access and be able to deploy on a server at a low cost. Finally, the system has to have an easy-to-use interface for volunteers to produce quality annotations.

To do this, we have created Pyrenote, a wildlife audio annotation tool based on the open-source human audio annotation tool, Audino [21]. Utilizing Audino's use of docker for an easier deployment onto a server, Pyrenote reworks Audino with a focus on simplifying the annotation process for wildlife audio. By using the best of Audino and simplifying the frontend, Pyrenote is an ideal platform for meeting bioacoustic data needs.

II. Related Works

Based on the specifications outlined for an ideal audio annotation tool for PAM data, many other audio annotation tools are not ideal for bioacoustic data. First off, some platforms cannot produce strongly labeled data. Zooniverse, a crowdsourcing site notable for the discoveries of galaxies [5], only allows their biggest audio projects to annotate for the presence of an audio event in a clip [17, 30]. Another powerful annotation tool is Prodigy, which is an active learning platform where a machine learning model recommends only difficult clips for users to label which reduces the number of clips needed to annotate [16]. Despite the benefits, this system also has a weakly labeled data approach [27]. Smaller, open-source projects, like Dynitag, also only use weakly labeled data [22]. Since we need a system capable of labeling data strongly, these approaches primarily miss the mark. Some annotation systems are also not found online, such as Audacity. Audacity is a downloadable annotation and audio analysis software [1].

Then there is the frontend design of the application, as many systems often have complex labeling processes. BAT, an audio annotator based on a JavaScript library for visualizing and playing audio on websites called Wavesurfer.js [28], features a 2 phased labeling system. For BAT, users first create annotations and then, in a second phase, relabel sections of audio events that overlap to indicate which audio event was louder. On top of this, the platform did not let users easily move labels across the spectrogram and the users can only resize labels [13]. Another open-source platform called Koe has a large number of features for users to annotate and analyze audio data [20]. Adding these complexities makes the site harder to use and navigate for annotators, making it less ideal for bioacoustics.

Other systems present technical challenges for researchers to deploy the sites on their servers. Audio-annotator, another Wavesurfer.js based annotation tool, for instance, lacks a backend to store audio labels and clips. This means that users can only produce labels on each annotator's machine. To have a system for each user to access other user's work, a backend is needed to store this data. Systems like EchoML and Microfaune Annotator which have backends require researchers to have access to cloud storage systems [26, 24] which require additional funds and the knowledge needed to set up these cloud storage accounts. Having a system that is easy for researchers to deploy with fewer additional steps as possible means researchers can get to annotating their audio clips faster.

III. Pyrenote

To address the need for a more ideal audio annotation tool for passive acoustic monitoring, we created Pyrenote. The annotation tool utilizes Audino and Wavesurfer.js to meet all the ideal qualities that an audio annotation tool should have for these researchers. With Audino, Pyrenote takes advantage of a pre-existing backend, simple to deploy docker containerization, web-based and free framework from which to build the system. However, since Audino's primary focus is human speech annotation [21], Pyrenote has improved on Audino by optimizing the frontend annotation page for labeling bioacoustic data rather than human speech.

Pyrenote's workflow is illustrated in figure 1. After admins manually upload data and create projects and labels from the admin portal on the website, users can access these audio files to annotate. Users create annotations by clicking and dragging regions that users can individually label. These annotations and their associated labels are uploaded to the backend where admins can retrieve these annotations in CSV or JSON formats. The process of creating annotations and curating audio data is in the hands of the users and the system currently automates the storage and giving the users audio files to annotate.

The first change made between Audino and Pyrenote was changing Audino's waveform to a spectrogram. Waveforms may work well with visualizing the stops and ends of human speech, but experiments have shown waveforms are about as effective as no visualization at all for improving the time it takes users to produce high-quality clips. Spectrograms were shown to improve the user's ability to produce higher quality clips at much faster rates than waveforms [8]. Additionally, users can also identify patterns in the spectrogram correlated with specific species calls. As seen in figure 2, species calls can produce repeatable patterns in spectrograms. Users can therefore use those frequency-time patterns that represent species calls to more easily identify species calls. Thus to make it easier for users to identify species, we include the spectrogram plugin, which uses Fast Fourier Transform to produce spectrograms [29] to Wavesurfer and render a spectrogram instead of a waveform. Now in addition to hearing patterns in audio data and the user's personal expertise at

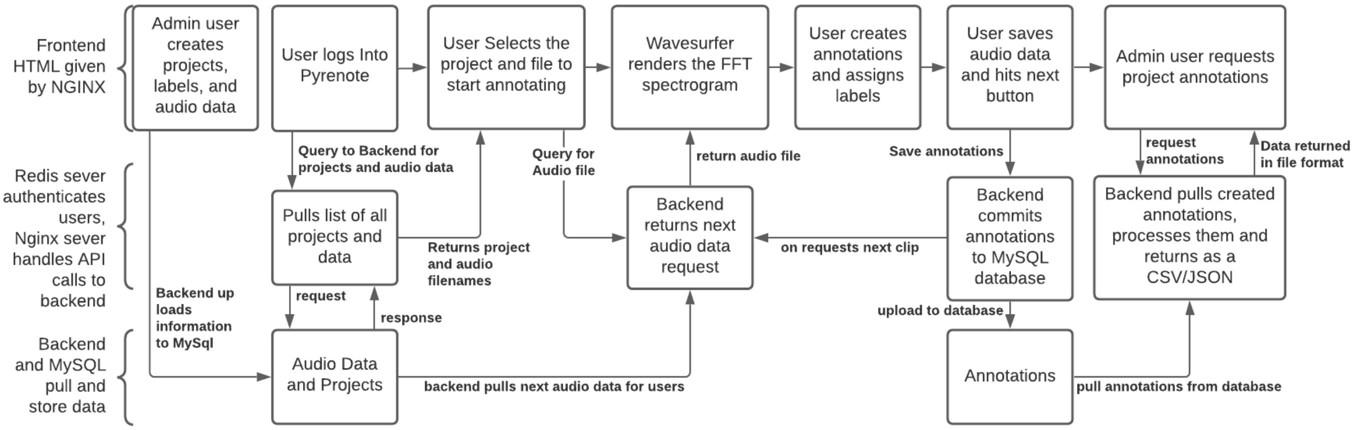
Frontend HTML given by NGINX
Redis sever authenticates users, Nginx sever handles API calls to backend
Backend and MySQL pull and store data
Admin user creates projects, labels, and audio data
User logs Into Pyrenote
User Selects the project and file to start annotating
Wavesurfer renders the FFT spectrogram
User creates annotations and assigns labels
User saves audio data and hits next button
Admin user requests project annotations
Query to Backend for projects and audio data
Query for Audio file
return audio file
Save annotations
request annotations
Data returned in file format
Pulls list of all projects and data
Returns project and audio filenames
Backend returns next audio data request
on requests next clip
Backend commits annotations to MySQL database
Backend pulls created annotations, processes them and returns as a CSV/JSON
Backend up loads information to MySql
request
response
upload to database
Audio Data and Projects
backend pulls next audio data for users
Annotations
pull annotations from database

Fig. 1. Pyrenote's workflow for obtaining strongly labeled annotations

TABLE I
ANNOTATION TOOLS COMPARISON

	Features						
Tools Annotation	***strongly labeled?***	***Label On Spectrogram?***	***Simple User Interface***	***Backend Ready?***	***Cheap To Run?***	***Web Based?***	***Easy to Deploy?***
Pyrenote	yes	yes	yes	yes	yes	yes	yes
Audino	yes	no	no	yes	yes	yes	yes
Audio-Annotator	yes	yes	yes	no	yes	yes	yes
Mircofaune	yes	yes	yes	yes	no	yes	yes
EchoML	yes	no	no	yes	no	yes	yes
BAT	yes	yes	no	yes	yes	yes	yes
Koe	yes	yes	no	yes	yes	yes	yes
Zooniverse	no	no	yes	yes	yes	yes	yes
Dylib	no	no	yes	yes	yes	yes	yes
Proidgy	no	no	yes	yes	yes	yes	yes
Audacity	yes	no	no	yes	no	no	n/a

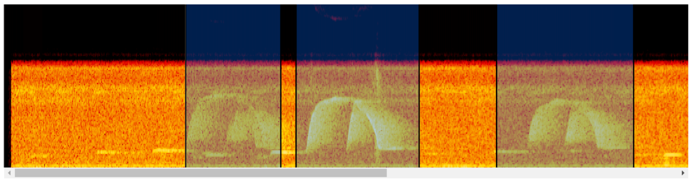

Fig. 2. A segment of audio from a Screaming Piha (Lipaugus vociferans) portrayed on a spectrogram. The highlighted regions are the actual calls. The spectrogram shows a clear pattern between the calls; Patterns that make it easier for users to quickly identify and annotate this species call

identification, users will also be all to see patterns that can help them identify audio events.

The next step was to make the process of assigning labels to annotations more suitable and streamlined for bioacoustic data. Currently, Audino has users transcribe what is spoken in the label in a transcription section located between the save button and the waveform as seen in figure 3. Transcribing human speech is not required for labeling non-human speech sounds like species calls. Thus we removed this section from the site to keep the focus closer to working with wildlife acoustic data.

Furthermore, some quality-of-life improvements were added to make the interface simpler. The first change is the addition of the next button. Whereas before a user had to navigate through to the dashboard to access the next clip to annotate, the user can now simply hit the next button to navigate to the next clip, reducing loading times when navigating between clips. The original save button from Audino has also been replaced with a save all button which goes through every region and uploads their data to the backend, thus making it more similar to text processing software that saves entire files rather than each word in a file. With a save all button, if the user did not make a mistake in annotating, they only need to press the save button once. Finally, users can see which clips are saved because when a save is made, the region in Pyrenote changes to a darker color. This way, users don't have to manually check each region to make sure it received an annotation and therefore was saved.

IV. DISCUSSION

With these changes, Pyrenote is an ideal platform for researchers looking for wanting to deploy a simple-to-use, web-based application to manually create strongly labeled

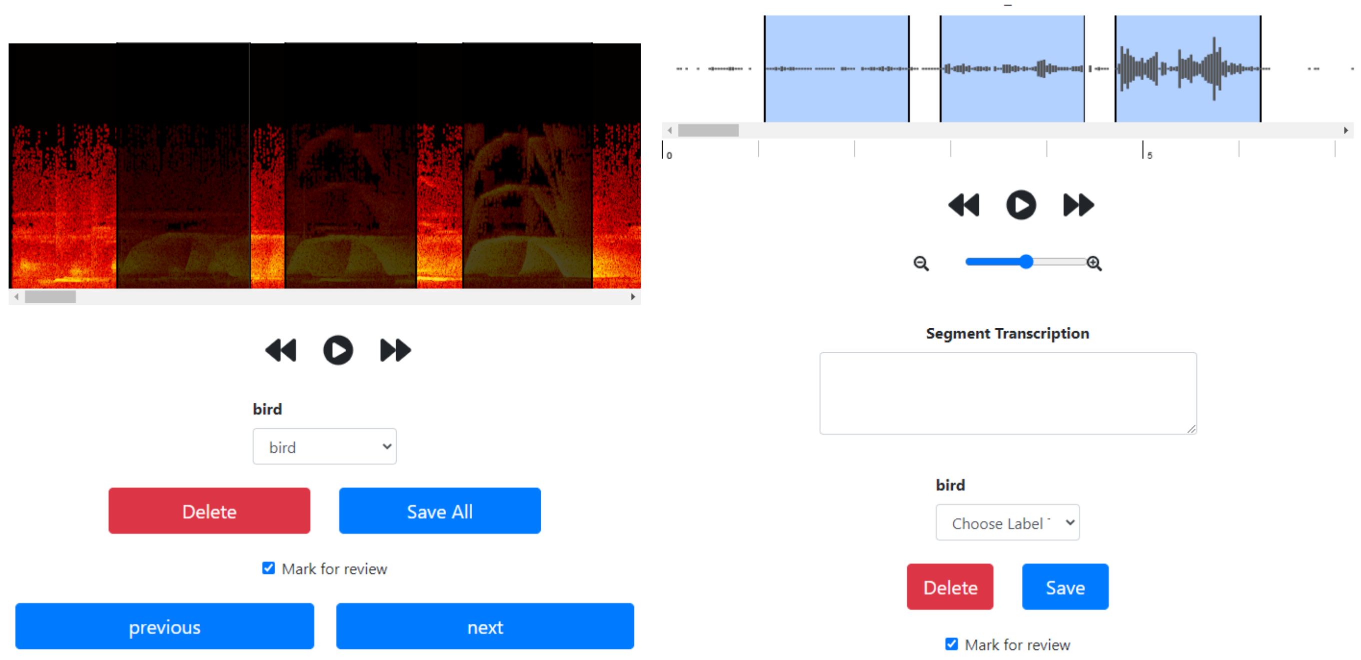

Fig. 3. Comparision betweeen the designs of Audino (right) and Pyrenote (left). Note the lack of waveform and transcription in Pyrenote. Previous and next buttons have also been added to Pyrenote.

TABLE II
EXAMPLE LABELING OUTPUT FROM PYRENOTE

IN FILE	CLIP LENGTH	OFFSET	DURATION	SAMPLING RATE	LABEL	TIME SPENT
20190612_080000.WAV	60	0.415	0.19	384000	bird	181.004
20190612_080000.WAV	60	2.265	0.2	384000	bird	181.007

data from acoustic monitoring systems for evaluation and/or training of automated segmentation of audio data. Simplifications made to the UI of Pyrenote streamline the annotation process and the addition of spectrograms allow for easy viewing of high-frequency audio calls. A further comparison between Pyrenote and other systems via the ideals listed in the introduction can be seen in Table 1. The addition of strongly labeled data, along with its role in evaluating machine learning systems, can improve the performance of machine learning models compared to models trained purely from weakly labeled datasets alone [12, 23]. A system like ours that can make manual annotations easier to produce can therefore be applied to these systems to improve accuracy in the segmentation and detection of audio events. Thus the system can help push the machine learning applications for passive acoustic monitoring by being able to make it easier for researchers to produce more strongly labeled annotations.

Currently, the system can produce labels in JSON and CSV output format. The CSV format was added to make it easier for an additional file input for programming languages such as Python and R. An example of the CSV output is shown above in Table 2. With this implementation in place, the system is ready for small teams to annotate bioacoustic data.

V. Future Plans

Plans for Pyrenote fall into two categories: improving user experience and increasing automation on the site. On the side of user experience, it involves things such as reducing the number of pages that a user needs to load while operating through this workflow, like using a single page to create annotations rather than loading a new page every time the user goes to a new audio clip when they hit the next button. This reduces the number of steps needed to use the site and helps speed up the annotation process. Additionally, for projects that require a greater number of users and clips, it will become important to give control over project leads rather than system admins for managing their projects. In this way, the management of a larger user base can be more decentralized and easier to manage. Giving further control to project leads means that users can directly upload data and download the label results from a private project without needing to work through an admin user that controls everything, thus cutting out the middle man as the system grows.

On the other side of our plans involve increasing automation on the site. The site requires high levels of human intervention to filter audio data and double-check data for quality control. Automation, for instance, could make it easier for admins to curate data for annotation. A preprocessing pipeline that runs before recommending clips to users can select a subset from an uploaded dataset that could have species calls in them to annotate. Automation similar to the system implemented in BAT [13] could also be used to ensure quality control via the magnitude of overlap between annotations from different users annoying the same audio clip. Clips with lower overlaps could then be rerecommended to users for clarification. Currently, alongside Pyrenote, our team is developing a python package called PyHa that can generate automated labels via machine learning techniques such as hybrid RNNs and digital signal techniques like 1D cross-correlation [25, 7, 15]. The package can help Pyrenote identify audio clips of interest as well as use statistical analysis to judge labels being produced by users in a quantifiable manner thus helping to automate the critical decision of what audio data should users annotate and ensure that users are producing high-quality annotations.

An additional step in automating Pyrenote would be to integrate active learning. Active learning systems use machine learning models as queries for human annotators. When a user "queries"' the system for audio data, the system returns audio clips the system had trouble working with. The human "clarify" these clips by labeling them to which the system learns based on the clip [4, 2, 11]. In this way, the system maximizes the user's time by having the user only label challenging clips. Interfacing it with Pyrenote can reduce the number of strongly labeled annotations needed while also creating a system capable of being deployed to automatically segment bioacoustic data.

VI. Conclusion

With Pyrenote ready for use, researchers have a tool at their disposal to manually create strongly labeled annotations for bioacoustic data on the web. Strongly labeled data can greatly improve the outcomes of machine learning techniques or at least serve as a test set against models trained on lower resource data. As we look to improve on Pyrenote, we aim to make labeling the mountain of data from passive acoustics monitoring systems easier and faster and thereby make the application of machine learning on these datasets easier as well. In doing so, researchers can create automated systems capable of monitoring our planet, biodiversity, and climate change.

Additional resources

The source code for Pyrenote can be seen at
https://github.com/UCSD-E4E/Pyrenote.

Acknowledgement

We would like to thank our collaborators at the San Diego Wildlife Alliance for providing audio data from the Peruvian Amazon to our project, as well as providing server space to host Pyrenote. We would also like to thank the support from the Computer Science and Engineering department and the Electrical and Computer Engineering department at the University of San Diego and everyone working on the Acoustics Species Identification project at Engineers For Exploration. Special thanks to Chris Crutchfield for his help in getting Pyrenote set up on the server.

References

[1] Beinan Li, John Burgoyne, and Ichiro Fujinaga. "Extending Audacity for Audio Annotation." In: 2006, pp. 379–380.

[2] Anita Krishnakumar. *Active Learning Literature Survey*. 2007.

[3] Andrew Digby et al. "A practical comparison of manual and autonomous methods for acoustic monitoring". In: *Methods in Ecology and Evolution* 4.7 (2013), pp. 675–683. ISSN: 2041-210X. DOI: 10.1111/2041-210X.12060. URL: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.12060 (visited on 07/29/2021).

[4] Charu C. Aggarwal, ed. *Data Classification*. 0th ed. Chapman and Hall/CRC, July 25, 2014. ISBN: 978-1-4665-8675-8. DOI: 10.1201/b17320. URL: https://www.taylorfrancis.com/books/9781466586758 (visited on 07/05/2021).

[5] Robert Simpson, Kevin R. Page, and David De Roure. "Zooniverse: Observing the World's Largest Citizen Science Platform". In: *Proceedings of the 23rd International Conference on World Wide Web*. WWW '14 Companion. event-place: Seoul, Korea. New York, NY, USA: Association for Computing Machinery, 2014, pp. 1049–1054. ISBN: 978-1-4503-2745-9. DOI: 10.1145/2567948.2579215. URL: https://doi.org/10.1145/2567948.2579215.

[6] Willem-Pier Vellinga and Robert Planqué. "The Xeno-canto collection and its relation to sound recognition and classification". In: 2015.

[7] Juan Sebastian Ulloa et al. "Screening large audio datasets to determine the time and space distribution of Screaming Piha birds in a tropical forest". In: *Ecological Informatics* 31 (Jan. 1, 2016), pp. 91–99. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2015.11.012. URL: https://www.sciencedirect.com/science/article/pii/S1574954115001934 (visited on 07/27/2021).

[8] Mark Cartwright et al. "Seeing Sound: Investigating the Effects of Visualizations and Complexity on Crowdsourced Audio Annotations". In: *Proc. ACM Hum.-Comput. Interact.* 1 (CSCW Dec. 2017). Place: New York, NY, USA Publisher: Association for Computing Machinery. DOI: 10.1145/3134664. URL: https://doi.org/10.1145/3134664.

[9] Jessica L. Deichmann et al. "Soundscape analysis and acoustic monitoring document impacts of natural gas exploration on biodiversity in a tropical forest". In: *Ecological Indicators* 74 (2017), pp. 39–48. ISSN: 1470-160X. DOI: https://doi.org/10.1016/j.ecolind.2016.11.002. URL: https://www.sciencedirect.com/science/article/pii/S1470160X16306392.

[10] Eduardo Fonseca et al. "Freesound Datasets: A Platform for the Creation of Open Audio Datasets". In: *ISMIR*. 2017.

[11] Simone Hantke, Zixing Zhang, and Björn Schuller. "Towards Intelligent Crowdsourcing for Audio Data Annotation: Integrating Active Learning in the Real World". In: *Proc. Interspeech 2017*. 2017, pp. 3951–3955. DOI: 10.21437/Interspeech.2017-406. URL: http://dx.doi.org/10.21437/Interspeech.2017-406.

[12] Anurag Kumar and Bhiksha Raj. "Audio event and scene recognition: A unified approach using strongly and weakly labeled data". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 3475–3482. DOI: 10.1109/IJCNN.2017.7966293.

[13] Blai Meléndez-Catalán, Emilio Molina, and E. Gómez. "BAT: An open-source, web-based audio events annotation tool". In: 2017.

[14] José Wagner Ribeiro, Larissa Sayuri Moreira Sugai, and Marconi Campos-Cerqueira. "Passive acoustic monitoring as a complementary strategy to assess biodiversity in the Brazilian Amazonia". In: *Biodiversity and Conservation* 26.12 (Nov. 2017), pp. 2999–3002. ISSN: 0960-3115, 1572-9710. DOI: 10.1007/s10531-017-1390-0. URL: http://link.springer.com/10.1007/s10531-017-1390-0 (visited on 07/05/2021).

[15] Veronica Morfi and Dan Stowell. "Deep Learning for Audio Event Detection and Tagging on Low-Resource Datasets". In: *Applied Sciences* 8.8 (Aug. 2018). Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, p. 1397. DOI: 10.3390/app8081397. URL: https://www.mdpi.com/2076-3417/8/8/1397 (visited on 07/27/2021).

[16] Minh-Quoc Nghiem and Sophia Ananiadou. "APLenty: annotation tool for creating high-quality datasets using active and proactive learning". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 108–113. DOI: 10.18653/v1/D18-2019. URL: https://aclanthology.org/D18-2019.

[17] Mark Cartwright et al. "Crowdsourcing Multi-Label Audio Annotation Tasks with Citizen Scientists". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–11. ISBN: 978-1-4503-5970-2. URL: https://doi.org/10.1145/3290605.3300522.

[18] Veronica Morfi et al. "NIPS4Bplus: a richly annotated birdsong audio dataset". In: *PeerJ Computer Science* 5 (Oct. 7, 2019), e223. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.223. URL: https://peerj.com/articles/cs-223 (visited on 07/05/2021).

[19] Larissa Sayuri Moreira Sugai and Diego Llusia. "Bioacoustic time capsules: Using acoustic monitoring to document biodiversity". In: *Ecological Indicators* 99 (2019), pp. 149–152. ISSN: 1470-160X. DOI: https://doi.org/10.1016/j.ecolind.2018.12.021. URL: https://www.sciencedirect.com/science/article/pii/S1470160X18309543.

[20] Yukio Fukuzawa et al. "Koe : Web-based software to classify acoustic units and analyse sequence structure in animal vocalizations". In: *Methods in Ecology and Evolution* 11 (2020). DOI: 10.1111/2041-210X.13336.

[21] M. S. Grover et al. "audino: A Modern Annotation Tool for Audio and Speech". In: *ArXiv* abs/2006.05236 (2020).

[22] *dynilib/dynitag*. original-date: 2018-06-27T12:08:17Z. Feb. 1, 2021. URL: https://github.com/dynilib/dynitag (visited on 07/06/2021).

[23] Shawn Hershey et al. "The Benefit of Temporally-Strong Labels in Audio Event Classification". In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 366–370. DOI: 10.1109/ICASSP39728.2021.9414579.

[24] *microfaune/MicrofauneAnnotator*. original-date: 2019-11-27T22:22:15Z. Feb. 16, 2021. URL: https://github.com/microfaune/MicrofauneAnnotator (visited on 07/06/2021).

[25] *UCSD-E4E/PyHa*. original-date: 2021-01-31T00:01:15Z. July 24, 2021. URL: https://github.com/UCSD-E4E/PyHa (visited on 07/27/2021).

[26] Rita Zhang. *ritazh/EchoML*. original-date: 2017-06-22T00:18:07Z. June 21, 2021. URL: https://github.com/ritazh/EchoML (visited on 07/06/2021).

[27] *Audio & Video · Prodigy · An annotation tool for AI, Machine Learning & NLP*. Prodigy. URL: https://prodi.gy/features/audio-video (visited on 07/06/2021).

[28] *wavesurfer.js*. URL: https://wavesurfer-js.org/ (visited on 07/06/2021).

[29] *wavesurfer.js*. URL: https://wavesurfer-js.org/plugins/spectrogram.html (visited on 07/06/2021).

[30] *Zooniverse*. URL: https://www.zooniverse.org/projects/cetalingua/manatee-chat (visited on 07/06/2021).