TISMIR

JSD: A Dataset for Structure Analysis in Jazz Music

DATASET

STEFAN BALKE
JULIAN RECK
CHRISTOF WEIß
JAKOB ABEßER
MEINARD MÜLLER

*Author affiliations can be found in the back matter of this article

ubiquity press

ABSTRACT

Given a music recording, music structure analysis aims at identifying important structural elements and segmenting the recording according to these elements. In jazz music, a performance is often structured by repeating harmonic schemata (known as choruses), which lay the foundation for improvisation by soloists. Within the fields of music information retrieval (MIR) and computational musicology, the Weimar Jazz Database (WJD) has turned out to be an extremely valuable resource for jazz research. Containing high-quality solo transcriptions for 456 solo sections, the dataset opened up new avenues for the understanding of creative processes in jazz improvisation using computational methods. In this paper, we complement this dataset by introducing the Jazz Structure Dataset (JSD), which provides annotations on structure and instrumentation of entire recordings. The JSD comprises 340 recordings with more than 3000 annotated segments, along with a segment-wise encoding of the solo and accompanying instruments. These annotations provide the basis for training, testing, and evaluating models for various important MIR tasks, including structure analysis, solo detection, or instrument recognition. As an example application, we consider the task of structure boundary detection. Based on a traditional novelty-based as well as a more recent data-driven approach using deep learning, we indicate the potential of the JSD while critically reflecting on some evaluation aspects of structure analysis. In this context, we also demonstrate how the JSD annotations and analysis results can be made accessible in a user-friendly way via web-based interfaces for data inspection and visualization. All annotations, experimental results, and code for reproducibility are made publicly available for research purposes.

CORRESPONDING AUTHOR:

Stefan Balke

International Audio Laboratories Erlangen

Am Wolfsmantel 33, 91058 Erlangen, DE

stefan@balke.at

KEYWORDS:

Dataset; Music Structure Analysis; Boundary Detection; Instrument Recognition; Jazz

TO CITE THIS ARTICLE:

Balke, S., Reck, J., Weiß, C., Abeßer, J., and Müller, M. (2022). JSD: A Dataset for Structure Analysis in Jazz Music. *Transactions of the International Society for Music Information Retrieval,* 5(1), 156–172. DOI: https://doi.org/10.5334/tismir.131

1. INTRODUCTION

In the interdisciplinary research field of Music Information Retrieval (MIR), one central research area is commonly subsumed under the name of *music structure analysis*. The general objective is to segment an audio recording with regard to various musical aspects, for example by identifying recurrent themes or detecting temporal boundaries between contrasting musical parts (Müller, 2015). One main challenge of structure analysis is that music is highly complex and diverse. Being organized in a hierarchical way, structure in music arises from various relationships between its basic constituent elements. The principles used to create such relationships include repetition, contrast, variation, and homogeneity (Paulus et al., 2010; Peeters, 2007). As a consequence, many different approaches to derive musical structures have been developed (see Dannenberg and Goto, 2008; Müller, 2015; Nieto et al., 2020; Paulus et al., 2010, for an overview). For evaluating the performance of automated procedures, one requires *reference annotations* typically generated by human experts. However, due to ambiguity and variety of musical structures, there may be significant cross-annotator differences, thus making the evaluation of automated procedures a research problem itself (Nieto et al., 2014; Smith et al., 2011; McFee and Kinnaird, 2019).

In jazz music, compared to other genres, higher-level musical structures are often less ambiguous as they are based on the succession of solo sections that follow the same characteristic harmonic schema of a certain length (known as the form). In the following, we call one full pass through the form a *chorus*. As a concrete example, Figure 1b shows the structure of the jazz recording "Jordu"[1] by Clifford Brown. In the first section, which is also called the *head* or *theme* section, the ensemble introduces the main theme or melody of the piece and the accompanying harmonic progression. In the following sections, the ensemble's musicians take turns playing solos, where they improvise on the repeating harmonic progression using different instruments such as a trumpet, saxophone, piano, and drums. At the end, the main theme is repeated, as presented in the first section. As for music structure analysis, central subtasks are to automatically segment the music recording according to these sections, to detect the sections' functions (theme, solo), and to identify the solo as well as accompanying instruments. As one main contribution of this paper, we provide a novel collection of reference annotations on structure and instrumentation for jazz recordings (see Figure 1a).

In MIR, the availability of well-documented and freely available reference annotations is of crucial importance for the development and evaluation of computational music analysis methods (Serra, 2014). A prominent example is the dataset MedleyDB (Bittner et al., 2014), which has triggered significant research efforts in areas such as melody extraction and source separation. Similarly, the dataset SALAMI (Smith et al., 2011) constitutes an excellent testbed for research in music structure analysis. As for jazz music, the Weimar Jazz Database (WJD) (Pfleiderer et al., 2017) has opened up new avenues for studying jazz improvisation using computational methods. Following these lines, we introduce in this paper the Jazz Structure Dataset (JSD), which provides more than 3000 annotated segments, along with a segment-wise encoding of the solo and accompanying instruments, for 340 jazz recordings. Being based on the same recordings, the JSD and the WJD provide in the initial version complementary annotations for the same underlying audio material.

The main contributions and the structure of this paper can be summarized as follows. In Section 2, we review prior work and discuss the relationship between JSD and existing datasets in more depth. Then, in Section 3, we cover aspects with regard to creation, organization, and accessibility of the JSD. In particular, we describe the annotations' properties and statistics and show how these annotations can be accessed and studied via a user-friendly web-based interface (see also Figure 1). As another main contribution of this paper, we show how the JSD may serve as basis for training, testing, and evaluating MIR tasks. In Section 4, we consider the task of structure boundary detection as a concrete example scenario. In particular, we use two conceptually different approaches: a classical novelty-based approach (Foote, 2000) and a more recent data-driven approach using deep learning (Grill and Schluter, 2015; Ullrich et al., 2014). While indicating the potential of the JSD, these experiments also provide baseline implementations and results for future research. Furthermore, we demonstrate how our web-based interfaces enable fellow researchers to easily access and understand experimental results (see also Figure 8). We present conclusions and future work in Section 5. All annotations, experimental results, and code to reproduce the figures and experimental results (including the extracted audio features) are made publicly available for research purposes via an online repository.[2]

2. RELATED WORK

As already noted in the introduction, the availability of publicly available datasets along with well-documented reference annotations is central for sustainable and reproducible research in MIR (McFee et al., 2019). The "Real World Computing" (RWC) database (Goto et al., 2002) is an early example of a systematic music database that was compiled specifically for research purposes. Besides the audio material, its main value lies in the availability of

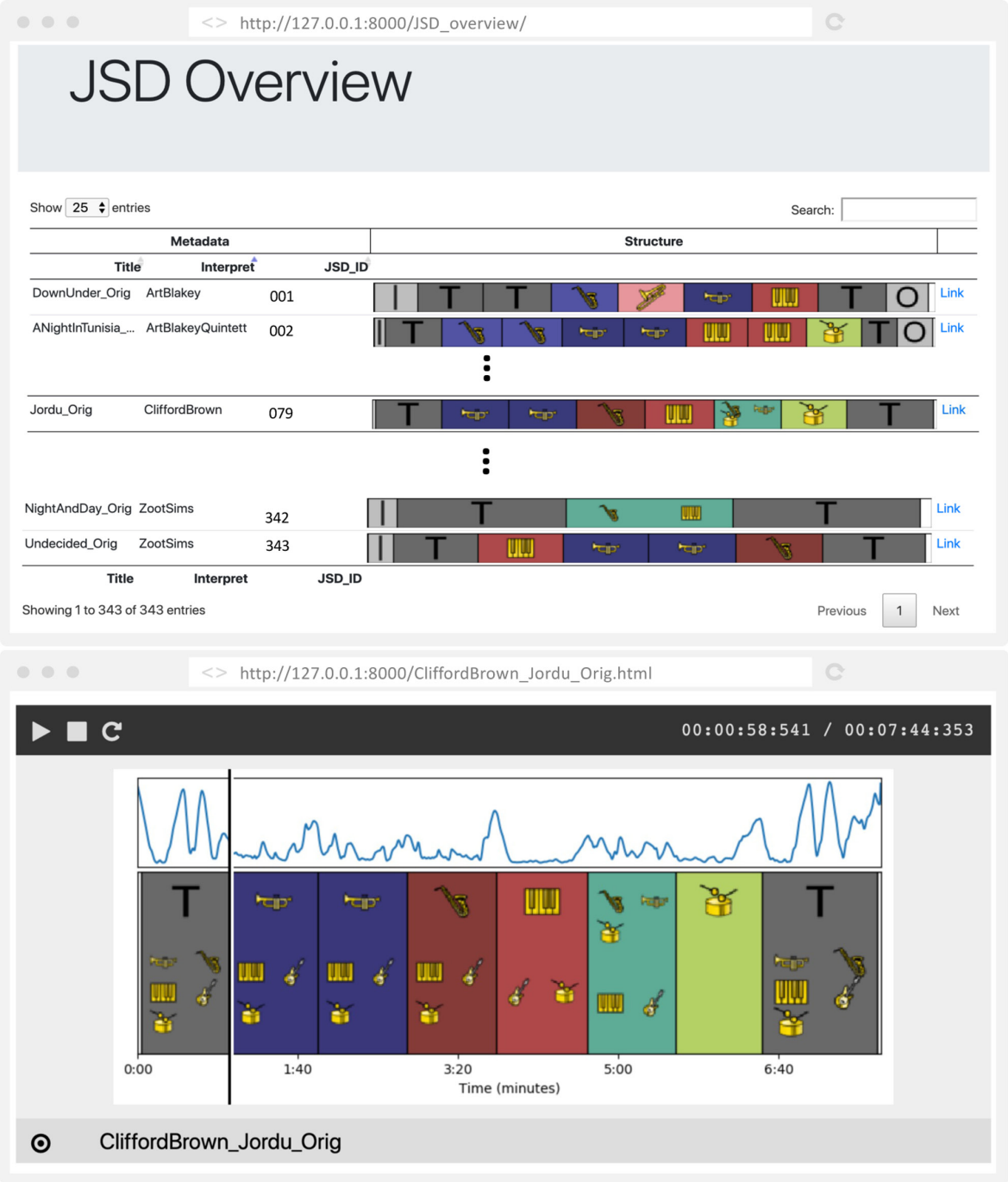

Figure 1: (a) Above: overview of the Jazz Structure dataset (JSD). **(b)** Below: running example "Jordu" by Clifford Brown. The figure shows a novelty function and structure annotations within a web-based interface (T = theme; the pictograms indicate the current soloist and the accompaniment).

various musical annotations (Goto, 2006). Another excellent example is MedleyDB, which is a dataset of royalty-free multitrack recordings and annotations for research in various areas including melody extraction, instrument recognition, and source separation (Bittner et al., 2014).

In recent years, several more specialized data collections have been released considering specific musical genres or addressing the needs of specific research directions and MIR tasks. For example, one of the most extensive databases for computational ethnomusicology was compiled within the CompMusic research project (Serra, 2014). The individual corpora (comprising Carnatic, Hindustani, Turkish, Chinese, and Andalusian music) along with annotations and metadata are well documented and hosted on the web-platform Dunya.[3] Other corpora have had a substantial impact on MIR research. For example, the Meertens Tune Collections[4] have been the basis for research on melodic similarity (van Kranenburg et al., 2019). The COFLA dataset is fundamental for the computational study of Flamenco music (Kroher et al., 2016). The Ballroom dataset has been used widely for genre classification and rhythm analysis (Gouyon et al., 2004). The Beatles

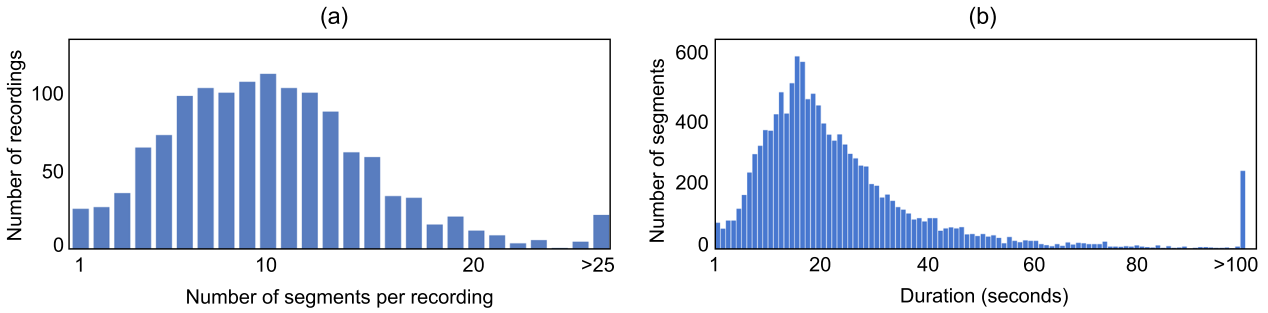

Figure 2: Statistics for the large-scale annotations of the SALAMI database. **(a)** Distribution of the number of segments per recording. The total number of segments is 12634. In average, each recording consists of 10.37 segments. **(b)** Distribution of segment durations (seconds). In average, a segment has a duration of 25.21 seconds.

dataset is famous for its extensive usage in automatic chord recognition (Harte et al., 2005). The JAAH dataset comprises annotations for 113 selected tracks to apply chord recognition to jazz music (Eremenko et al., 2018). Many of these collections do not only contain data, but also tools for parsing, processing, and accessing the data. A recent example for such a collection is the Erkomaishvili Dataset (Rosenzweig et al., 2020), which not only contains historic tape recordings and various kinds of annotations of Georgian songs, but also web-based interfaces for conveniently accessing and understanding the data.

There are many more freely accessible datasets that have been of central importance for MIR and computational musicology. Giving a comprehensive overview is beyond the scope of this article, and we refer to the website of the International Society for Music Information Retrieval (ISMIR)[5] for further links. In the following, we will have a closer look at two datasets that have a strong relationship with JSD.

2.1 SALAMI DATASET

The "Structural Analysis of Large Amounts of Musical Information" (SALAMI) database consists of over 2400 human-made structure annotations of 1356 music recordings from various music genres (Smith et al., 2011). Structure annotations are specified on three different levels, namely a functional, a large-scale, and a small-scale level. More than half of the tracks (65.9%) were independently annotated by two people, which allows for studying structural ambiguities and cross-annotator differences (Flexer, 2014). In Section 4, we will use the SALAMI dataset for a cross-dataset experiment considering the large-scale annotations. Most recordings of the SALAMI dataset have a duration of 3–5 minutes, with a mean duration of 4.65 minutes. In total, SALAMI contains 12634 large-scale segments. Figure 2a shows the distribution of the number of segments per recording, and Figure 2b the distribution of the durations of recordings (given in seconds). Currently, SALAMI is one of the largest publicly available databases on music structure and is widely used for research in MIR. See Nieto et al. (2020) for an overview.

2.2 WJD DATASET

The Weimar Jazz Database (WJD) is a comprehensive and publicly available collection of jazz solo transcriptions (Pfleiderer et al., 2017). Spanning a wide range of renowned jazz musicians covering periods from 1925 to 2009 (e.g., Louis Armstrong, Charlie Parker, or Chris Potter), the WJD is an invaluable resource for exploring the cognitive and cultural foundations of jazz solo improvisation. The core of the collection consists of 456 transcriptions of instrumental solos extracted from 340 tracks[6] stemming from 197 different records. The solos were manually annotated by students of musicology and jazz at the University of Music Franz Liszt Weimar and stored as piano-roll-like representations, which are time-aligned to the original audio recordings. In addition, the database offers various music-related annotations such as chord sequences or beat positions. Thanks to its high-quality annotations, the WJD provides a controlled environment for systematic experiments in MIR. Among others, it has served for investigations on melodic phrasing (Frieler et al., 2016), tonal complexity (Weiß et al., 2018), intonation and pitch modulation analysis (Abeßer et al., 2017), solo voice enhancement (Balke et al., 2017), and swing analysis (Dittmar et al., 2018).

A critical issue that the JSD shares with other datasets (e.g., Beatles, SALAMI, CompMusic) is the copyright protection of the audio recordings underlying the annotations. Therefore, while the annotations are publicly available, the audio material is not. This restricts the usefulness of datasets for audio-centered research, where both the annotations and the corresponding audio material are required. Since we decided to take the WJD as the basis for the JSD and thus were bound to the selected recordings, copyright-free music was not an alternative. However, in future versions of the JSD, alternative datasets where the audio recordings are available could become relevant, e.g., the Niven

Figure 3: Examples of structure annotations on the chorus and solo level.

Jazz Collection, which in itself provides the potential for several research questions in MIR and musicology.[7] Exact specifications of the audio material by means of identifiers as provided by MusicBrainz[8] allow researchers to purchase the recordings. In case of the WJD, many of the recordings are also available on video-sharing platforms such as YouTube. Balke et al. (2018) present a web-based approach to identify and link most of the jazz recordings underlying the WJD to corresponding videos on YouTube.[9]

2.3 POSITIONING OF JSD

We now want to place our novel jazz structure dataset (JSD) within the context of the previously described datasets. As for the underlying audio material, the JSD is based on the same recordings (audio tracks) as the WJD. While the WJD provides fine-grained melodic annotations for a selection of *solo sections*, the JSD comprises structural annotations of *entire recordings*. In this sense, the JSD complements the WJD. While covering structural aspects and being inspired by SALAMI, the JSD differs from SALAMI in several ways. First, the recordings of JSD are not contained in SALAMI, thus yielding an independent dataset for training and testing structure analysis algorithms. Second, the JSD contains structural annotations as well as segment-wise annotations of the instrumentation, including solo and accompaniment instruments. Finally, the JSD not only provides annotations, but also tools for parsing, analyzing, and accessing the data. In this respect, it resembles the approach described by Rosenzweig et al. (2020). In summary, the JSD builds bridges to several existing datasets, while complementing them in different ways.

3 JAZZ STRUCTURE DATASET (JSD)

In this section, we describe the Jazz Structure Dataset including its organization and characteristics. Furthermore, we introduce web-based interfaces that yield a direct and intuitive access to the data.

3.1 STRUCTURE ENCODING

Many jazz recordings follow a particular fixed structure. In case of JSD, we make the assumption that a jazz performance is built on a characteristic harmonic schema, which stems from the harmonic accompaniment of the main melody (theme) and often comprises several structural parts (possibly including a modulation to a different key). A full cycle of the harmonic schema (form) is called a *chorus* and often comprises 16 or (far) more musical measures. In the first chorus, the ensemble introduces the main melody (theme) of the song. In the following choruses, the ensemble's musicians alternate in playing solos, where they improvise over the harmonic schema (also reflecting on melodic material of the theme). At the end, the main theme is repeated in its original form, as presented in the first time through the form. Possibly, there is also an additional intro and outro (Sikora, 2019).

Figure 3 visualizes the structure of two examples, where the labels `T`, `I`, and `O` denote the theme, intro, and outro sections, respectively. The solo choruses are indicated by icons of the solo instrument. As can be seen in our running example "Jordu" by Clifford Brown, the trumpet solo comprises two choruses. In general, a solo may consist of one or several choruses. Also, the original theme at the beginning and end of a jazz performance may be repeated. For example, this is the case for the song "Juju"[10] by Wayne Shorter, where the initial theme occurs twice in a row and the first saxophone solo consists of six choruses. Also, this example shows that the overall musical structure may deviate from the prototype pattern described above.

As indicated by Figure 3, we consider musical structures on two different levels: the *chorus level* and the *solo level*. The chorus level is a refinement of the solo level, where a solo consists of one or several choruses. Intuitively, the choruses correspond more to the repetition properties (in terms of the underlying characteristic harmonic schema), whereas the solo sections to homogeneity properties (in terms of the solo and accompanying instruments).

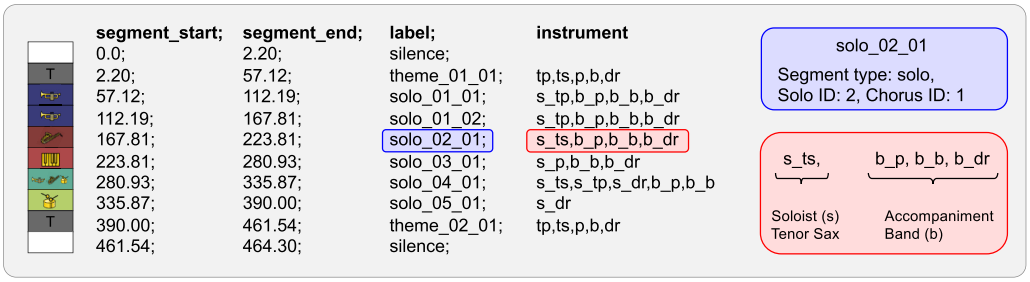

Figure 4: Raw annotation format for "Jordu" as contained in the JSD. Each row of the CSV file corresponds to a segment. The columns indicate the start time, the end time, the label, and the instrumentation of each segment.

In our encoding, both levels are considered simultaneously. To better understand our conventions, we consider the encoding of our running example (see Figure 4). We represent the musical structure of a jazz recording by a list of *labeled segments*. Each segment is specified by its start time and end time (given in seconds). Motivated by the fact that a theme or solo section may consist of one or several contiguous choruses, we use extended segment labels that contain a solo identifier (enumerating theme and solo sections) and a chorus identifier (enumerating choruses within a theme or solo section). For example, the first solo in "Jordu" (played by the trumpet) lasts for two choruses. We use the label `solo_01_01` to denote the first chorus of the first solo. The label `solo_01_02` then denotes the second chorus of the first solo. The second solo (played by saxophone) corresponds to only one chorus, which goes with the label `solo_02_01`, and so on. Similarly, the starting and ending theme sections may last one or even several choruses. Here, we use labels such as `theme_01_01` (as for the first chorus in "Jordu") and `theme_02_01` (as for the last chorus in "Jordu"). In case there is an intro or outro section, we use the labels `intro` and `outro`, respectively. Note that, at the beginning and end of a music recording, there is typically a short duration of silence (or other non-musical events such as applause). Even though not being sections from a musical point of view, we encode these non-musical aspects by two (possibly very short) additional segments labeled as `silence`.[11] The instrument labels will be discussed in Section 3.3.

3.2 STRUCTURE ANNOTATIONS

The JSD annotations were generated in a manual process by a group of three semi-professional musicians with a background in jazz music listening and performance. At first, the annotation process was defined by the main author (trumpet player, several years of experience in jazz combo playing) and given to the annotators, including the selection of pieces, the choice of the annotation tool, and the definition of the task (what is a boundary). While listening to each of the 340 recordings and relying on a leadsheet, the annotators marked the start time of each segment on the chorus level. To this end, the program *Sonic Visualiser*[12] was used for playback, synchronous display of a recording's spectrogram representation, and segment annotation. A segment's end time (except for the last `silence` segment) is given by the next segment's start time. Note that segments on the solo level can be easily derived from the label conventions described in Section 3.1.

The resulting temporal annotations were then cross-checked and refined by the main author. While there is high agreement about the order and the overall start position of the choruses, determining the exact location of the boundaries involves a higher degree of subjectivity. Here, we aimed for annotating the first downbeat in the *harmonic* schema of the chorus, which implies that the theme or solo *melody* may start earlier (due to pickups) or later (after the downbeat). In the annotation process, we identified two major sources of ambiguity. First, the identification of the downbeat within complex rhythmic situations may be difficult (metrical ambiguity). Second, the exact downbeat position is challenging (onset ambiguity). To evaluate this ambiguity, we took a subset of six recordings and let another expert annotator, who was not involved in the previous annotation process, refine the structural boundaries. From the 57 boundaries in these recordings, 37 were refined (64%). Among these, 24 were changed by at least 0.1 seconds pointing to a metrical ambiguity, However, only 13 boundaries (23%) were adjusted by more than 0.5 seconds and none of them by more than two seconds. From this, we conclude that metrical ambiguity of the chorus boundaries is less relevant for structure analysis tasks. In particular, our evaluation with the larger tolerance threshold of 3 seconds (Section 4.3) is insensitive to this problem.

Table 1 shows the number of occurrences and the total duration of different segment types. In total,

Type	# Segments	Total duration (min)
Intro	229	59.76
Theme	813	546.74
Solo	2223	1325.31
Outro	80	35.15
Silence	680	36.93
	Σ 4025	2003.89

Table 1: Overview of annotated (chorus-level) segments for the 340 recordings. From the segments, we derive 4365 segment boundaries (these include the 4025 start positions of each segment plus the 340 end positions of the last segments) from which 3005 are musical and 1360 non-musical.

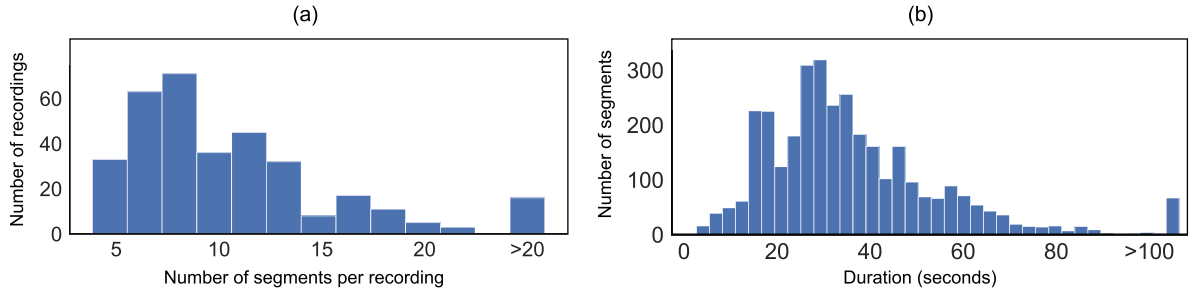

Figure 5: (a) Distribution of number of (chorus-level) segments per recording (`silence` segments are discarded). The total number of segments is 3345 (sum of Intro, Theme, Solo, and Outro segments, excluding `silence` segments). On average, a recording consists of 3345/340 ≈ 9.84 segments. **(b)** Distribution of segment durations (seconds) of all 3345 segments. On average, a segment has a duration of 35 seconds.

there are 3345 musical (non-silence) segments (Intro, Theme, Solo, Outro) and 680 (two per recording) silence segments. Among the musical segments, there are 2223 segments corresponding to solo choruses, 813 corresponding to themes, 229 intros, and 80 outros. Every recording in JSD consists of at least one theme and one solo segment, while 67.35% of the recordings contain an intro and 23.53% an outro.

For example, in Figure 4 the annotations of "Jordu" are shown. On the chorus level, this recording consists of 10 segments, of which 8 are musical and 2 non-musical (the small silence segments at the beginning and at the end). Furthermore, including the start and end positions of the recording, the example counts 11 segment boundaries. We further distinguish between musical and non-musical boundaries. A "musical boundary" is a boundary between two musical segments. The other boundaries are called "non-musical boundaries". In the example, out of the 11 boundaries, 7 are considered "musical" and 4 "non-musical" (start of the recording, transition from silence segment to theme, transition from theme to silence at the end, end of the recording).

Figure 5a gives an overview of the number of musical (i.e., non-silence) segments of the JSD recordings. In particular, each recording consists of at least three musical segments, while some recordings contain up to 45 musical segments. The average number of musical segments per recording is 9.84. The distribution of the segment durations is displayed in Figure 5b. The mean segment duration is 35 seconds, which is roughly ten seconds longer than for the SALAMI dataset (cf. Figure 2b).

A further analysis shows that the 340 JSD recordings contain 1074 solo sections, which is an average of 1074/340 ≈ 3.16 solos per performance. Each solo, in turn, consists of 2223/1074 ≈ 2.07 choruses on average. One of the longer solos (in terms of choruses) is the saxophone solo in "Juju" (see Figure 3) consisting of six choruses. The longest solo in terms of duration is the saxophone solo in "Impressions"[13] by John Coltrane, lasting thirteen minutes.

3.3 INSTRUMENTATION ANNOTATIONS

Besides the structure annotations, JSD also provides the information on the active instruments for each annotated segment. Typically, in theme sections, all instruments of the ensemble are active. In the solo sections, there is a single solo instrument, which is accompanied only by a subset of the ensemble instruments, often the *rhythm section* (typically piano/guitar, bass, drums). Table 2 provides an overview of all instrument types (along with abbreviations used as instrument identifiers) that occur in at least one of the 340 jazz recordings—either as solo or as accompanying instrument. Furthermore, the table indicates the total number of solos and solo choruses per instrument type. Note that the WJD annotations (see Section 2.2), which are based on the same recordings, comprise solo transcriptions for 456 out of the 1074 solos (corresponding to 33.75% of the solos). In the last two columns of Table 2, we indicated the availability of transcriptions per instrument (in terms of total number and percentage).

Figure 4 shows how the instrumentation of each segment is encoded. In the case that there is no solo instrument (e.g., in a theme segment), the annotation file contains a comma-separated list of identifiers of active instrument types (e.g., "tp,ts,p,b,d"). In the case that there is a solo instrument, the prefix "s_" is used to indicate the solo instrument, and the prefix "b_" to indicate the accompanying band instruments (e.g., "s_tp,b_p,b_b,b_dr" in the first trumpet solo of "Jordu"). In the fourth solo, tenor saxophone, trumpet, and drums share the solo chorus. We reflect this in the annotations by adding the prefix "s_" to each solo instrument "s_ts,s_tp,s_dr,b_p,b_b", i.e., the current annotation schema cannot encode structures smaller than a chorus.

As Table 2 indicates, the majority of the 1074 solos, instruments such as the tenor saxophone (245 solos), the piano (222 solos), the trumpet (170 solos), or the alto saxophone (107 solos), are involved. The dominance of these four solo instruments is also evident when looking at the accumulated duration of solo segments per instrument type (see Figure 6a). Particularly for data-driven machine learning approaches, such class

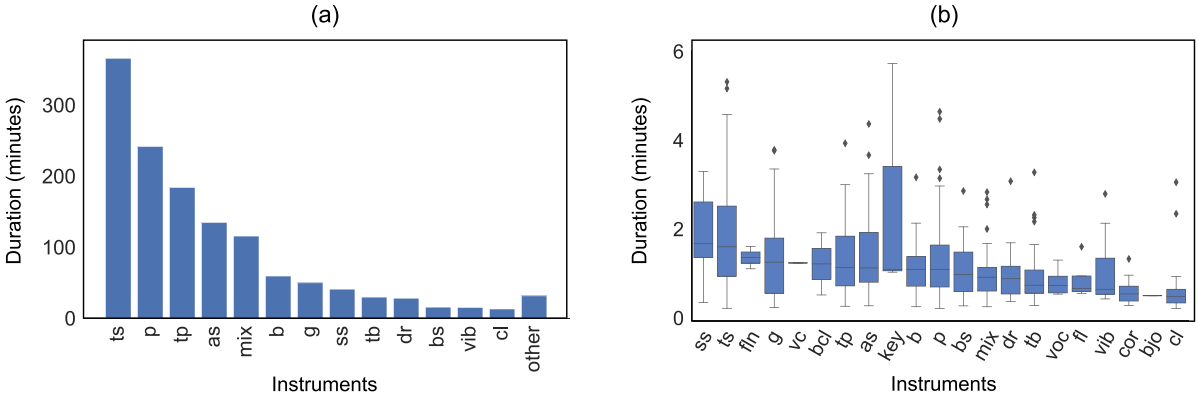

Figure 6: (a) Accumulated duration of all solos (minutes) per instrument. The total duration of all solos is 1325 minutes. **(b)** Statistics on durations of solo sections (seconds) broken down by instrument. The outlier "Impressions" by John Coltrane (containing a 13-minute long saxophone solo) is not shown.

#	Abbr.	Instrument	#Solo	#Chorus	#Trans.	%Trans.
0	cl	Clarinet	23	35	15	65.22
1	bcl	Bass clarinet	4	14	2	50
2	ss	Soprano saxophone	25	74	23	92
3	as	Alto saxophone	107	239	80	74.77
4	ts	Tenor saxophone	245	727	158	64.49
5	bs	Baritone saxophone	21	35	11	52.38
6	tp	Trumpet	170	376	102	60
7	fln	Flugelhorn	2	4	0	0
8	cor	Cornet	18	24	15	83.33
9	tb	Trombone	38	83	26	68.42
10	p	Piano	222	456	6	2.70
11	key	Keyboard	3	8	0	0
12	vib	Vibraphone	15	28	12	80
13	voc	Vocals	8	15	0	0
14	fl	Flute	4	6	0	0
15	g	Guitar	39	88	6	15.38
16	bjo	Banjo	1	1	0	0
17	vc	Violoncello	1	2	0	0
18	b	Bass	61	113	0	0
19	dr	Drums	65	131	0	0
20	perc	Percussion	2	8	0	0
			1074	2467†	456	33.75

Table 2: List of instrument types occurring in JSD. The abbreviations are used as instrument identifiers. The last four columns indicate the number of solos (#Solo), the number of solo choruses (#Chorus), the number of transcribed solos (#Trans.), and the percentage of transcribed solos (%Trans.). †Note that the number of solo choruses is not identical to the number of solo segments from Table 1 (2467 vs. 2223). The former can be higher since there can be multiple soloists in a single solo section. (e.g., drums and bass).

imbalances must be considered when training a model. Finally, Figure 6b shows statistics on the duration of solo sections broken down by instrument type. For example, the solos played by tenor saxophone tend to be longer than the ones played by trumpet.

3.4 WEB-BASED INTERFACES

For the purpose of an easy access and a better understanding of the JSD annotations, we created web-based interfaces using similar technologies as the ones described by Balke et al. (2018) and Rosenzweig et al. (2020). The starting website, as indicated by Figure 1a, provides an overview of the JSD. The website's table contains one row for each of the 340 recordings, showing a recording's JSD title, the interpreter (performer), a three-digit JSD ID, as well as "snapshot" visualization of the structure. It also includes a hyperlink that leads to a separate recording-specific web page, which is shown in Figure 1b for our running example "Jordu." Being based on the *trackswitch.js* player (Werner et al., 2017), this website allows a user to listen to the recording with a time-synchronized visualization of the structure annotation. Above the annotation, the novelty curve of a classical structure analysis method (Foote, 2000) can be seen, which is explained in Section 4. Additionally to the visualized structure, a click sound is added to the audio file at the time position of each boundary. In Section 4.3.2, we show how our web-based interface can also be used for accessing and comparing analysis results obtained from automated approaches.

4 EXAMPLE TASK: BOUNDARY DETECTION

In this section, we illustrate the potential of the JSD for MIR research by considering the task of structure boundary detection as a concrete example scenario. In our experiments, we employ two conceptually different

approaches: a classical novelty-based approach (Foote, 2000) and a more recent data-driven approach using deep learning (Grill and Schluter, 2015; Ullrich et al., 2014). After giving a short summary of these approaches (Section 4.1), we describe our experimental setup including database splits, peak picking, and evaluation metrics (Section 4.2). Then, we discuss our experimental results, giving deeper insights into the JSD and the MIR task at hand, while yielding baselines for future research (Section 4.3).

4.1 BOUNDARY DETECTION APPROACHES

As already mentioned in the introduction, there are many principles a musical structure may be based on. Paulus et al. (2010) and Müller (2015) distinguish three different classes of structure analysis methods used in MIR. First, *repetition-based* methods are used to identify recurring patterns. Second, *homogeneity-based* methods are used to determine passages that are consistent with respect to some musical property (e.g., instrumentation or tempo). Third, *novelty-based* methods are employed to detect transitions between contrasting parts. In the following, we consider two different segmentation approaches that are based on the principle of novelty.

4.1.1 Foote Approach

One classical novelty-detection approach was introduced by Foote (2000) to the field of MIR. In this approach, the music recording is first converted into a sequence of feature vectors, from which one obtains a square self-similarity matrix (SSM) by comparing all elements of the sequence in a pairwise fashion (see Figure 8b for an example). The crucial observation is that the resulting SSM reveals block-like structures in the case that the underlying feature sequence reveals only small variations over the duration of an entire section (e.g., being homogeneous with respect to instrumentation). Often, such a homogeneous segment is followed by another homogeneous segment that stands in contrast to the previous one (e.g., due to different instrumentation). To identify the boundary between two contrasting sections, one correlates a checkerboard-like kernel function along the main diagonal of the SSM. This yields a *novelty function*, the peaks of which indicate instances where significant changes occur in the audio signal.

In our experiments, we use the lower twenty (excluding the first two) *mel-frequency cepstral coefficients* (MFCCs) as underlying feature representation with a feature rate of 10 Hz (Davis and Mermelstein, 1990). These features parametrize the rough shape of the spectral envelope and thus capture properties related to timbre and instrumentation (Aucouturier and Pachet, 2004; Terasawa et al., 2005). To compute the SSM, we use the inner product of normalized feature vectors. To compute the novelty function, we use Hann-weighted kernel functions with two different sizes corresponding to 16 and 32 seconds, respectively. The boundary positions are obtained by applying a suitable peak picking procedure, which is explained in Section 4.2.3. Using MFCCs as the underlying features, the peak positions of a novelty function are good indicators for changes in timbre or instrumentation. In the following, we denote this overall procedure by $\text{Foote}_{\text{short}}$ for the short kernel, and $\text{Foote}_{\text{long}}$ for the long kernel (for a detailed description, see, e.g., Foote, 2000; Müller, 2015).

4.1.2 CNN-Based Approach

As second approach, we use a convolutional neural network (CNN) closely following the architecture proposed by Ullrich et al. (2014). The general idea is to interpret the boundary-detection task as a binary classification problem. The input consists of a short audio snippet (e.g., in form of a time–frequency patch of a spectrogram) with the goal to predict if its center corresponds to a musical boundary (output probability close to 1) or not (output probability close to 0). The structure of the network is shown in Table 3. Further details can be found in Ullrich et al. (2014) and our reference implementation accompanying this paper. The most important design choices are as follows. First, as input patches, we extract standard *mel-spectrograms* with 80 bands at a feature rate of 43 Hz. Similar to the Foote-based approach, we consider two settings denoted as $\text{CNN}_{\text{short}}$ and CNN_{long}, respectively. For $\text{CNN}_{\text{short}}$, the feature rate is 7.17 Hz (43 Hz divided by a sub-sampling factor of 6) and patches consists of 116 time frames, thus corresponding to 16.19 seconds. For CNN_{long}, the feature resolution is 3.58 Hz (43 Hz divided by a sub-sampling factor of 12) and patches consists again of 116 time frames, this time corresponding to 32.38 seconds. With this choice of parameters, the patch sizes roughly correspond to the kernel sizes of $\text{Foote}_{\text{short}}$ and $\text{Foote}_{\text{long}}$, respectively.

In both cases, we use the same network as shown in Table 3. The loss function is the binary cross-entropy. Note that the binary classification problem is highly unbalanced, since there are only few boundary frames

Layer Type	Size	Output Shape
InputLayer	—	(116, 80, 1)
Batch Normalization	—	(116, 80, 1)
Conv2D (ReLU)	8×6	(109, 75, 32)
MaxPooling2D	3×6	(36, 12, 32)
Conv2D (ReLU)	6×3	(31, 10, 64)
Flatten	—	(19,840)
Dropout (50%)	—	(19,840)
Dense (ReLU)	—	(128)
Dropout (50%)	—	(128)
Dense (Sigmoid)	—	(1)

Table 3: Layer structure of the CNN-based approach.

compared to non-boundary frames. To compensate for that, we apply the concept of *target smearing* with a smearing length of 1.5 seconds for CNN_{short} and 6 seconds for CNN_{long}. The target smearing is applied by centering a Gaussian kernel with a width of, for instance, 1.5 seconds at the position of the boundary annotation. This leads to non-zero regions around the annotated boundaries which lead to more (weighted) "positive samples" for the actual learning process. Furthermore, it partly compensates for the annotation ambiguities mentioned in Section 3.2. More details are described by Ullrich et al. (2014) and in the reference implementation. Furthermore, when selecting mini-batches for training, we non-uniformly sample the training set so that at least 20% of the patches belong to the boundary class.

As said before, the output of our trained network is a probability value between 0 and 1. Concatenating the output values of subsequent patches (shifted frame by frame) yields a curve for a full recording similar to the novelty curve as obtained from the Foote approach. To obtain the boundary positions, we again apply a suitable peak picking procedure (see Section 4.2.3).

The result of multiple networks can be averaged to improve the boundary detection performance. This averaging technique is also called *bagging* (Breiman, 1996). For our task, we average the output novelty curves of five networks, which are initialized differently (in a random fashion) and then trained using the same optimization procedure. Peak picking is then applied to the averaged output. The output of five CNN-based novelty functions and their average is shown for our running example "Jordu" in Figure 8c.

4.2 EXPERIMENTAL SETUP

In this section, we describe the general experimental setup including dataset splits, evaluation metrics, and other important aspects.

4.2.1 Dataset Splits

For our experiments, we consider the SALAMI dataset as well as the JSD. We split both datasets into three disjoint subsets for training, validation, and testing. In Table 4, the number of recordings contained in each subset is shown. Additionally, we combine both datasets to form "SALAMI+JSD" with the split obtained by merging the corresponding subsets of the two separate datasets. For SALAMI, we use the same test set as Ullrich et al. (2014). The rest of the recordings are randomly split into the training and validation set using a split ratio as indicated in Table 4. In the JSD split, we enforced that tracks from the same album are not distributed between both test and training set in order to avoid the "album effect". Note that we currently do not account for an "artist effect", i.e., the same artist could occur in the train and test set. For a documentation of the exact splits, we refer to our reference implementation.

Dataset	Training Set	Val. Set	Test Set	Σ
SALAMI (S)	772 (56.8%)	100 (7.4%)	487 (35.8%)	1359
JSD (J)	244 (71.7%)	28 (8.16%)	68 (20.1%)	340
SALAMI+JSD (S+J)	1016 (59.9%)	128 (7.5%)	555 (32.7%)	1699

Table 4: Overview of the splits for the datasets SALAMI, JSD, and SALAMI+JSD. The numbers refer to recordings (with the corresponding percentage given in brackets).

4.2.2 Evaluation Metrics

For our evaluation, we consider standard metrics as described by, e.g., Müller (2015); Raffel et al. (2014). Given a tolerance parameter τ, an estimated boundary is considered correct if it lies in the τ-neighborhood of a reference (ground-truth) boundary. From this, one can derive a threshold-dependent precision (P_τ), recall (R_τ), and F-measure (F_τ). For further details and an implementation of these evaluation measures, we refer to the Python package `mir_eval` (Raffel et al., 2014). Following the conventions of prior work in structure analysis, we consider two different thresholds τ that correspond to 0.5 and 3 seconds, respectively. These measures were also used by Ullrich et al. (2014).

4.2.3 Peak Picking Optimization

Both the Foote and the CNN-based approach yield a novelty curve whose peak positions are used as estimates for musical boundaries. However, peak picking is far from trivial, and the peak picking strategy may have a significant impact on the final F_τ-values. In our CNN-based experiments, we use the same peak picking algorithm as Ullrich et al. (2014) (which is adapted from Böck et al., 2012). For the Foote-based approach, we use the `find_peaks()` implementation from SciPy (Virtanen et al., 2020). For details, we refer to the cited literature. At this point, it is important to note that both strategies depend on a parameter $\delta \in \mathbb{R}$ (added to a local moving average), which enforces that novelty values below δ are not counted as peaks.

In general, the optimal δ depends on the procedure used to compute the novelty curve as well as on the recording itself. In our experiments, we optimize the peak picking parameter δ globally using the validation set. To this end, we compute the value of δ that yields the highest mean F_τ-value averaged over the recordings of the validation set. Note that optimization of this value is done individually for each method (Foote, CNN), dataset (SALAMI, JSD), and τ value (0.5, 3).

4.2.4 Non-Musical Boundaries

Finally, we want to emphasize that the informative value of an evaluation may also depend on other (often hidden) factors. One such issue is the choice of reference boundaries included in the evaluation. For example, as already mentioned in Section 3.1, many

recordings start or end with non-musical segments such as silence, people talking, or applause. Should one include boundaries between non-musical segments and musical segments—boundaries that are relatively easy to detect—in the evaluation? In the case of JSD, the percentage of such boundaries (having 1360 non-musical and 3005 musical boundaries, see Table 1) amounts to roughly 30%. In other words, if we include such boundaries in the evaluation, an approach focusing only on these boundaries may easily achieve a recall of R = 0.33, a precision of P = 1, and a resulting F-measure of F = 0.5. In the subsequent evaluation, we remove such *non-musical boundaries* before and after the music, and only consider *musical boundaries* that separate two musical sections.

4.3 EXPERIMENTAL RESULTS

We now report on the evaluation results obtained for different settings. As also noted by Nieto et al. (2014); Smith et al. (2011), such evaluation results have to be taken with care. Besides algorithmic issues (including novelty computation and peak picking), the choice of evaluation metrics (including their tolerance parameters), the datasets used for training and testing, as well as the way reference boundaries are chosen have a substantial impact on the experimental outcome. Also, the usefulness of the results will crucially depend on the application in mind. Therefore, the quantitative evaluation in Section 4.3.1 should be seen mainly as an illustrative case study and baseline for more detailed analyses. In Section 4.3.2, we introduce some interfaces for a user-centric and more qualitative evaluation of the results.

4.3.1 Quantitative Evaluation

In our experiments, we consider the classical approaches $\text{Foote}_{\text{short}}$ and $\text{Foote}_{\text{long}}$ (see Section 4.1.1) and the CNN-based approaches $\text{CNN}_{\text{short}}$ and CNN_{long} (see Section 4.1.2). For both of the CNN-based settings, in turn, we consider different training scenarios using SALAMI only (S), JSD only (J), and SALAMI and JSD jointly (S+J). We use additional subscript (S, J, or S+J) to denote the underlying training set, e.g., CNN_{J} was trained on the JSD database (see Table 4 for the splits).

First, for a direct comparison of our reimplementation to the original approach (Ullrich et al., 2014), we show in Table 5a the evaluation results for the test set of SALAMI. The F-measure $F_{0.5} = 0.422$ as reported by Ullrich et al. (2014) (corresponding to $\text{Ullrich}_{\text{S, short}}$) is only slightly higher than $F_{0.5} = 0.358$ for $\text{CNN}_{\text{S, short}}$, which is the setting closest to the original approach. We conjecture that the difference in the F-measures is mainly due to a different handling of non-musical boundaries in the evaluation (see Section 4.2.4) and slight variations in the actual implementation. Different settings in the network optimization may be another reason. Our reimplementation is close to what has been considered the state of the art in boundary detection up to the year 2015 (at least, for SALAMI and related datasets).

(a) Evaluation results for SALAMI.

	τ = **0.5 s**			τ = **3.0 s**		
	$\mathbf{P}_{0.5}$	$\mathbf{R}_{0.5}$	$\mathbf{F}_{0.5}$	$\mathbf{P}_{3}$	$\mathbf{R}_{3}$	$\mathbf{F}_{3}$
$\text{Ullrich}_{\text{S, short}}$	0.422	0.490	0.422	—	—	—
$\text{CNN}_{\text{S, short}}$	0.357	0.414	**0.358**	0.419	0.750	0.512
$\text{CNN}_{\text{S, long}}$	0.234	0.223	0.213	0.563	0.672	**0.580**
$\text{CNN}_{\text{J, short}}$	0.231	0.075	0.100	0.432	0.420	0.386
$\text{CNN}_{\text{J, long}}$	0.136	0.049	0.066	0.494	0.233	0.287
$\text{CNN}_{\text{S+J, short}}$	0.347	0.423	0.357	0.484	0.660	0.522
$\text{CNN}_{\text{S+J, long}}$	0.242	0.226	0.221	0.508	0.729	0.571
$\text{Foote}_{\text{short}}$	0.227	0.274	0.223	0.467	0.610	0.477
$\text{Foote}_{\text{long}}$	0.199	0.167	0.169	0.534	0.466	0.463
Baseline (equal)	0.042	0.041	0.043	0.237	0.231	0.244

(b) Evaluation results for JSD.

	τ = **0.5 s**			τ = **3.0 s**		
	$\mathbf{P}_{0.5}$	$\mathbf{R}_{0.5}$	$\mathbf{F}_{0.5}$	$\mathbf{P}_{3}$	$\mathbf{R}_{3}$	$\mathbf{F}_{3}$
$\text{CNN}_{\text{S, short}}$	0.186	0.230	0.189	0.297	0.610	0.382
$\text{CNN}_{\text{S, long}}$	0.122	0.126	0.118	0.423	0.579	0.465
$\text{CNN}_{\text{J, short}}$	0.303	0.125	0.165	0.428	0.556	0.452
$\text{CNN}_{\text{J, long}}$	0.193	0.117	0.139	0.615	0.439	0.482
$\text{CNN}_{\text{S+J, short}}$	0.242	0.269	**0.232**	0.409	0.531	0.428
$\text{CNN}_{\text{S+J, long}}$	0.199	0.169	0.166	0.401	0.682	0.485
$\text{Foote}_{\text{short}}$	0.186	0.247	0.192	0.436	0.601	0.454
$\text{Foote}_{\text{long}}$	0.216	0.185	0.184	0.548	0.505	**0.488**
Baseline (equal)	0.051	0.051	0.051	0.225	0.225	0.225

Table 5: Evaluation results for boundary detection on the test sets of **(a)** SALAMI and **(b)** JSD. The shown precision, recall, and F-measure values are averaged over the respective test set tracks.

From Table 5a, one can observe the following tendencies, all of which may not come as a surprise. First, one obtains higher evaluation values using the larger tolerance $\tau = 3$ seconds compared with ∓ 0.5 seconds. Second, the CNN-based methods generally perform better than the Foote-based methods. Third, for $\tau = 0.5$, the "short" CNN-based approaches yield higher values (e.g., $F_{0.5} = 0.358$ for $\text{CNN}_{\text{S, short}}$) than the "long" ones (e.g., $F_{0.5} = 0.213$ for $\text{CNN}_{\text{S, long}}$). This is different when using the evaluation measures based on $\tau = 3$. For example, the value $F_3 = 0.522$ for $\text{CNN}_{\text{S+J, short}}$ is lower than $F_3 = 0.571$ for $\text{CNN}_{\text{S+J, long}}$. Fourth, using the joint dataset for training (i.e., S+J) slightly degrades the results over using individual datasets (i.e., either S or J).

Most of these tendencies are confirmed when evaluating the same approaches on the test set of JSD (see Table 5b). However, overall, the values of the evaluation metrics lie in lower ranges when comparing to the SALAMI test splits (e.g., $F_{0.5} = 0.358$ for $\text{CNN}_{\text{S, short}}$

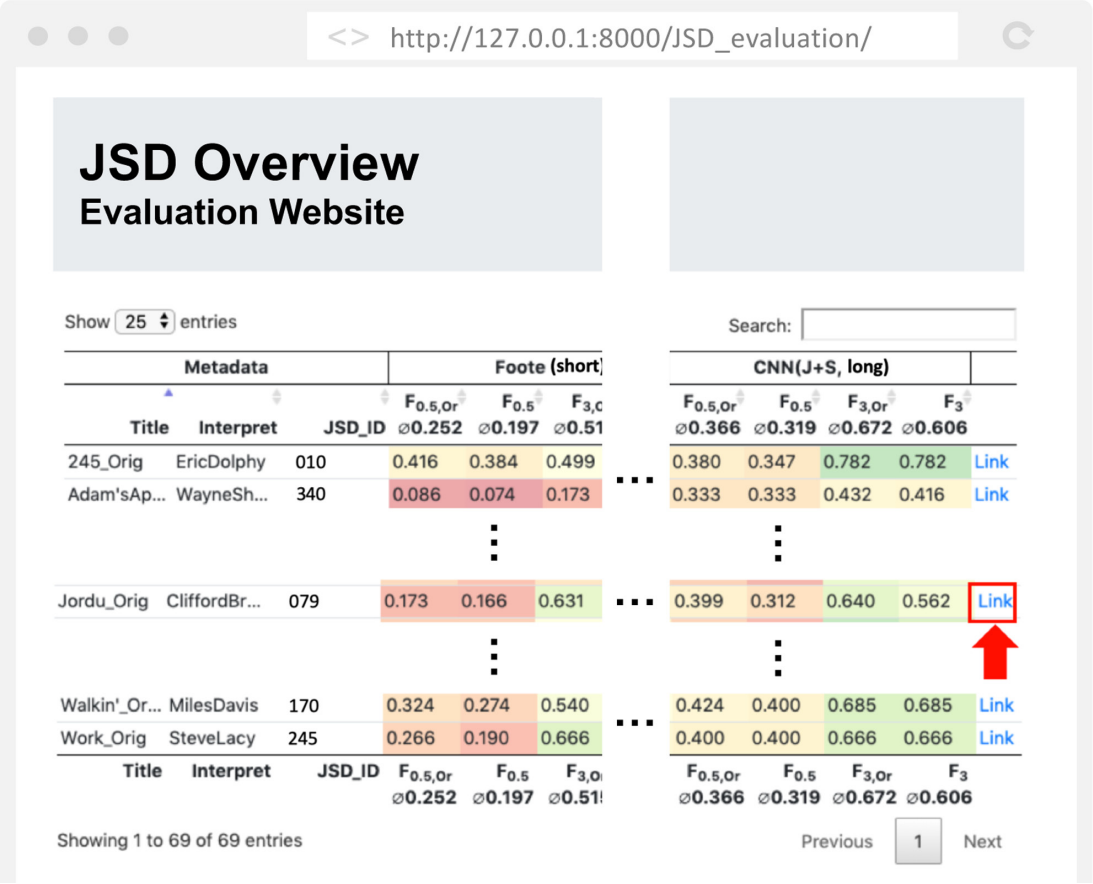

Figure 7: Overview of the evaluation results for all recordings contained in the JSD's test set. The link (red arrow) leads to the details page as depicted in Figure 8.

on the SALAMI test set vs. $F_{0.5}$ = 0.189 on the JSD test set). This indicates that the model trained on SALAMI is not generalizing well on the JSD dataset. Furthermore, a model trained solely on the JSD dataset performs better on the JSD test set but still performs worse than the Foote approach. When training on SALAMI and JSD, the model performs slightly worse than Foote on average, but we do not consider this as a substantial difference in performance.

We close our quantitative evaluation by reporting on a baseline experiment. In this baseline approach, we assume that we know the number of ground-truth segments for each recording. We then simply split each recording into the corresponding number of segments of equal duration and use the resulting boundaries as estimates. The evaluation results are shown in the rows "Baseline (equal)" of Table 5. In particular for the threshold $\tau = 3$ seconds, one obtains already an F-measure that exceeds the value 0.244. This again demonstrates that, while indicating certain tendencies, a quantitative evaluation should always go along with qualitative analyses. This is where interfaces, as introduced in the next section, may help.

4.3.2 Qualitative Evaluation

Using the same web-based technology as discussed in Section 3.4, we also provide interactive interfaces that allow researchers to access, compare, and better understand evaluation results in a qualitative fashion. The website, as indicated in Figure 7, yields an overview of all JSD evaluation results.[14] In particular, the website's table contains a row for each recording of the JSD test set, indicating the recording's metadata (title, interpreter, ID) as well as the F-measures for the Foote-based and CNN-based approaches listed in Table 5b. For a better visual impression, all F-measure values are color-coded, with red shades encoding low values and green shades encoding high values. By a simple click, the rows can be sorted with respect to the table's categories (e.g., alphabetically with respect to the title or in ascending order with respect to any of the F-measures). The F-measures averaged over the full test set are shown in the corresponding table's column header and footer. Furthermore, every row also contains a hyperlink leading to a separate recording-specific web page for more details.

Figure 8a shows such a separate web page for our running example "Jordu" by Clifford Brown. Based on the *trackswitch.js* player (Werner et al., 2017), the interface offers synchronized audio playback and a user menu with the functionality for selecting and switching between different visualizations. As default visualization, the interface shows an overview of the novelty functions obtained from the eight different settings listed in Table 5b. The compact overview of the various novelty

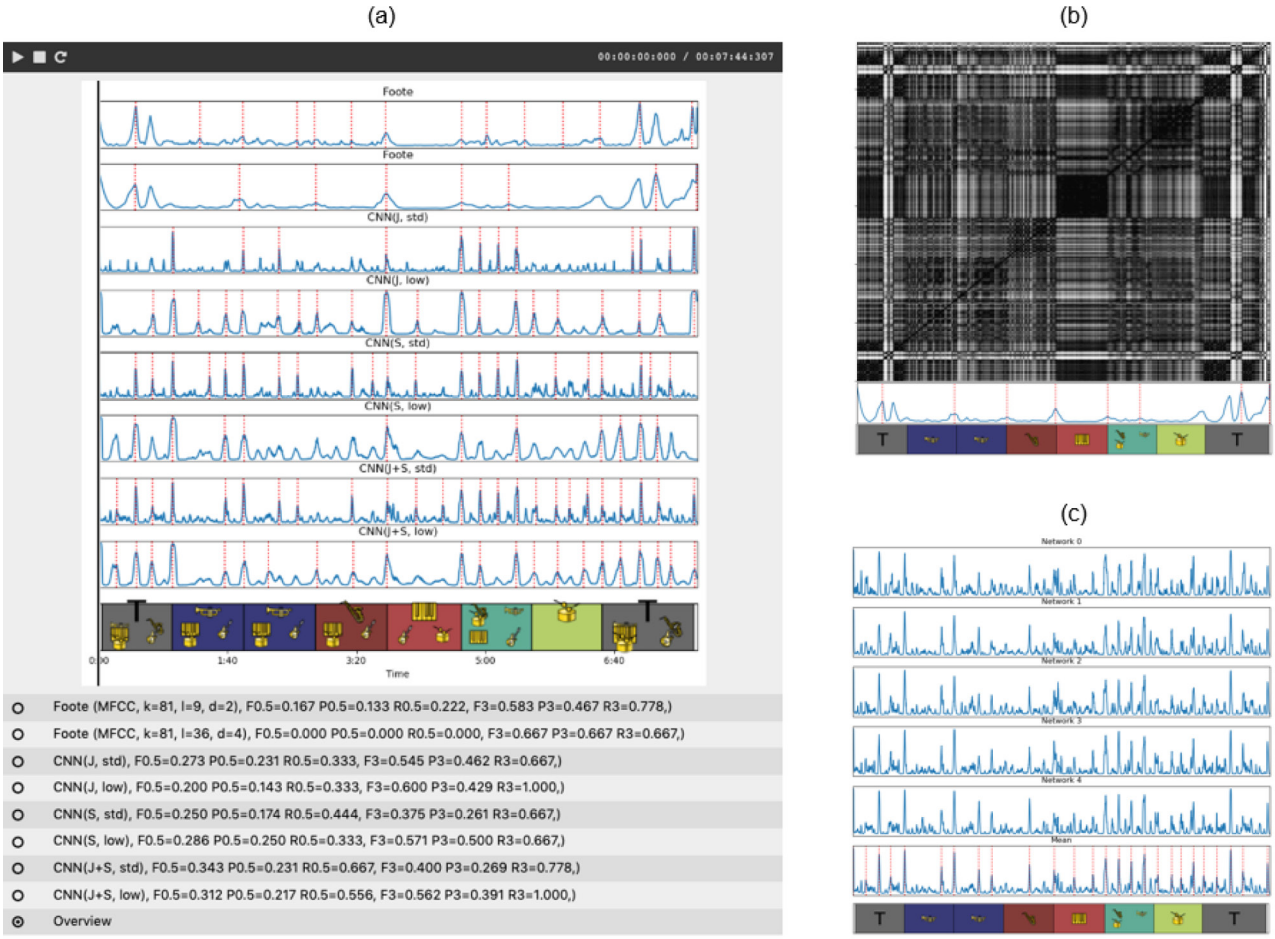

Figure 8: (a) Evaluation web page showing the output of all methods for the running example "Jordu" by Clifford Brown. **(b)** Evaluation results of Foote's method with the input SSM based on MFCCs. **(c)** Evaluation results of a CNN consisting of the novelty curve of five networks and the bagged novelty curve.

functions along with the peak positions (indicated by red vertical lines) allows a researcher to understand better the behavior of the different approaches based on specific examples. Furthermore, by listening synchronously to the underlying jazz recording, one can better understand the challenges of the boundary detection task from an algorithmic, modeling, and musical perspective. To understand an approach in even greater depth, one may switch to a visualization of individual approaches. For example, Figure 8b shows the SSM and novelty curve of a Foote-based approach, while Figure 8c shows the novelty curve of five networks and the resulting bagged novelty curve for a CNN-based approach.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the Jazz Structure Dataset (JSD), which provides structure as well as instrument annotations for 340 jazz recordings. We demonstrated the potential of the dataset for MIR research by considering structure boundary detection as an application task. The JSD is released together with interactive web-based interfaces and Python-based reference implementations. Such tools are necessary not only to gain a better understanding of the data and results, but also to critically question the evaluation measures, the (often hidden) model assumptions, and the task at hand.

Especially during the implementation and evaluation phase of the baselines, the role of the peak picker became evident again. For the sake of being consistent, we left out extensive optimization runs on the peak picker's hyper-parameters, although being aware of the fact that this may lead to sub-optimal results. However, our experiments showed that SciPy's `find_peaks()` implementation produced inferior results compared to the picker used by Ullrich et al. (2014) on the CNN-based approaches. Vice versa, on the Foote approach, the behavior was flipped, i.e., SciPy's picker worked better. This unsatisfying fact leads to the question whether we really evaluate the quality of the model or in the end "overfit" to the typical novelty functions seen in a specific dataset by choosing the better-performing peak picker. Such considerations are of even higher importance as soon as the improvement of model performance is addressed. More advanced deep-learning models involving recurrent, residual, and self-attention components may be considered to better incorporate sequential information, as done with enormous success with the Transformer architecture (Vaswani et al., 2017)

in natural language processing. The use of such large and data-hungry approaches requires an even more critical evaluation to discover their actual benefit over simpler approaches.

Besides the boundary detection task, the JSD and its tools provide the basis for different research questions. First, complementing the SALAMI dataset, it can serve for studying general structure analysis approaches. In particular, repeating harmonic progressions that are superimposed by solo improvisations (as characteristic for jazz music) are a challenging scenario for repetition-based approaches to structure analysis. Furthermore, the instrumentation annotations of the JSD can be used for central MIR tasks related to instrument recognition (Gómez et al., 2018). The JSD allows for studying such tasks with a particular focus on the jazz-specific instrument taxonomy, as indicated by Table 2. In this context, one fundamental task is to detect when certain instrument groups such as percussive instruments (drums, percussion), polyphonic accompaniment instruments (e.g., piano, guitar, vibraphone), or monophonic solo instruments (e.g., trumpet, saxophone, clarinet) are active or not. The specific detection of instruments coming from the same family, such as soprano, alto, and tenor saxophones, is demanding as these instruments share similar sound production mechanisms. A further task is a general instrument-agnostic local estimation of the ensemble size, yielding the number of performing instruments in a structural section such as a chorus.

As for the task of boundary detection, the achieved performance on the JSD shows that Jazz music is a challenging scenario. With the implemented baseline approaches, we assume that a complex model such as a deep neural network can obtain enough structural information to obtain chorus boundaries by simply "looking" at the respective MFCC features. As Jazz music is highly dependent on repeating harmonic patterns (chorus structure), integrating this knowledge into the input representation and/or the model architecture, could improve the results. However we performed baseline experiments and could show that the task by itself is not solved yet. Complementing and extending datasets such as the WJD and the SALAMI dataset, we hope that the JSD contributes another valuable building block to MIR research while highlighting aspects of reproducibility and the need for critical scrutiny of research results.

NOTES

1 Quadromania Jazz Edition, Clifford Brown, Easy Living, CD 2, 2005, Membran Music Ltd.

2 https://github.com/stefan-balke/jsd. Our implementation makes use of the open source packages of Harris et al. (2020), Virtanen et al. (2020), and McFee et al. (2015), among others.

3 https://dunya.compmusic.upf.edu/.

4 http://liederenbank.nl/.

5 https://ismir.net/resources/.

6 The original WJD contains 343 tracks, however, we removed three duplicates (see source code documentation for details).

7 https://archive.org/details/davidwnivenjazz.

8 https://musicbrainz.org/.

9 http://mir.audiolabs.uni-erlangen.de/jazztube.

10 The Blue Note Years – The Best of Wayne Shorter, 1988, Blue Note.

11 Even in the case that there is no silence at all, we added, for the sake of consistency, a silence segment of duration zero.

12 http://www.sonicvisualiser.org/.

13 The Complete 1961 Village Vanguard Recordings, CD 3, 1987, impulse!

14 https://www.audiolabs-erlangen.de/resources/MIR/2022-JSD-BoundaryDetection/.

ACKNOWLEDGEMENTS

We would like to thank Moritz Berendes for helping with structure annotations. Furthermore, many thanks to Jan Schlüter for quickly commenting on questions regarding the CNN. This work was supported by the German Research Foundation (MU 2686/11-1, AB 675/2-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS. Parts of the ideas for this publication were discussed during the Dagstuhl Seminar 16092 on "Computational Music Structure Analysis."

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR CONTRIBUTIONS

Stefan Balke and Julian Reck contributed equally to the annotation process, dataset creation, experimental setup, documentation, and writing. Meinard Müller closely supervised this work and substantially contributed to the writing of this article. Christof Weiß and Jakob Abeßer gave advice on musical and database aspects and helped with compiling this article. All authors have read and agreed to the published version of the manuscript.

AUTHOR AFFILIATIONS

Stefan Balke orcid.org/0000-0003-1306-3548
International Audio Laboratories Erlangen, Germany

Julian Reck orcid.org/0000-0003-0190-299X
International Audio Laboratories Erlangen, Germany

Christof Weiß orcid.org/0000-0003-2143-4679
International Audio Laboratories Erlangen, Germany

Jakob Abeßer orcid.org/0000-0003-4689-7944
Fraunhofer IDMT, Ilmenau, Germany

Meinard Müller orcid.org/0000-0001-6062-7524
International Audio Laboratories Erlangen, Germany

REFERENCES

Abeßer, J., Frieler, K., Cano, E., Pfleiderer, M., and **Zaddach, W.** (2017). Score-informed analysis of tuning, intonation, pitch modulation, and dynamics in jazz solos. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):168–177. DOI: https://doi.org/10.1109/TASLP.2016.2627186

Aucouturier, J.-J. and **Pachet, F.** (2004). Improving timbre similarity: How high's the sky. *Journal of Negative Results in Speech and Audio Sciences*, 1.

Balke, S., Dittmar, C., Abeßer, J., Frieler, K., Pfleiderer, M., and **Müller, M.** (2018). Bridging the gap: Enriching YouTube videos with jazz music annotations. *Frontiers in Digital Humanities*, 5. DOI: https://doi.org/10.3389/fdigh.2018.00001

Balke, S., Dittmar, C., Abeßer, J., and **Müller, M.** (2017). Data-driven solo voice enhancement for jazz music retrieval. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 196–200, New Orleans, Louisiana, USA. DOI: https://doi.org/10.1109/ICASSP.2017.7952145

Bittner, R. M., Salamon, J., Tierney, M., Mauch, M., Cannam, C., and **Bello, J. P.** (2014). MedleyDB: A multitrack dataset for annotation-intensive MIR research. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 155–160, Taipei, Taiwan.

Böck, S., Krebs, F., and **Schedl, M.** (2012). Evaluating the online capabilities of onset detection methods. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 49–54.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140. DOI: https://doi.org/10.1007/BF00058655

Dannenberg, R. B. and **Goto, M.** (2008). Music structure analysis from acoustic signals. In Havelock, D., Kuwano, S., and Vorländer, M., editors, *Handbook of Signal Processing in Acoustics*, volume 1, pages 305–331. Springer, New York, NY, USA. DOI: https://doi.org/10.1007/978-0-387-30441-0_21

Davis, S. B. and **Mermelstein, P.** (1990). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Readings in Speech Recognition*, pages 65–74. DOI: https://doi.org/10.1016/B978-0-08-051584-7.50010-3

Dittmar, C., Pfleiderer, M., Balke, S., and **Müller, M.** (2018). A swingogram representation for tracking micro-rhythmic variation in jazz performances. *Journal of New Music Research*, 47(2):97–113. DOI: https://doi.org/10.1080/09298215.2017.1367405

Eremenko, V., Demirel, E., Bozkurt, B., and **Serra, X.** (2018). Audio-aligned jazz harmony dataset for automatic chord transcription and corpus-based research. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 483–490.

Flexer, A. (2014). On inter-rater agreement in audio music similarity. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 245–250, Taipei, Taiwan.

Foote, J. (2000). Automatic audio segmentation using a measure of audio novelty. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 452–455, New York, NY, USA. DOI: https://doi.org/10.1109/ICME.2000.869637

Frieler, K., Pfleiderer, M., Zaddach, W.-G., and **Abeßer, J.** (2016). Midlevel analysis of monophonic jazz solos: A new approach to the study of improvisation. *Musicae Scientiae*, 20(2):143–162. DOI: https://doi.org/10.1177/1029864916636440

Gómez, J. S., Abeßer, J., and **Cano, E.** (2018). Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning. In Gómez, E., Hu, X., Humphrey, E., and Benetos, E., editors, *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 577–584.

Goto, M. (2006). AIST annotation for the RWC music database. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 359–360.

Goto, M., Hashiguchi, H., Nishimura, T., and **Oka, R.** (2002). RWC music database: Popular, classical and jazz music databases. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 287–288, Paris, France.

Gouyon, F., Dixon, S., Pampalk, E., and **Widmer, G.** (2004). Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of the Audio Engineering Society (AES) International Conference*, London, UK.

Grill, T. and **Schlüter, J.** (2015). Music boundary detection using neural networks on combined features and two-level annotations. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 531–537, Malaga, Spain. DOI: https://doi.org/10.1109/EUSIPCO.2015.7362593

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and **Oliphant, T. E.** (2020). Array programming with NumPy. *Nature*, 585:357–362. DOI: https://doi.org/10.1038/s41586-020-2649-2

Harte, C., Sandler, M. B., Abdallah, S., and **Gómez, E.** (2005). Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 66–71, London, UK.

Kroher, N., Díaz-Báñez, J. M., Mora, J., and **Gómez, E.** (2016). Corpus COFLA: A research corpus for the computational study of flamenco music. *Journal on Computing and Cultural Heritage (JOCCH)*, 9(2):10:1–10:21. DOI: https://doi.org/10.1145/2875428

McFee, B., Kim, J. W., Cartwright, M., Salamon, J., Bittner, R. M., and **Bello, J. P.** (2019). Open-source practices for music signal processing research: Recommendations for transparent, sustainable, and reproducible audio research. *IEEE Signal Processing Magazine*, 36(1):128–137. DOI: https://doi.org/10.1109/MSP.2018.2875349

McFee, B. and **Kinnaird, K.** (2019). Improving structure evaluation through automatic hierarchy expansion. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands.

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and **Nieto, O.** (2015). Librosa: Audio and music signal analysis in Python. In *Proceedings the Python Science Conference*, pages 18–25, Austin, Texas, USA. DOI: https://doi.org/10.25080/Majora-7b98e3ed-003

Müller, M. (2015). *Fundamentals of Music Processing*. Springer Verlag. DOI: https://doi.org/10.1007/978-3-319-21945-5

Nieto, O., Farbood, M., Jehan, T., and **Bello, J. P.** (2014). Perceptual analysis of the F-measure to evaluate section boundaries in music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 265–270, Taipei, Taiwan.

Nieto, O., Mysore, G. J., Wang, C., Smith, J. B. L., Schlüter, J., Grill, T., and **McFee, B.** (2020). Audiobased music structure analysis: Current trends, open challenges, and applications. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 3(1):246–263. DOI: https://doi.org/10.5334/tismir.54

Paulus, J., Müller, M., and **Klapuri, A.** (2010). Audiobased music structure analysis. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 625–636, Utrecht, The Netherlands.

Peeters, G. (2007). Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 35–40, Vienna, Austria.

Pfleiderer, M., Frieler, K., Abeßer, J., Zaddach, W.- G., and **Burkhart, B.** (2017). *Inside the Jazzomat*. Schott Campus, Mainz, Germany.

Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and **Ellis, D. P. W.** (2014). MIR_EVAL: A transparent implementation of common MIR metrics. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 367–372, Taipei, Taiwan.

Rosenzweig, S., Scherbaum, F., Shugliashvili, D., Arifi-Müller, V., and **Müller, M.** (2020). Erkomaishvili Dataset: A curated corpus of traditional Georgian vocal music for computational musicology. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 3(1):31–41. DOI: https://doi.org/10.5334/tismir.44

Serra, X. (2014). Creating research corpora for the computational study of music: The case of the CompMusic project. In *Proceedings of the AES International Conference on Semantic Audio*, London, UK.

Sikora, F. (2019). *Jazz Harmony: Think - Listen - Play - A Practical Approach*. Schott.

Smith, J. B. L., Burgoyne, J. A., Fujinaga, I., Roure, D. D., and **Downie, J. S.** (2011). Design and creation of a large-scale database of structural annotations. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 555–560, Miami, Florida, USA.

Terasawa, H., Slaney, M., and **Berger, J.** (2005). The thirteen colors of timbre. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 323–326. DOI: https://doi.org/10.1109/ASPAA.2005.1540234

Ullrich, K., Schluter, J., and **Grill, T.** (2014). Boundary detection in music structure analysis using convolutional neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 417–422, Taipei, Taiwan.

van Kranenburg, P., de Bruin, M., and **Volk, A.** (2019). Documenting a song culture: The Dutch Song Database as a resource for musicological research. *International Journal on Digital Libraries*, 20(1):13–23. DOI: https://doi.org/10.1007/s00799-017-0228-4

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and **Polosukhin, I.** (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, Long Beach, CA, USA.

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., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and **SciPy 1.0 Contributors.** (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272. DOI: https://doi.org/10.1038/s41592-020-0772-5

Weiß, C., Balke, S., Abeßer, J., and **Müller, M.** (2018). Computational corpus analysis: A case study on jazz solos. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pages 416–423, Paris, France.

Werner, N., Balke, S., Stöter, F.-R., Müller, M., and **Edler, B.** (2017). trackswitch.js: A versatile webbased audio player for presenting scientific results. In *Proceedings of the Web Audio Conference (WAC)*, London, UK.

TO CITE THIS ARTICLE:
Balke, S., Reck, J., Weiß, C., Abeßer, J., and Müller, M. (2022). JSD: A Dataset for Structure Analysis in Jazz Music. *Transactions of the International Society for Music Information Retrieval*, 5(1), 156–172. DOI: https://doi.org/10.5334/tismir.131

Submitted: 06 March 2022 **Accepted:** 28 June 2022 **Published:** 07 November 2022