THE FREESOUND LOOP DATASET AND ANNOTATION TOOL

António Ramires[1]   **Frederic Font**[1]   **Dmitry Bogdanov**[1]   **Jordan B. L. Smith**[2]
Yi-Hsuan Yang[3]   **Joann Ching**[3]   **Bo-Yu Chen**[3]   **Yueh-Kao Wu**[3]
Hsu Wei-Han[3]   **Xavier Serra**[1]

[1] Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[2] TikTok, Lodon, United Kingdom
[3] Research Center for IT Innovation, Academia Sinica, Taipei, Taiwan

antonio.ramires@upf.edu

ABSTRACT

Music loops are essential ingredients in electronic music production, and there is a high demand for pre-recorded loops in a variety of styles. Several commercial and community databases have been created to meet this demand, but most are not suitable for research due to their strict licensing. We present the Freesound Loop Dataset (FSLD), a new large-scale dataset of music loops annotated by experts. The loops originate from Freesound, a community database of audio recordings released under Creative Commons licenses, so the audio in our dataset may be redistributed. The annotations include instrument, tempo, meter, key and genre tags. We describe the methodology used to assemble and annotate the data, and report on the distribution of tags in the data and inter-annotator agreement. We also present to the community an online loop annotator tool that we developed. To illustrate the usefulness of FSLD, we present short case studies on using it to estimate tempo and key, generate music tracks, and evaluate a loop separation algorithm. We anticipate that the community will find yet more uses for the data, in applications from automatic loop characterisation to algorithmic composition.

1. INTRODUCTION

Repurposing audio material to create new music—also known as *sampling*—was a foundation of electronic music and is a fundamental component of this practice. Loops are audio excerpts, usually of short duration, that can be played repeatedly in a seamless manner [28]. These loops can serve as the basis for songs, which music makers can combine, cut and rearrange, and have been extensively used in Electronic Dance Music (EDM) tracks [4].

Audio loops have been made available for amateur and professional music makers since the early ages of electronic music. Currently, large-scale databases of audio offer huge collections of audio material for users to work with. Some databases, like Freesound [1] and Looperman [2], are community-oriented: people upload their sounds so that other users can employ them in their works. More commonly, these collections are commercially oriented: loops are available to paying costumers, either through a subscription service (*e.g.* Sounds.com,[3] Splice[4]) or by allowing customers to buy packs of loops (*e.g.* Loopmasters,[5] and Prime Loops[6]).

Despite the number of loops available on these databases, the technologies used to analyse and navigate these databases still rely on human annotations and human content curation to, for instance, group sounds into packs for specific genres or styles. Loops are being manually annotated with information like instrument, tonality (key), tempo (bpm) and music genre. This is a time-consuming task which is often unfeasible, which results in badly annotated databases and poor user experience when browsing them. In the field of Music Information Retrieval (MIR), a substantial effort has been put into automatically identifying the aforementioned characteristics for musical pieces. However, loops are inherently different from music pieces (*i.e.* with reduced instrumentation and short length). Therefore, existing MIR algorithms need to be tested and (possibly) adapted to work successfully in this scenario. Furthermore, new MIR tasks are emerging with the study of music loops including loop retrieval [12], loop detection [18], loop discovery [19] and extraction [27], loop recommendation [5], exploration of large loop databases [29], and automatic loop generation [26].

In this paper, we present FSLD, an open dataset with 9,455 music loops to support reproducible research in MIR. FSLD contains production-ready loops from Freesound which are distributed under Creative Commons licenses and can, therefore, be freely shared among the research community and industry. Part of the dataset has been manually annotated with information about rhythm, tonality, instrumentation and genre, in a similar way as

[1] https://freesound.org/
[2] https://www.looperman.com/
[3] https://sounds.com/
[4] https://splice.com/
[5] https://www.loopmasters.com/
[6] https://primeloops.com/

commercially available loop collections are annotated. The annotation service is made public [7] so that the community can work on enlarging the annotations of this collection. We expect this dataset to have an impact on the research community as it supports further research into several timely research topics which are also of great interest to the industry.

The rest of the paper is structured as follows. In Sec. 2, we present some of the datasets used in the literature for loop analysis. Sec. 3 details how the proposed dataset was collected and annotated. In Sec. 4, general statistics of the dataset are given. In Sec. 5 and 6, we present some potential applications and provide a benchmark of the dataset using some classic MIR tasks. Finally, in Sec. 7, we conclude and suggest future work directions.

2. RELATED WORK

Early work on the retrieval of loops focused on tempo extraction and transcription from drum loops [11, 15]. Gouyon et al. compared several tempo induction algorithms proposed in the ISMIR 2004 competition [15]. The loop dataset used in this work has been commonly used for evaluating tempo estimation algorithms and is divided into three subsets. One of these comprises two thousand audio loops (with tempo annotations) from Sound Effects Library. [8] These audio loops are not free and a license needs to be obtained to use them for research.

Automatic transcription of drum loops focuses on identifying when the different percussion instruments occur in a loop. Gillet and Richard used a collection of 315 drum loops for evaluating their system and provided "a compressed version of a few drum loops" [11]. The URL to the webpage the authors provide is broken and, presumably, the lower quality versions of the loops do not represent commercial-quality content. The authors also use this dataset for automatically retrieving drum loops from spoken queries [12]. This database was later used by Bello et al. for automatic rhythm modification and analysis of drum loops [1, 23].

The work from Gómez-Marín et al. explores rhythmic similarity measures for audio loops [14]. The authors validate the proposed metric using 9 drum break loops from Rhythm Lab. [9] The authors do not specify which are the drum loops used.

Font et al. presented a dataset of audio loops from Freesound [9] in their work on tempo estimation and a confidence measure for audio loops [10]. The authors use two commercial datasets, loops bundled with music production software Apple's Logic Pro [10] and Acoustica's Mixcraft, [11] and two community datasets. The first one is a private collection of loops downloaded from Looperman, which was previously used for research in [24]. Looperman does not allow the re-distribution of loops "as is", and considers as misuse the automatic download of their

loops. [12] A collection of 4000 loops from Freesound, obtained by searching Freesound for sounds with the queries "loop" and "bpm" is also proposed. The sounds' filenames, tags and textual descriptions are parsed to identify tempo annotations provided by the users. However, these annotations are not always accurate, and, to enable further work on audio loops, more information besides the tempo is desired.

In short, existing academic work which employs loops resorts to commercial samples as the source of data and open datasets do not have complete and reliable annotations. This makes it difficult to reproduce existing research. To promote open and accessible research on audio loops, we propose a free and distributable database of loops from Freesound, which provides production-ready sounds with high-quality annotations.

3. DATASET CREATION

In this section the process we have followed to create the dataset is described. We show how we collected the loops to annotate, how they were pre-analysed for a faster annotation procedure and explain what was annotated and how the annotation tool was implemented. Finally, we present how the dataset is distributed and organised.

3.1 Loop Selection

To select an initial pool of candidate loops, we followed the same methodology as in [10]: i.e., we retrieved sounds with both "loop" and "bpm" keywords on Freesound, resulting in 9,490 sounds. Using the Freesound API, it was straightforward to obtain these loops and their metadata—title, tags, textual description, and author's username.

3.2 Loop Annotation

We want the loops in our dataset to be annotated in a way which is similar to commercially available loops. This way, we make sure that the loop characterisation is compatible with industry standards. For this, we decided to annotate the loops' instrumentation, tempo, time signature, key and genre, as described below. The annotation was performed by 8 MIR researchers and students, with knowledge of electronic music production. To make the annotation procedure as efficient as possible, we created a web application for the annotators with several tools at their disposal, which can be seen in Fig. 1. This application was developed using Flask, [13] a web framework for Python.

This interface provides fields for the annotators to fill in the desired information, which will be described in the following sections. The instructions are provided on tooltips for quick access by annotators.

3.2.1 Instrumentation

Instead of annotating instruments in a traditional way, which would not be straightforward in heavily processed

---

Figure 1. The user interface provided to the annotators, available online. [7]

audio or more experimental loops, we chose to annotate general *roles* which can be useful for both music makers and automatic generation of music. We asked annotators to tick all the roles that apply to each loop. Usually, specific instruments could be easily assigned to a specific role. We present the roles along with some examples in Table 1.

Role	Example Instruments
Percussion	Drums, glitches, tuned percussion
Bass	Synth bass, fingered bass
Chords	Piano chords, guitar chords, synth pads
Melody	Instrument playing a melody, arpeggiator
Sound FX	Risers, cinematic sounds, foley, scratching
Vocal	Singing voice, spoken word, vocoder

Table 1. Instrumentation roles and the examples provided for each category.

3.2.2 Rhythmic Characteristics

We asked for annotations on three rhythmic aspects:

Tempo provides an easy measure of rhythmic compatibility and is the most common information provided in commercial loop databases. We ask annotators if the loop has a clear and steady tempo, to identify loops with constant tempo and clear beat (BPM value and steady tempo), with changing tempo (BPM value of the initial tempo

and no steady tempo), and loops with no clear beat but where the tempo can be inferred (BPM value and no steady tempo).

Meter is not a feature we see annotated as often as BPM, which might be due to the common use of 4/4 meter in electronic music. This feature is relevant to annotate, for calculating the number of bars in a loop, from its meter, tempo and duration.

Finally, as sometimes the length of the audio file is not the length of the loop, we also annotate if it is **well-cut**. If there is some silence at the beginning or the end of the file or if there is a "tail" (*e.g.* a decay of a reverb effect) when the audio is exported, it might not loop correctly just by staring the loop again when it finishes playing.

3.2.3 Tonal Characteristics

We annotate if the loop has prominent tonal content and, if so, to indicate a root key and mode that matches the tonal content of the loop (*i.e.*, root note from a chromatic scale and *Major/Minor* mode). We explained "prominent tonal content" as whether it is easy to sing along to the loop or to find a meaningful root note for the loop. For root key annotations, we asked to choose a note from a dropdown with 12 notes, or "Unknown" in case the key could not be found. For annotating mode, the annotators had the choice

of "Major" or "Minor" if the loop sounded good with one of these modes; "None" if the loop could not be clearly assigned to either "Major" or "Minor" (e.g. loop contains a single note); or "Unknown" for other cases.

3.2.4 Genre

We annotate genre in non-exclusive categories, where each is assigned to a loop if it can be used to make music in that genre. To create a taxonomy which would be similar to commercially available ones, we merged the taxonomies of Sounds.com and Splice. These were chosen as they provided several examples for each genre and had similar parent categories. We present the taxonomy in Table 2.

Genre	Examples
Bass Music	Dubstep, Drum and Bass, Jungle
Live Sounds	Rock, Jazz, Disco
Cinematic	Sound FX, Filmscore, Sci-Fi
Global	Reggae, Dancehall, Indian Music
Hip Hop	Trap, Boom Bap, Lofi Hip Hop
Electronic	Ambient, IDM, Chill Out
House / Techno	Deep House, Electro, Tech House
Other Dance Music	EDM, Psy Trance, Hardstyle

Table 2. Taxonomy of genres used for the annotation and examples for each category.

3.2.5 Loop Pre-Analysis and Annotation Tools

We performed a pre-analysis on the loops to obtain tempo, key and genre suggestions. To obtain the tempo information, we followed the same approach of [10], parsing the title, description and the tags of the loop for tempo information provided by users. To propose an initial key and mode to the annotators, we analysed the loops using the algorithm proposed by Faraldo et al. [7], which is implemented in the Essentia audio analysis library [3]. Finally, by taking the genre information from the textual metadata of the loops, we were able to map some of the sounds to the genres to annotate. The checkboxes were selected for the genres which either were mentioned or had a sub-genre mentioned in the textual metadata. Our annotators were familiar with the annotation procedure and took the pre-annotations only as suggestions to speed-up the annotation process.

In the annotation tool, at the top of the display is the loop's metadata: its unique sound id, title, author's username, and the tags and textual description provided by the author (see Fig. 1). The waveform of the loop and a playhead is also shown, which are linked to an audio player. The audio player always restarts the playback of the loop when it finishes, and triggers a metronome with the BPM provided in the BPM annotation field. We provide stop, play and pause controls for the loop and metronome and volume controls for the loop. A button which restarts only the metronome is also present. To ease finding a key and mode which suits the loop, we present a synthesizer which plays the chord present in the tonal annotation section. In case the mode selected is "None" or "Unknown", the synthesizer will just play the root note of the key selected. Using the computer's keyboard, the annotator can cycle

through the options for key and mode, in several octaves. Finally, buttons are provided for submitting the annotation when it is finished, saving the sound for later and discarding the sound in case it is not a loop.

3.3 Dataset Availability

The loops and corresponding annotations (provided in a JSON file) are publicly available on Zenodo. [14] This dataset can be divided into three subsets, defined by their level of annotations. These are:

- Multiple-annotations (MA): the loops annotated by at least two researchers. It contains 1,472 loops.
- Single-annotation (SA): the loops annotated by a single researcher. Currently contains 1,464 loops.
- Automatic-annotations (AA): the loops annotated by the analysis algorithms mentioned in Section 3.2.5. Contains 9,455 loops: the loops in MA and SA and 6,519 more.

In addition to the main dataset, we provide a repository [15] with the code used for the annotation tool interface and server, the pre-analysis that generated the subset of automatic annotations, and the analysis and potential applications presented in Sections 4, 5 and 6.

4. DATASET ANALYSIS

To understand the diversity and reliability of the dataset, we investigate the distribution of annotated characteristics and inter-annotator agreement.

4.1 Annotation Distribution

The human-annotated part of the dataset contains 1,579 sounds which, in total, have been annotated 2,809 times. The distribution of genres, instrumentation, and keys are shown in Tables 3 and 4 and the tempo histogram in Figure 2. It is well-balanced in terms of instrument and genre; reasonably balanced in terms of tempo, although 120 bpm dominates; and highly imbalanced in terms of key, with C Major and Minor dominating.

Percussion	54.95%	Bass Music	32.04%
Bass	19.10%	Live Sounds	21.38%
Chords	11.90%	Cinematic	19.95%
Melody	21.31%	Global	14.26%
FX	24.80%	Hip-hop	17.29%
Vocal	2.29%	House/Techno	29.05%
		Other Dance Music	25.63%

Table 3. Distribution of the instrument roles and genre in our dataset.

4.2 Inter-annotator Agreement

To measure the agreement of the annotators in our dataset, we measure the inter-annotator agreement for the MA annotations subset. To do this, we use two metrics: proportion of overall agreement (Agr.) for all the annotations,

---

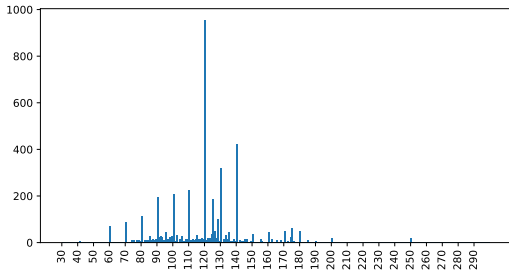[14] https://zenodo.org/record/3967852
[15] https://github.com/aframires/
freesound-loop-annotator

Figure 2. Distribution of BPMs in FSLD.

Key	Maj	Min	None	Unknown
C	9.63%	8.38%	3.65%	0.95%
C#	1.38%	2.84%	0.60%	0.43%
D	3.31%	5.37%	1.85%	0.39%
D#	1.29%	2.49%	0.73%	0.21%
E	2.28%	3.65%	1.20%	0.26%
F	4.64%	4.43%	1.16%	0.34%
F#	1.12%	2.49%	1.12%	0.13%
G	2.66%	4.25%	1.93%	0.43%
G#	1.72%	2.58%	0.90%	0.13%
A	3.01%	5.50%	1.68%	0.26%
A#	1.50%	1.89%	0.52%	0.04%
B	0.82%	3.01%	0.64%	0.21%

Table 4. Distribution of the keys in our dataset.

and positive and negative agreement (PA and NA) [8] for binary classification tasks. The proportion of overall agreement reflects the number of cases when both annotators agree on a label, and is calculated by dividing their number by the total number of annotations. This overall metric does not distinguish the agreement in positive and negative cases, so for the binary annotation tasks we also calculated the positive and negative agreement. The formulas for calculating these are given in Eq. 1, where the variables represent the annotations by the annotators (e.g., NP = first annotator answered negative, second positive).

$$PA = \frac{2PP}{2PP + NP + PN} , \quad (1)$$

$$NA = \frac{2NN}{2NN + NP + PN} , \quad (2)$$

Table 5 presents the results for this analysis. We can see that overall, the values for the agreement are high. Bass, melody and chords have a lower positive agreement value, despite the high negative agreement. This might indicate that annotators are not able to easily distinguish if an element should fit in one of the 3 roles, but can say when it is not present. The lower value for root key agreement indicates that several keys are used to describe the same sounds. This fits our annotating indications, where we asked annotators to select a key which sounds good with the loop and therefore, personal taste may arise in this choice. Finally, the positive agreement for genres always has values lower than 65%, which might be due to how genre might be perceived subjectively between annotators.

5. BENCHMARKING MIR TASKS

To demonstrate the usefulness of this dataset, we use it in several short case studies. To benchmark tempo, we fol-

Char.	Sub-Char.	Agr.	PA	NA
Inst.	Percussion	85.16%	86.62%	83.35%
	Bass	76.73%	45.83%	85.19%
	Melody	82.33%	60.57%	88.61%
	Chords	87.40%	47.35%	92.84%
	FX	72.04%	43.61%	81.41%
	Vocal	98.66%	71.88%	99.31%
Tempo	BPM	87.84%	NA	NA
	Signature	97.84%	NA	NA
	Well Cut	86.88%	92.73%	32.82%
Key	Root	67.56%	NA	NA
	Mode	69.80%	NA	NA
Genre	Bass Music	69.50%	53.26%	77.37%
	Live Sounds	80.09%	55.28%	87.19%
	Cinematic	81.66%	57.14%	88.33%
	Global	82.33%	51.53%	89.19%
	Hip-Hop	79.05%	31.30%	87.64%
	House/Techno	69.35%	48.56%	78.17%
	Other	73.53%	45.64%	82.50%

Table 5. Inter-annotator agreement for the MA subset.

lowed the evaluation approach of [10] and used the *Accuracy 1* and *Accuracy 2* presented in [16], together with the *Accuracy 1e* proposed in [10]. Due to space constraints, here we only report the mean of the 3 accuracies. Full results can be seen in an accompanying website. [16] The algorithms selected for the tempo benchmarking were the following (details for each algorithm can be found in respective papers):

- **Percival [21]:** We use both the original implementation and the one provided in Essentia.
- **Zapata [31]:** Implementation provided in Essentia.
- **Degara [6]:** We also use Essentia's implementation.
- **Böck [2]:** We use the 3 variants available in the Madmom library [17] : COMB, ACF and DBN.

We validate tempo estimation algorithms on the 3 proposed subsets. Key estimation is only validated on the MA and SA subsets as we do not have original uploader annotations for key. The MA subset, which has at least 2 annotations per loop, was analysed in two ways: BOTH and EITHER. In BOTH, we run the MIR algorithms exclusively on the loops which have the same labels from both annotators. In EITHER, the output of the algorithm was deemed correct if it was at least one of the annotated labels. The results are presented in Table 6

Algorithm	AA	SA	BOTH	EITHER
Percival14	58.09	62.98	65.75	84.13
Percival14e	57.82	64.00	65.49	84.98
Zapata14	51.81	58.79	58.99	77.97
Degara12	52.32	58.77	59.31	79.16
Bock15COMB	44.42	51.17	52.92	71.35
Bock15ACF	48.65	51.96	54.75	74.90
Bock15DBN	45.76	50.60	52.32	70.90

Table 6. Evaluation of tempo estimation algorithms in the proposed subsets.

---

[16] https://aframires.github.io/freesound-loop-annotator/
[17] https://github.com/CPJKU/madmom

We can see that the results are similar to the ones obtained in [10], with Percival14 having better accuracy across all the datasets. We can see that the accuracy increases from AA to SA, and from SA to BOTH. This might be due to the user-annotated loops having incorrect annotations; it may also be that when both annotators agree on a tempo, the tempo is strong and defined. The EITHER evaluation gives the largest accuracies, which may be due to its broader criteria for considering tempos correct.

For benchmarking key estimation algorithms, we used the evaluation metrics from MIREX, [18] which evaluates how *close* the estimated key and the annotated key are to provide an accuracy. The algorithms compared in the evaluation were the following:

- **EDMKey [7]:** We use the implementation in Essentia, with 4 key profiles: Krumhansl [17], Temperley [30], Shaath [25] and the one proposed in [7].
- **EssentiaBasic [3]:** Essentia's implementation of the algorithm presented by Gomez [13].
- **QMUL [20]:** We use the Key Detection implementation available in QM Vamp Plugins. [19]

In Table 7, we present part of the results of the key estimation evaluation. Due to lack of space, only the final MIREX scores for each dataset are presented. The full results can be seen in the accompanying website. [16]

Algorithm	SA	BOTH	EITHER
Edmkey	72.26	88.25	85.63
EdmkeyKrumhansl	66.99	84.85	82.98
EdmkeyTemperley	61.46	71.78	71.77
EdmkeyShaath	72.38	88.25	85.63
EssentiaBasic	71.25	88.80	85.30
QMULKeyDetector	35.09	42.15	46.25

Table 7. Evaluation of key estimation algorithms in the proposed subsets.

We see that EssentiaBasic and EDMkey are the best performing algorithms here. EDMKey has been specially tuned to be used for EDM, which might make it more suitable to the loops we are annotating. We again see that the accuracy increases from SA to BOTH, which might indicate again that when the key is clear and defined, the algorithms are also able to correctly identify it.

6. MUSIC GENERATION AND DECOMPOSITION

Another way the dataset is valuable is for creating synthetic datasets of songs for evaluating loop-extraction algorithms, such as [27]. We created 100 random songs, each using 5 random drum loops and 5 non-drum loops (chosen from a subset of 4/4, 120-bpm, 1-bar, single-instrument loops for which there was no disagreement among the annotators on the instrument role). Each song is a random arrangement of the loops, either in a *sparse* arrangement, in which one drum and one non-drum loop occurs per bar, or a *dense* one, in which 4 loops occur per bar (i.e., 2 drum

and 2 non-drum). For comparison, we also recreated the "composed" and "factorial" layouts from [27]. Examples of each layout are shown on the accompanying website.

We used the public implementation [20] of [27] to extract loops for each song, informed with the true number of loop segments (4 or 10) and the true downbeat boundaries. The metrics SDR, SIR and SAR (the signal to distortion, interference and artefacts ratios [22]) are reported in the left part of Table 8.

These are normally computed by trying all permutations of estimated sources to true sources and using that which maximises the score. This is infeasible for permutations of 10 items, so we first find the permutation that maximises the similarity between the source and true loop spectra.

Layout	SDR	SIR	SAR	F1	Acc.
Sparse	–5.2	–3.9	15.8	0.194	0.691
Dense	–7.9	–7.3	14.4	0.294	0.542
Composed	12.5	18.4	22.6	0.585	0.546
Factorial	19.8	29.2	24.1	0.560	0.551

Table 8. Evaluation of loop source quality (SDR, SIR, SAR) and estimated layouts (F-measure and accuracy) for each song layout.

This permutation is also used to evaluate the quality of the estimated layout. We binarize each row of the estimated layout, using the row's mean as threshold. We then compute the raw accuracy (as in [27]), but here we propose also using the F-measure, so as not to weight true negatives unduly. The results are in the right columns of Table 8.

SDR, SIR and SAR are all lower for the random 10-part songs than for the 4-part songs, showing that we have created a more challenging testing ground for loop extraction systems. For the layout evaluation, our evaluation makes clear that the raw accuracy gives undue weight to true negatives: the highest accuracy was obtained for the sparse layouts, despite having the lowest F-measure. This short evaluation is a proof of concept; with more space, we could study the impact of the instrumentation, number of loops, loop duration, and other factors on the separation quality. We can also generate layouts with loops of many durations and evaluate hierarchical loop extraction systems.

7. CONCLUSION

In this paper, we presented our work on addressing the lack of standard loop datasets to carry MIR tasks. We presented FSLD, a dataset of audio loops annotated at a level similar to commercial loop collections. These loops are licensed for redistribution and can be used and redistributed for research purposes. We provide a detailed analysis of the dataset and its annotations and provided several use cases for tempo and key benchmarking, music generation and loop separation. Furthermore, we present the online annotation tool used to build the dataset, and we make it available online so other researchers and the general public can contribute and extend the dataset.

---

[18] https://www.music-ir.org/mirex/wiki/2019:
Audio_Key_Detection
[19] https://vamp-plugins.org/plugin-doc/
qm-vamp-plugins.html#qm-keydetector

[20] https://github.com/jblsmith/loopextractor

8. ACKNOWLEDGEMENT

9. REFERENCES

[1] Juan P. Bello, Emmanuel Ravelli, and Mark B. Sandler. Drum sound analysis for the manipulation of rhythm in drum loops. In *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006.

[2] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 2015.

[3] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, Jose R. Zapata, and Xavier Serra. Essentia: an audio analysis library for music information retrieval. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, 2013.

[4] Mark Jonathan Butler. *Unlocking the groove: Rhythm, meter, and musical design in electronic dance music*. Indiana University Press, 2006.

[5] Bo-Yu Chen, Jordan Smith, and Yi-Hsuan Yang. Neural loop combiner: Neural network models for assessing the compatibility of loops. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020.

[6] Norberto Degara, Enrique Argones Rúa, Antonio Pena, Soledad Torres-Guijarro, Matthew EP Davies, and Mark D Plumbley. Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):290–301, 2011.

[7] Ángel Faraldo, Emilia Gómez, Sergi Jordà, and Perfecto Herrera. Key estimation in electronic dance music. In *38th European Conference on Information Retrieval*, pages 335–347. Springer-Verlag, 2016.

[8] Alvan R Feinstein and Domenic V Cicchetti. High agreement but low kappa: I. the problems of two paradoxes. *Journal of clinical epidemiology*, 43(6):543–549, 1990.

[9] Frederic Font, Gerard Roma, and Xavier Serra. Freesound technical demo. In *ACM International Conference on Multimedia*, 2013.

[10] Frederic Font and Xavier Serra. Tempo estimation for music loops and a simple confidence measure. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016.

[11] Olivier Gillet and Gaël Richard. Automatic transcription of drum loops. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[12] Olivier Gillet and Gaël Richard. Drum loops retrieval from spoken queries. *Journal of Intelligent Information Systems*, 24(2):159–177, 2005.

[13] Emilia Gómez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, 18(3):294–304, 2006.

[14] Daniel Gómez-Marín, Sergi Jordà, and Perfecto Herrera. Pad and sad: Two awareness-weighted rhythmic similarity distances. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 2015.

[15] Fabien Gouyon, Anssi Klapuri, Simon Dixon, Miguel Alonso, George Tzanetakis, Christian Uhle, and Pedro Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.

[16] Fabien Gouyon, Anssi Klapuri, Simon Dixon, Miguel Alonso, George Tzanetakis, Christian Uhle, and Pedro Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.

[17] Carol L. Krumhans. Cognitive foundations of musical pitchl. *Music Perception: An Interdisciplinary Journal*, 9:476–492, 07 1992.

[18] Patricio López-Serrano, Christian Dittmar, Jonathan Driedger, and Meinard Müller. Towards modeling and decomposing loop-based electronic music. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016.

[19] Patricio López-Serrano, Christian Dittmar, and Meinard Müller. Finding drum breaks in digital music recordings. In *Proceedings of the International Symposium on Computer Music Multidisciplinary Research*, 2017.

[20] Katy Noland and Mark Sandler. Signal processing parameters for tonality estimation. In *Audio Engineering Society Convention 122*, May 2007.

[21] Graham Percival and George Tzanetakis. Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1765–1776, 2014.

[22] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel PW Ellis. mir_eval: A transparent implementation of common MIR metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014.

[23] Emmanuel Ravelli, Juan P. Bello, and Mark Sandler. Automatic rhythm modification of drum loops. *IEEE Signal Processing Letters*, 14(4):228–231, 2007.

[24] Gerard Roma. *Algorithms and representations for supporting online music creation with large-scale audio databases*. PhD thesis, Universitat Pompeu Fabra, June 2015.

[25] ibrahim Sha'ath. Estimation of key in digital music recordings. Master's thesis, Birkbeck College, University of London, 2011.

[26] Z. Shi and G. J. Mysore. LoopMaker: Automatic creation of music loops from pre-recorded music. In *Proceedings of SIGCHI International Conference on Human Factors in Computing Systems*, 2018.

[27] Jordan B. L. Smith and Masataka Goto. Nonnegative tensor factorization for source separation of loops in audio. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 171–175, 2018.

[28] Glenn Stillar. Loops as genre resources. *Folia Linguistica*, 39(1-2):197 – 212, 2005.

[29] S. Streich and B. S. Ong. A music loop explorer system. In *Proceedings of International Computer Music Conference*, 2008.

[30] David Temperley. What's key for key? the Krumhansl-Schmuckler key-finding algorithm reconsidered. *Music Perception: An Interdisciplinary Journal*, 17(1):65–100, 1999.

[31] José R Zapata, Matthew EP Davies, and Emilia Gómez. Multi-feature beat tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):816–825, 2014.