MEDLEYDB 2.0 : NEW DATA AND A SYSTEM FOR SUSTAINABLE DATA COLLECTION

Rachel M. Bittner[(1)], **Julia Wilkins**[(2)], **Hanna Yip**[(3)], **Juan P. Bello**[(1)]
[(1)]Music and Audio Research Lab, New York University,
[(2)] Northwestern University, [(3)] The Spence School

ABSTRACT

We present *MedleyDB 2.0*, the second iteration of a dataset of multitrack recordings created to support Music Information Retrieval (MIR) research. *MedleyDB 2.0* introduces several new tools to reduce the effort required to add new content, ensure dataset sustainability, and improve the quality of multitrack audio files. The dataset has now grown to contain over 250 multitracks after the addition of 132 tracks in this release.

1. INTRODUCTION

MedleyDB [3] was released in 2014 as a dataset of royalty-free multitrack recordings developed to support MIR research. It provides a stereo mix and both dry and processed multitrack stems for each song in the dataset. The dataset covers a wide distribution of genres and primarily consists of full length songs with professional or near-professional audio quality. The dataset has been used for a variety of tasks including source separation [8], melody extraction [2], data augmentation [6], and as source material for music cognition studies [7].

While the dataset has proven to be a valuable resource, the data collection and annotation process proved to be difficult to sustain, and several types of errors in the audio were discovered. *MedleyDB 2.0* serves to address these problems by creating a website to make data collection management frictionless, and an application to help automate the process and prevent errors. These tools were used to create a second batch of data, expanding the total collection to 254 multitracks.

2. DATA COLLECTION & MANAGEMENT

We identified seven stages in the "life cycle" of a multitrack: (1) an artist expresses their interest in contributing to the database, (2) a recording session is scheduled, (3) the session is recorded, (4) an engineer mixes the session, (5) the session is exported to `wav` files, (6) the tracks are processed through the multitrack application, and finally, (7) the tracks are added to the *MedleyDB* database. Currently, in the process of transforming a recording from the studio into a completed multitrack, individual multitracks can easily be lost through long chains of emails and lack of follow through. For example, an artist may have expressed interest in contributing, but the session never gets recorded; a session may have been recorded, but the audio never gets mixed by an engineer, etc.. As a result, valuable data is easily lost because of the lack of a centralized system.

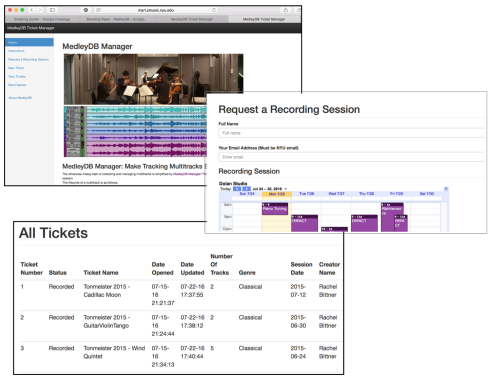

Figure 1. MedleyDB Manager adds sustainability and usability to the dataset.

The otherwise messy task of collecting and managing multitracks is simplified by MedleyDB Manager [1]. This collaborative website creates a "ticket" for each recording session. Each ticket includes details about the recording session, the individual multitracks recorded in the session, and the people involved at each stage. When a user requests a recording session on the website, automated emails are sent to the appropriate people containing the signed consent form and the specifics of the request. If a session has already been recorded, users can create a new ticket and fill out the information about the ticket status, session, multitracks, and people involved. Existing tickets can be viewed, edited and kept track of on a consolidated table that is connected to a database. Appropriate people can be contacted if questions or problems arise. MedleyDB Manager also includes rendering instructions for Pro Tools and Mac OSX for mixing engineers.

[1] `http://marl.smusic.nyu.edu:5080`

© Rachel M. Bittner[(1)], Julia Wilkins[(2)], Hanna Yip[(3)], Juan P. Bello[(1)] . Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Rachel M. Bittner[(1)], Julia Wilkins[(2)], Hanna Yip[(3)], Juan P. Bello[(1)] . "*MedleyDB 2.0* : New Data and a System for Sustainable Data Collection", Extended abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference, 2016.

Figure 2. MedleyDeBugger checks the audio files for errors.

3. AUTOMATION & ERROR CATCHING

In the initial release of *MedleyDB*, we found that submitted multitracks often contained errors ranging from the incorrect alignment of stem files to their associated mix, to a common Pro Tools error in which a track is exported as a silent file. Additionally, manually renaming files and creating a metadata file presents obvious room for user error. To address these potential errors and improve consistency and accuracy within the dataset, we created MedleyDeBugger, an application that automatically checks the users audio files for errors and establishes a standardized naming system.

MedleyDeBugger, available for download on Github [2], performs a series of error-proofing checks on the audio files using `pysox` [1] and `librosa` [5] before they are labeled and submitted to the dataset. Files are initially checked for silence, file format (i.e. number of channels, bit depth, sample rate), and length consistency. We also include a feature that allows one-click deletion of unintentional silent files if they are detected during this check. Before users are prompted for metadata information about the final mix, a final processing check occurs to ensure that raw files are correctly aligned and included with their associated stem and that the same is true for the stem files relative to the final mix. To perform the alignment check, we first downsample the audio (to reduce runtime) and then perform an unbiased cross-correlation between the raw files and their associated stems (and stem files with associated mix). To make certain that each raw and stem component is actually included in the mix, our inclusion check calculates weighted mixing coefficients and flags weights that are below a threshold. Lastly, we have included a single-instrument classifier that will alert users if they have labeled a raw or stem file instrument incorrectly (i.e. when the probability for the labeled instrument falls below a threshold). Users also have the opportunity to listen to each audio component at each stage of the application.

After the files have been checked for errors, a folder containing the raw, stem, and mix files is created. This application reorganizes the files into a standardized structure and naming system, and additionally creates a metadata file. In the future we plan to also run all automatic annotations directly within the application.

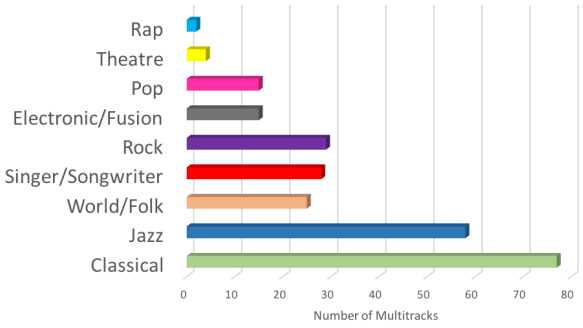

Figure 3. Genre distribution of 254 multitracks from *MedleyDB 1.0* and *2.0* combined.

4. ANNOTATIONS

The annotations and metadata from the first release of MedleyDB have been moved to a public Github repository [3], so that mistakes can be centrally reported and corrected as needed. This also allows for others to easily contribute new types of annotations. Similarly, the annotations and metadata for this release will be available on Github.

The initial release of *MedleyDB* contained human-generated melody annotations using the Tony tool [4]. However, the process was difficult to sustain in the long term, thus for this iteration of the dataset we rely primarily on automatic annotations. The automatic annotations include instrument activations and synthetic melody, multi-f0 and bass annotations.

5. NEW DATA

In addition to creating the website database management system and error-checking application, we have also added 132 new multitracks to the dataset. These multitracks contain a variety of genres but are predominately classical and jazz pieces, as displayed in Figure 3. This addition to the dataset brings the total number of multitracks in *MedleyDB 2.0* to 254.

6. CONCLUSION

MedleyDB 2.0 is an expanded dataset with a more sustainable, organized, and collaborative system for managing data and error catching. MedleyDB Manager eases the multitrack management process, while MedleyDeBugger ensures that the final multitrack audio files are error-free and clearly named.

[2] `https://github.com/marl/medley-de-bugger`

[3] `http://www.github.com/marl/medleydb`

7. REFERENCES

[1] Rachel M. Bittner, Eric Humphrey, and Juan P. Bello. pysox: Leveraging the audio signal processing power of sox in python. In *Proceedings of the 17th International Society for Music Information Retrieval Conference Late Breaking and Demo Papers*, 2016.

[2] Rachel M Bittner, Justin Salamon, Slim Essid, and Juan P Bello. Melody extraction by contour classification. In *Proc. ISMIR '16 (New York, NY)*, pages 500–506.

[3] Rachel M Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello. MedleyDB: A multitrack dataset for annotation-intensive mir research. In *ISMIR '14 (Taipei, Taiwan)*, pages 155–160, 2014.

[4] Matthias Mauch, Chris Cannam, Rachel Bittner, George Fazekas, Justin Salamon, Jiajie Dai, Juan Bello, and Simon Dixon. Computer-aided melody note transcription using the tony software: Accuracy and efficiency. 2015.

[5] B. McFee, C. Raffel, D. Liang, D.P.W. Ellis, M. McVicar, E. Battenberg, and O. Nieto. librosa: Audio and music signal analysis in python. In *14th annual Scientific Computing with Python conference*, SciPy, July 2015.

[6] Brian McFee, Eric J Humphrey, and Juan P Bello. A software framework for musical data augmentation. In *ISMIR '15 (Malaga, Spain)*, 2015.

[7] Jenefer Robinson and Robert S Hatten. Emotions in music. *Music Theory Spectrum*, 34(2):71–106, 2012.

[8] Andrew JR Simpson, Gerard Roma, and Mark D Plumbley. Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network. In *LVA/ICA '15 (Liberec, Czech Republic)*, pages 429–436. Springer, 2015.