Mickaël Zehren, Marco Alunno, and Paolo Bientinesi
High-Performance and Automatic Computing Group
Umeå University
Campustorget 5, MIT Building, Second Floor
901 87 Umeå, Sweden
{mzehren, pauldj}@cs.umu.se,
malunno@eafit.edu.co

Automatic Detection of Cue Points for the Emulation of DJ Mixing

Abstract: The automatic identification of cue points is a central task in applications as diverse as music thumbnailing, generation of mash ups, and DJ mixing. Our focus lies in electronic dance music and in a specific kind of cue point, the "switch point," that makes it possible to automatically construct transitions between tracks, mimicking what professional DJs do. We present two approaches for the detection of switch points. One embodies a few general rules we established from interviews with professional DJs, the other models a manually annotated dataset that we curated. Both approaches are based on feature extraction and novelty analysis. From an evaluation conducted on previously unknown tracks, we found that about 90 percent of the points generated can be reliably used in the context of a DJ mix.

In recent years, there has been a growing interest in the automatic generation of DJ mixes, that is, uninterrupted music sequences constructed by partially overlapping music tracks. In a DJ mix, successive tracks are synchronized (i.e., tempo-adjusted and beat-matched), possibly overlapped (for a long or short time), and cross-faded. Intuitively, a "switch point" corresponds to the point in time when the next track in the sequence becomes louder than the current track, which eventually fades out. Since this point affects the listening experience, DJs choose it carefully. In this article we investigate the possibility of identifying switch points automatically. In particular, we focus on electronic dance music (EDM) and seamless transitions between tracks, by far the most common type of transitions in subgenres such as house and techno. As such, this research constitutes an important building block toward the creation of a fully automatic algorithm for DJ mixes.

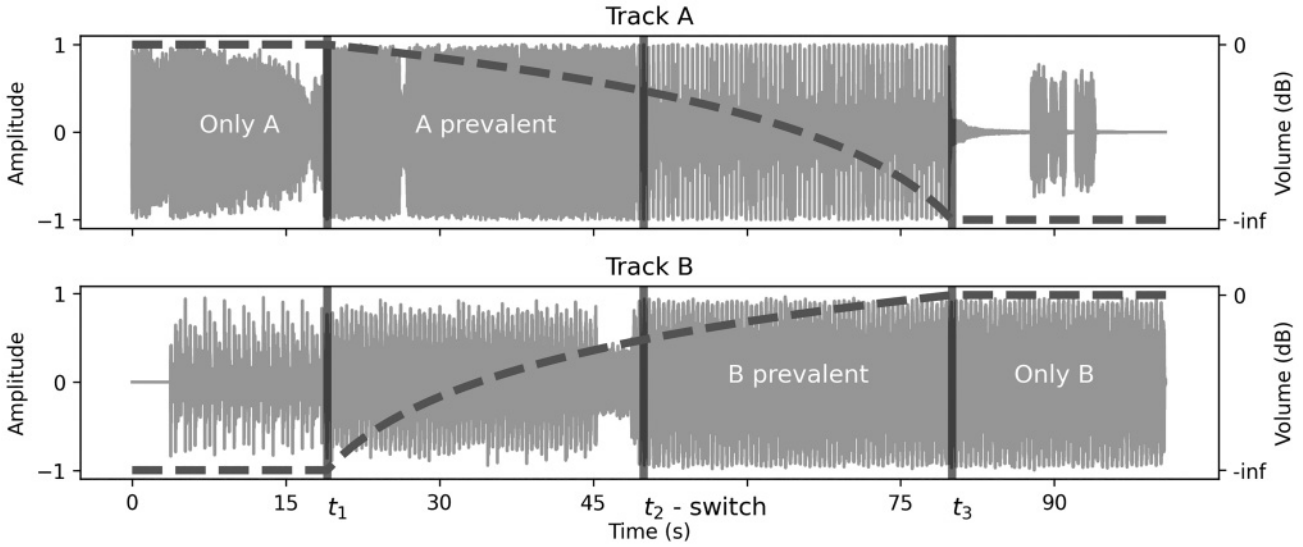
Musicologists, music experts, professional DJs, and the overall attendees of dance clubs generally agree that EDM was born roughly at the end of the 1970s. Since then, the function of a DJ has been to provide people with music to which they can dance uninterruptedly for a long period of time (from one to several hours). Because songs last no more than a few minutes, DJs have to find a way to join the songs together to create a continuous flow of music. Moreover, they have to select which songs to use next, in real time, according to the response of the dancing audience. These operations conflate into two main tasks a DJ has to perform simultaneously: distributing the musical energy appropriately over a long span of time, and making the transitions between consecutive songs pleasant and, frequently, imperceptible. Both tasks require a mastery that only the experience and a deep knowledge of the repertoire can provide. Uninterrupted streams of EDM are also used in contexts where a DJ may not be always available (gyms, fashion stores, private parties, etc.); automated approaches become useful, thus increasing the motivation to investigate them. In fact, the last few years have seen a remarkable concern for developing software that not only generates playlists, but also adds transitions between individual tracks (Bittner et al. 2017; Kim et al. 2017; Vande Veire and De Bie 2018; Schwarz, Schindler, and Spadavecchia 2018).

To better frame the problem, let us consider the scenario of a live performance, where "Track A" is currently played, and "Track B" is selected to be played next. As illustrated in Figure 1, the transition from A to B can be identified by means of three timestamps:

1. t_1, the point in time when B becomes audible but A is still dominant—this point marks the beginning of the transition, also known as the cross-fade section;
2. t_2, the point in time when B becomes louder than A—the "switch point" ("switch-in"

Computer Music Journal, 46:3, pp. 67–82, Fall 2022
doi:10.1162/COMJ_a_00652

Figure 1. Schematic representation of a simple transition from Track A to Track B. Dashed lines represent volume.



and "switch-out" in relation to track B or A, respectively);

3. t_3, the point in time when A becomes inaudible—the end of the transition section.

Depending on the mixing style, t_1 or t_3 might coincide with t_2, leading to the sudden introduction of track B or the sudden removal of track A. In the literature, both t_1 and t_2 are frequently referred to as "cue points." This study focuses only on switch points, and in particular, on their relative position within track B ("switch-in"), arguably the most critical position to create an effective mix.

Our approach to the automatic identification of switch points is inspired by, but not meant to replace, the thought process of professional DJs. To this end, we conducted semistructured interviews with DJs specializing in different genres and styles, and we collected a set of criteria used by DJs to identify viable switch points in a track. To exploit the knowledge gathered in the interviews, we first turned these criteria into high-level rules written in natural language (i.e., English). Then we built an algorithmic implementation of the rules in a formal language (i.e., a programming language), effectively translating the concepts we want to reproduce and automate for the computer. This step is particularly difficult because of the semantic gap between the high-level nature of the rules written in English and the low-level operations performed by a computer. For example, the beat (or pulse) is easy to conceptualize but notoriously difficult to formally explain as a succession of low-level computer operations.

Two approaches can be used to bridge the gap between these two worlds and convert the rules into an algorithm: One is based on machine learning and relies on data to take decisions that are not explicitly coded, whereas the other exploits expert knowledge to handcraft what directions the program should follow. These approaches are not mutually exclusive; in fact, it is common to combine them at different stages of a workflow (e.g., expert-defined features can be used as input data to a machine-learning algorithm). At one end of the spectrum, a fully data-driven approach could be realized as a single deep-learning algorithm working on raw audio. In this case, expert knowledge might still be used to specify the architecture of the network, but most of the algorithm would emerge from automatic optimization of data. At the other end of the spectrum, a fully handcrafted approach could be an algorithm where none of its stages (e.g., feature retrieval, feature selection, or classification) are optimized on data, and results depend only upon concepts established a priori. The former technique

requires a significant amount of training data and is typically not explainable, whereas the latter is more easily explainable but possibly not as effective. In fact, as noted by Geoffroy Peeters (2021, page 3) when discussing handcrafted features, although "those were indeed shallow and explainable at the start, they tended to be deep, data-driven and unexplainable over time, already before the reign of deep learning."

To build our workflow, we looked for a combination of these two extremes and developed two different approaches: one approach based on statistics, called STAT, and one based on expert knowledge, called EXPERT. The former relies on a statistical method—linear discriminant analysis (LDA)—used in music structure analysis (MSA) and illustrated in the work of McFee and Ellis (2014a), whereas the latter capitalizes on expert knowledge. In both approaches, the estimation of the features is hybrid, as the state of the art suggests, with data-driven techniques used to retrieve features for tasks such as beat estimation and drum transcription, and knowledge-driven procedures used for the extraction of features related to timbre and pitch. Therefore, in both workflows, low-level features retrieved from the raw audio with external libraries use data-driven decision making only to some extent. Instead, high-level decisions built on top of low-level features are either statistical or knowledge-driven. Unlike current trends in music information retrieval, neither of the two approaches relies significantly on deep-learning techniques (besides the complex low-level features extracted). This is due to our wish both to keep the workflows reasonably explainable and to limit the size of the available datasets limited.

The rest of this article is organized as follows: First we present a survey of related works, then we describe the rules, the creation of the dataset, and the methodology to identify switch points. Finally, after providing and discussing the results, we draw our conclusions and outline future work.

Related Work

In the last 20 years, several approaches have been proposed to automate different stages of the DJ's workflow. With regard to choosing a location for transition, two of these approaches seem to be more effective: One expresses the compatibility between two tracks, or portions of tracks (Lin et al. 2009; Davies et al. 2014; Gebhardt, Davies, and Seeber 2016; Hirai, Doi, and Morishima 2016; Bittner et al. 2017); the other captures how well-suited a specific position is for a transition, independently of the next track in the DJ mix (Cliff 2000; Davies et al. 2014; Lin et al. 2015; Kim et al. 2017; Bittner et al. 2017; Schwarz, Schindler, and Spadavecchia 2018; Vande Veire and De Bie 2018). In this work, we are only concerned with the latter, which we call "intra-track mixability"; the former, which we call "inter-track mixability," will be incorporated in the future.

The goal of intratrack mixability is to detect switch points in a music track. Those positions, as explained in the following section, have a strong relation to the track's structural boundaries (i.e., the position between two musical segments) and always coincide with some of them. This seems to imply that finding switch points means first performing an MSA (a well-defined task in music information retrieval), because MSA, as well as switch-point detection, is typically solved by looking for novelty, homogeneity, or repetition in a music track. This is not necessarily the case, however: Although all the switch points are structural boundaries, not every structural boundary is a switch point. Indeed, MSA is a complex and large endeavor that does not stop when the macrostructure (i.e., the division in sections) of a piece of music has been established, although this may be a sufficient goal when searching for switch points. A thorough structure analysis also deals with other dimensions of music (i.e., the mesostructure and microstructure—for example, periods, phrases, half phrases, etc.). Therefore, MSA as a whole is not a primary objective of our research, but some aspects thereof are nonetheless useful. For an analysis and an evaluation of different MSA algorithms, we would point the reader to the work by Nieto and Bello (2016).

One of the earliest discussions on identification of switch points is found in a technical report by Dave Cliff (2000), "Hang the DJ," which deals with the automatic creation of DJ mixes. The idea there is to transition between tracks during segments with

no clear pulse (also known as "breakdowns"), then to identify such segments as the portions of a track in which a beat detection algorithm fails to detect the beat. The report does not provide any evaluation of the results.

To the best of our knowledge, the first work about the segmentation of EDM tracks was proposed by Rocha, Bogaards, and Honingh (2013). There, segmentation is applied as a preprocessing step to assess tracks' similarity, a feature that plays an important role in intertrack mixability. In that research, the structural boundaries are identified by using the novelty detection algorithm, developed by Jonathan Foote (2000), on a beat-synchronous representation of a track's timbre. The approach is motivated by the observation that a change in timbre (i.e., the introduction or removal of one or more instruments) is a significant compositional element in EDM and thus displays high novelty at a segment's boundaries. The boundaries detected are then quantized to the closest downbeat. When evaluated on an in-house EDM dataset, this approach yielded good results, although a comparison to other algorithms was not performed.

Segmentation of EDM tracks was also performed by Yadati et al. (2014). In that work, the authors target the detection of "drops" (i.e., points of "emotional release" that follow a part where the song's energy increases, called "build up") by analyzing a list of candidate positions drawn from the macrostructure analysis of the track. The segmentation is performed with a technique proposed by Serra et al. (2014), which is known for being able to identify more than 92 percent of drop positions in the list of candidates. The average distance of 2.5 sec between the ground truth and the generated position is too large for finding switch points that can be used in the context of a DJ mix, however.

The identification of switch points is a task relevant also for the automatic generation of mashups and medleys. For instance, in AutoMashUpper (Davies et al. 2014), the transitions between tracks are constrained to take place at the boundary between phrases detected with Foote's algorithm.

In a more recent study (Bittner et al. 2017), switch points are collected by combining three different methods: First crowdsourcing is used to identify drop locations, then the structure of the track is determined by a repetition-based algorithm (McFee and Ellis 2014b), and finally downbeat locations are retrieved from Echo Nest Analyzer, a music intelligence platform (Jehan 2005). Multiple heuristics are used to prune candidates and select the best switch point for any tuple of tracks, and most of the resulting transitions are rated as satisfactory, with only about 8 percent of the transitions considered "bad." Unfortunately, this approach is no longer viable, as Echo Nest's API is now privately owned.

Another approach proposes to combine Foote's novelty-based algorithm for structure analysis with deep neural networks, but an evaluation of the results is missing (Kim et al. 2017; Foote 2000).

Besides our current study, two other studies (Vande Veire and De Bie 2018; Schwarz, Schindler, and Spadavecchia 2018) proposed rule-based approaches for the identification of switch points. In their work, Vande Veire and De Bie aim for the automatic generation of drum-and-bass DJ mixes. They then go on to explain how to improve the quality of the boundaries returned by Foote's algorithm through rules that encode knowledge of the specific music genre under consideration. This method was positively evaluated, but it was not extended to other musical genres. The context of Schwarz and coworkers' project is "point-of-sale (PoS) automatic mixing in shops." This method aims to improve the structure analysis obtained with the module "IRCAM Summary" by means of heuristics from experts in music branding and the knowledge obtained from a database of 30 tracks manually annotated (Kaiser 2012; Kaiser and Peeters 2013). They only implement a small subset of the expert criteria, however, and the evaluation only proves the usefulness of the method as a starting point for a human annotator.

Table 1 summarizes the literature discussed in this section.

Rule-Based Approach

Our approach to switch point detection is meant to establish a set of general rules that can subsume the relative uniformity of our repertoire. For the specific

Table 1. Summary of Related Work

Work	*Features*	*Boundary detection*
Cliff (2000)	Beats	—
Rocha, Bogaards, and Honingh (2013)	Mel-frequency cepstral coefficients (MFCC)	CK
Yadati et al. (2014)	Pitch class profiles (PCP)	SF
Davies et al. (2014)	Semitone spectrogram	CK
Bittner et al. (2017)	Constant-Q transform (CQT), MFCC	LS
Kim et al. (2017)	CQT	CK
Vande Veire and De Bie (2018)	MFCC, signal energy	CK
Schwarz, Schindler, and Spadavecchia (2018)	MFCC, spectral centroid, spread, skewness, spectral flatness, PCP, and acoustic features	CK and SF, NSMF
EXPERT, STAT	Automatic drum transcription (ADT), harmonic-percussive signals energy, CQT, and PCP	CK

Boundary detection techniques: CK (checkerboard kernel, Foote 2000); SF (structural features, Serra et al. 2014); CK and SF (Kaiser and Peeters 2013); LS (Laplacian segmentation, McFee and Ellis 2014b); NSMF (nonnegative similarity matrix factorization, Kaiser 2012).

goal described in this article, this translates into detecting structural boundaries in music tracks, because we found that positions where some events of structural importance occur are prime candidates to be switch points. Such a task is simplified, to a certain extent, by the modular nature of the music genre we target, as modularity usually provides structural predictability to musical form. Indeed, EDM is modular because it is highly regular both metrically and formally—regular features, as well as several types of ambiguities of EDM tracks are analyzed by Mark Butler (2003). Although exceptions exist, they should not affect a general characterization, as they are vastly outweighed by regularity: EDM is almost exclusively in $\frac{4}{4}$ and is composed of a periodic repetition of phrases (sometimes called "loops") combined into larger periods that constitute EDM's structural building blocks. Furthermore, it is expected that all periods are constituted of two, or a multiple of two, repetitions of a phrase, which is itself composed of two, or a multiple of two, bars. In other words, in most EDM tracks every phrase of four bars can potentially constitute a period and, therefore, may be followed by a new music section.

Formally, we identify three macrosections: the "intro" (the initial section), the "core" (the central section), and the "outro" (the last section). The intro and the outro may be absent. Often, all these sections can be further segmented into portions such as "breakdowns" (where energy drops), and build-ups, sometimes followed by a drop. Knowledge of the process that leads a DJ to choose where to make a transition is condensed into the following rules.

Rule 1: Beat Gridding

A switch point is always located on a strong beat.

This first rule stems from the fact that in EDM all the structural boundaries are always aligned with the beat grid. In fact, they are aligned with a strong beat (beats 1 and 3 of a $\frac{4}{4}$ bar).

Rule 2: Period Alignment

A switch point always occurs on the downbeat at the start of a period.

Due to EDM's structural modularity, switch points always occur on the downbeat of the first bar of a period (i.e., a phrase of four bars).

Rule 3: Novelty

A switch point marks a position of high novelty in rhythmic density, loudness, instrumental components, or harmony.

This rule states that structural events are points in time where new events occur on the music surface, and this novelty is interpreted as a boundary between two consecutive music periods. Common parameters in EDM that convey a sense of novelty are changes in rhythmic density, loudness, instrumental components (i.e., the instruments currently playing), and harmony.

Rule 4: Salience

Switch points are located in the initial portion of a track that precedes the first point of salience.

The novelty rule led to the identification of many candidates. But, from the interviews we conducted, it emerged that DJs tend to look for switch points only in the intro, because doing so makes it possible to highlight the following core section in its entirety and not transition into it after it has already started. Therefore, this final rule limits the search space, that is, switch points are only considered in the portion of a track from the beginning to the first point of "salience." This point in time represents the opening of a section that is prominent or particularly noticeable. For example, the perception of a salient section can be aroused by the presence of a full texture or an intense rhythmic section.

We remind the reader that all these rules are meant for the algorithmic detection of potential switch points, not as a model for the techniques used by human DJs. In live settings, it is in fact possible and likely that DJs take actions to musically interact with the audience, for instance, by transitioning to the next track four or more bars before the switch point of choice, thus creating a sense of anticipation.

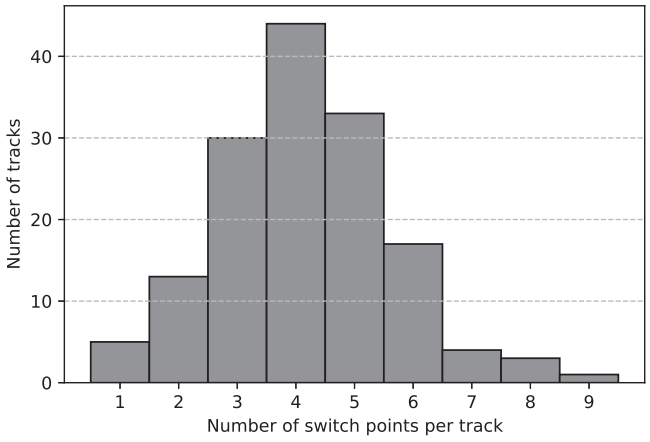
Dataset Creation

Both to steer algorithmic choices and to evaluate different solutions, we curated a dataset of switch points that constitutes our ground truth (Zehren, Alunno, and Bientinesi 2019). Although other datasets were available, none were ultimately suitable for our purpose. For instance, the 1001tracklist dataset (Kim et al. 2020) is of limited use, since it is automatically generated from real DJ mixes and provides only one switch point per track (unless the same track appears in multiple mixes). On the other hand, the UnmixDB does not reflect real expertise, because the switch points are algorithmically generated (Schwarz and Fourer 2021). Unlike 1001tracklist and UnmixDB, our dataset was manually annotated by human experts and contains more than one annotation per track.

The selection of switch points is a subjective process and depends both on the taste and style of the DJ and the peculiarities of the track itself. To create a homogeneous set of annotations, our group of annotators—five experts with a level of qualification ranging from a professional composer to a semiprofessional DJ—were given the set of rules listed in the Rule-Based Approach section as a guideline. The rules were discussed, and the annotators agreed to them. Furthermore, to mitigate the impact of subjectivity, each track was annotated by three (out of the five) experts and their annotations were merged. Due to a large number of possible switch points per track, we also asked the annotators to constrain the annotations from the start of the track up to what they believed to be the beginning of the core section. Such a recommendation shortened the annotation process while keeping those switch points that most valuable and commonly used in the track.

Our dataset consists of 150 tracks of EDM, selected from a period of 30 years (1987–2016), a variety of musical subgenres, and tempi ranging from 99 to 148 bpm. About 60 percent of the tracks come from the digitization of vinyl records. All the tracks were converted to a standard compressed format using FFmpeg (128kpbs, 44.1kHz); the average duration is 7:20 per track, for a combined duration of 18 hours and 20 minutes. As shown in Figure 2, each track contained between one and nine switch points (mean of 4.3). Most of the points are identified by multiple annotators at the same time: all three annotators agreed on 185 of the switch points, two annotators agreed on each of 183 points, with only one annotator identifying each of the remaining 277 points.

Figure 2. Distribution of the number of switch points per track.



Algorithmic Implementation

The two approaches STAT and EXPERT, which we developed to algorithmically express the rules outlined above, are presented in Figure 3. For the most part, they share the same workflow, consisting of five stages: feature extraction, novelty detection, period-offset detection, identification of the DJ's search space, and classification. A step-by-step description for each stage follows.

Feature Extraction

Feature extraction is the first step in the identification of switch points. Since the raw signal of a track cannot be directly interpreted in musical terms, a feature-based representation is constructed instead as represented in Figure 4. To build this representation we decided to focus on two aspects: the granularity of the features computed and which features to use.

First, the granularity of the features is dictated by Rule 1, which states that relevant changes are expected to be synchronous with the beat, in particular with the grid of strong beats. To this end, all features are computed over nonoverlapping strong-beat windows, estimated following Böck, Krebs, and Widmer (2016), so that every value computed effectively represents the characteristics of one half bar. Such a synchronicity of strong beats is common practice, as it helps to smooth out irrelevant finer-grained events in the tracks. In reality, the features are initially computed at a finer granularity (e.g., 100 Hz, depending on the specific feature), then the values are aggregated to achieve strong-beat synchronicity. This is done by by computing either the sum or the root mean square (RMS) of all values located between each consecutive strong beat, as we will see for each specific feature. We also considered using a coarser downbeat synchronicity (one value per bar), but we found that the downbeats could not be estimated as reliably as the strong beats. Indeed, it is known that in music in general and in EDM in particular it may be problematic to tell apart the downbeat from the third beat of a bar (cf. Butler 2003).

Second, to select which features to use, we considered the analysis of Nieto and Bello (2016) that shows how, in music structure analysis, the initial choice of features impacts the accuracy of the algorithm as a whole. Furthermore, in terms of accuracy, it is known that no single feature consistently outperforms the others. These analyses seem to apply also for the identification of switch points, for which many different features are used throughout the literature (as shown in Table 1). Since our Rule 3 states that a switch point must be located at a position of high novelty in rhythmic density, energy, instrumental component, or harmony, we identified features of interest for those categories. Both STAT and EXPERT draw on subsets of these features.

To extract rhythmic features, we used software by Vogl and coworkers (Vogl et al. 2017; Vogl, Widmer, and Knees 2018) to estimate two of the three main components of the drum set, i.e., the bass drum and the hi-hat cymbal. We did not extract the snare drum, as it is sparser and usually less reliably detected (Zehren, Alunno, and Bientinesi 2021, p. 6). We paid special attention to these components because of their function in stressing period boundaries in EDM. For example, the start of the typical $\frac{4}{4}$ EDM steady rhythm with the bass drum on each beat, also known as "four to the floor," is a critically important event for the selection of

Figure 3. Workflow of the STAT and EXPERT approaches.

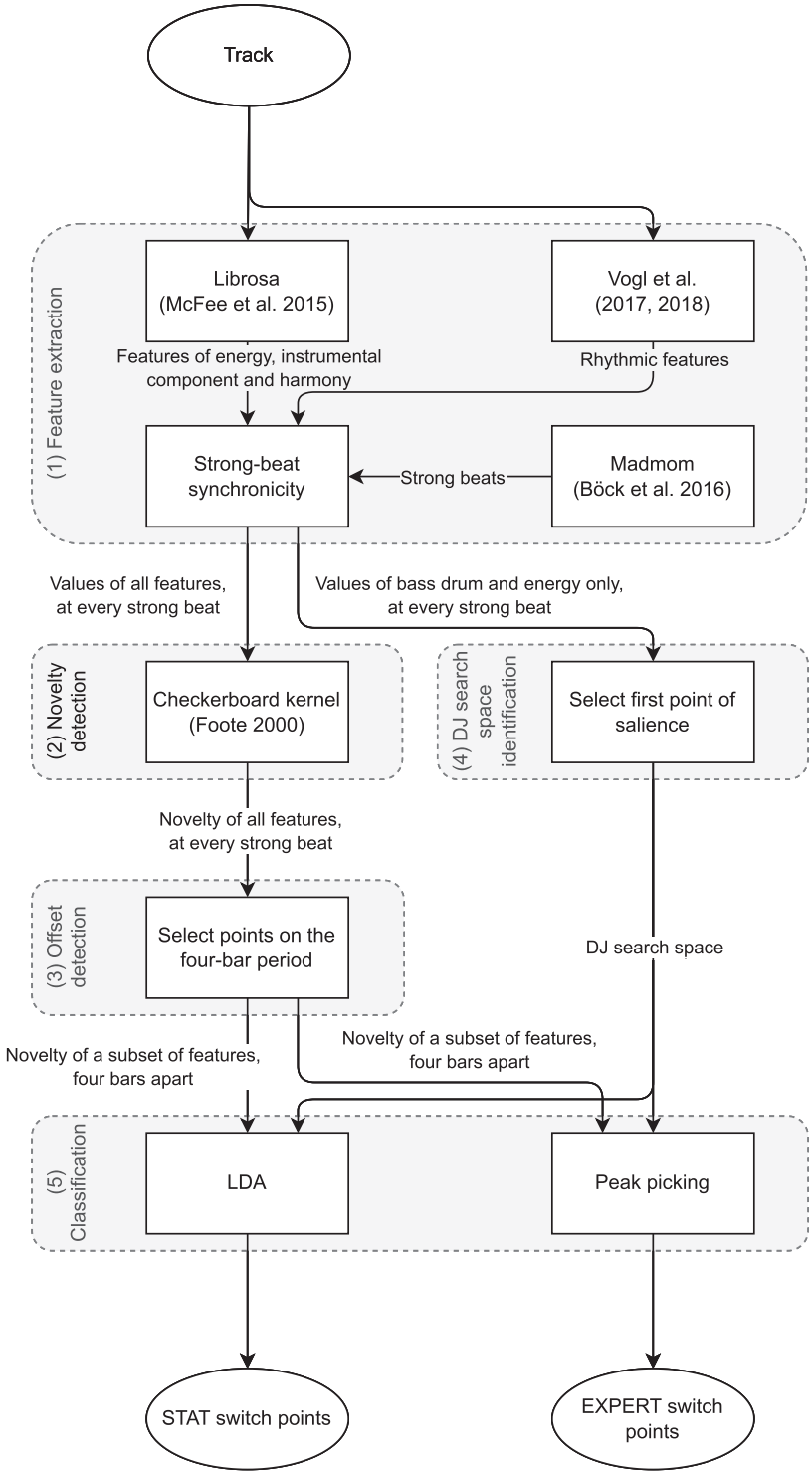
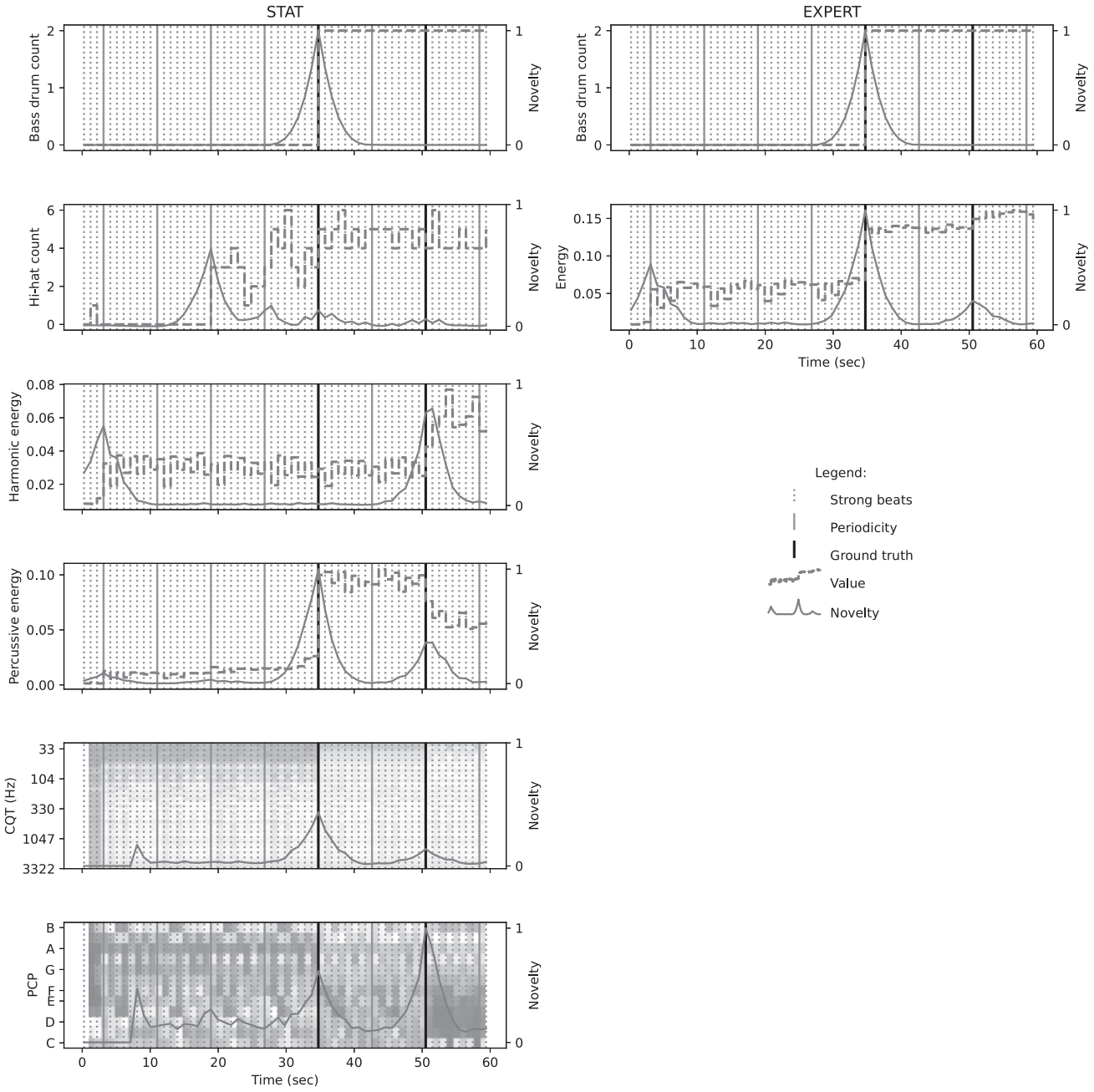


Figure 4. Values of the features (dashed lines and heat maps) and corresponding novelties (solid curves), for STAT and EXPERT (left and right columns, respectively), for the track "Where Love Lives (Classic Mix)" by Alison Limerick. From the novelty curves, the offset is identified on the fourth strong beat (dotted vertical lines), which marks the beginning of a four-bar period (solid vertical lines). One can appreciate how the ground truth annotations (bold vertical lines) fall onto those boundaries of the period that exhibit high novelty.



candidates for switch points. We aggregated each of the two features extracted (bass drum and hi-hat) to the grid of strong beats by counting the number of onsets occurring within two consecutive strong beats; effectively this is a measure of the density of events for those instruments. In STAT, both these

features are used whereas in EXPERT only the bass drum is employed.

With regard to the energy of a signal, we used the amplitude of the audio samples. Then, as suggested by the standard signal processing definition of energy, we computed the RMS magnitude of the sample for each strong-beat window. Notice that this feature is different from loudness (e.g., as computed with the European Broadcast Union's standard EBU R 128), that is, the subjectively perceived sound pressure. For our purposes, loudness is not as useful, as weighting the frequencies according to human perception greatly reduces the influence of the bass frequencies, which are of great interest to DJs. To compute the energy, in STAT we first split the signal into its harmonic and percussive components with the algorithm by Driedger and colleagues (included in the Librosa software, cf. Driedger et al. 2014; McFee et al. 2015). In EXPERT, on the other hand, we only looked at the raw signal energy without harmonic–percussive source separation.

Finally, in STAT, we also considered changes in the instrumental components and harmony. For this reason, we employed both the constant-Q transform (CQT) and pitch class profiles (PCP), as they are known to deliver good results in analysis of musical structure (Nieto and Bello 2016), and they offer quite different representations of the signal. On the one hand, CQT computes a high-resolution spectrogram that is well-suited to timbre identification. On the other hand, PCP does not permit the visualization of timbre, but it does allow us to identify harmony from the intensity of the twelve pitch classes of Western music notation. From the example shown in Figure 4, we see that CQT and PCP novelties have different sensitivities to musical events. The features were extracted with Librosa; each component was aggregated via RMS to the strong-beat synchronicity. The popular mel-frequency cepstral coefficients (MFCCs) could also have been used; they are difficult to interpret in the context of a DJ mix, however, and according to the experimental results of Nieto and Bello (2016) for the related task of MSA, they perform slightly worse than CQT when used as an input for the checkerboard kernel.

Novelty Detection

To comply with Rule 3, we aimed at identifying novelty points in the track. For that, we first normalized each feature x with the formula

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)},$$

where the feature has values $x = (x_1, \ldots, x_n)$ and z_i is the normalized data at the i-th strong beat. Then we used the most common approach to finding novelty points in a signal, making use of the signal's self-similarity matrix (SSM) convolved with a checkerboard kernel. This method is described by Foote (2000) and used by Rocha, Bogaards, and Honingh (2013), Davies et al. (2014), Kim et al. (2017), and Vande Veire and De Bie (2018). We relied on the strong-beat synchronized representations of each feature (as discussed in the Feature Extraction section) to build the associated SSM with the standardized Euclidean distance. Finally, a checkerboard kernel was created with a size set to compute novelty between segments of four bars (hence, a kernel of eight bars) corresponding to the typical smallest length of a period, as required by Rule 2. For most features, we used a convolution with valid padding to avoid any trivial novelty values at the start of a track. But, for the energy features only, we found that zero padding gave good results in the intro section of a track.

This stage yields a "novelty curve" for each of the features, that is, an array containing the novelty value corresponding to each strong beat of the track (as shown in Figure 4).

Offset Detection

To comply with Rule 2, candidates for switch points are required to lie at the boundaries between periods. As discussed in the Rule-Based Approach section, those boundaries occur every four bars, or a multiple of four bars. For this reason, we restricted the search space to strong beats that are four bars apart from one another. Due to anacruses, however, b_1, the first strong beat of a track, is guaranteed

to be neither the first strong beat of a bar nor, consequently, the first strong beat of a period. Since we consider strong beats (two per bar) and four-bar periods, the boundary of the first (complete) period of a track will be $k \in [0, 1, \ldots, 7]$ strong beats after b_1. The exact value of the integer k, also known as "offset" or "phase offset," has to be determined.

The following is our method to identify this offset k. Let g be a weight function and nov_f the function that computes the novelty for a given feature f. Thus, $nov_f(b_j)$ is the novelty value of feature f at the b_j-th strong beat. Moreover, let N be the total number of strong beats of the track. For the sake of simplicity, let us assume N to be divisible by eight. Mathematically, the problem consists in identifying the value of k that maximizes the sum of the weighted novelty for all features at each strong beat every four bars.

This approach is inspired by the work of Vande Veire and De Bie (2018), who consider a similar maximization problem. In their method, however, g filters out all the strong beats for which the novelty is below a certain threshold. We chose instead to set g to the RMS average and to include the contribution of all the strong beats, even if their novelty is low:

$$g(k) := \left(\sum_f \sqrt{\frac{1}{N/8} \sum_j^{N/8} nov_f(b_{k+8j})^2} \right), \quad \text{with} \quad k \in [0, 1, \ldots, 7].$$

Then, offset $= \arg\max_k g(k)$. An example was shown in Figure 4.

In contrast to Rocha, Bogaards, and Honingh (2013), our approach imposes a strict four-bar periodicity, thus achieving higher precision. In fact, whereas Rocha et al. quantize all the novelties within a period to its downbeat, we select only those positions that coincide with that downbeat.

Identification of DJ Search Space

In both approaches, we reduced the portion of the track in which we search for switch point candidates. This aims at simulating the process used by DJs while they are "learning" the next track from the start until they find a point of salience (Rule 4). We refer to this portion of the track as the "DJ search space."

In our context, a point of salience is the beginning of a portion of the track that exhibits high energy. In particular, it is where specific features (i.e., the bass drum count and the raw signal RMS energy) achieve high amplitudes for a sustained period of time. To this end, we set both a minimum threshold for the bass drum to two onsets per bar (equivalent to the sparsest nontrivial drum pattern in EDM) and a minimum threshold for raw energy to the track's median value minus a delta (this latter step following Vande Veire and De Bie 2018). The first point in the track that satisfies these two requirements for the upcoming bar marks the end of the DJ search space.

Classification

Once the novelty for each feature throughout the track has been extracted and the DJ search space has been identified, the EXPERT approach uses a peak-picking procedure. This is a standard method for detecting local maxima in a signal (Rocha, Bogaards, and Honingh 2013; Davies et al. 2014; Kim et al. 2017; Vande Veire and De Bie 2018); in our case, it is used on novelty to return candidate switch points. In general, by peak picking we extract points that are: (1) a local maximum in a small window, (2) above a minimum threshold, and (3) at a minimum distance from the previous peak. In our work, conditions 1 and 3 are always fulfilled, since we restrict all the candidate switch points to the four-bar periodicity. Therefore, we only have to search for points that comply with the second condition, exceeding a threshold. But because fixing a threshold would inevitably be an arbitrary choice, we instead use only the position of the global maximum in the DJ search space for each of the two features considered (i.e., bass drum and raw energy). Consequently, we return at most two candidates per track.

On the other hand, STAT takes into account the novelty of more features than the two used in

EXPERT. Not all features play an equally important role in the selection of switch points, however. In STAT we aim at statistically estimating the weights of the different features through LDA. This method takes as input a set of data points and their associated labels (i.e., "switch point" or "not a switch point"). It then computes a linear transformation, maximizing the separation of the labels in the feature space (i.e., maximizing the distance between the centroids of the two classes and minimizing the variance of each class). This is performed with the Scikit-learn library (Pedregosa et al. 2011). To estimate this linear transformation, we fit the model to all the period boundaries (strong beats four bars apart) in the annotated portion of the training tracks, using the dataset presented in the Dataset Creation section. The points are labeled according to the presence of an annotation in its vicinity provided by a human annotator (less than 0.3 sec away). From these weights, a cumulative score for the strong beats of each period is computed. All the positions within the DJ search space that have a positive score are returned as switch point candidates; if no such position exists, the position with the highest nonpositive score is returned instead.

Experimental Results and Discussion

In this section, we evaluate the overall quality of STAT and EXPERT, both in comparison to other algorithms and with respect to one another, using both objective and subjective criteria. We go on to discuss how some of the design choices we made in STAT and EXPERT are supported by the correlation between feature novelty and switch points.

Objective Evaluation

Each of the two approaches, STAT and EXPERT, yields a list of switch point candidates. Every candidate needs to be evaluated to determine whether or not it is actually a switch point. We now describe the evaluation conducted on the dataset presented earlier in the Dataset Creation section. In particular, the dataset was used to compute the precision of both approaches, that is, the ratio between the number of candidates that were "close enough" (within a 0.3-sec window) to an annotation and the total number of candidates. We computed precision for the test tracks in two different ways: by picturing the tracks as if they all were part of one single large track ("sum precision"), and by considering each track individually and averaging the outcome ("mean precision"). For the sake of completeness, we point out that precision was computed only with respect to the annotated portion of the track and that out of the 150 tracks that constitute our dataset, seven were discarded because the beat-detection algorithm we used (Böck, Krebs, and Widmer 2016) failed.

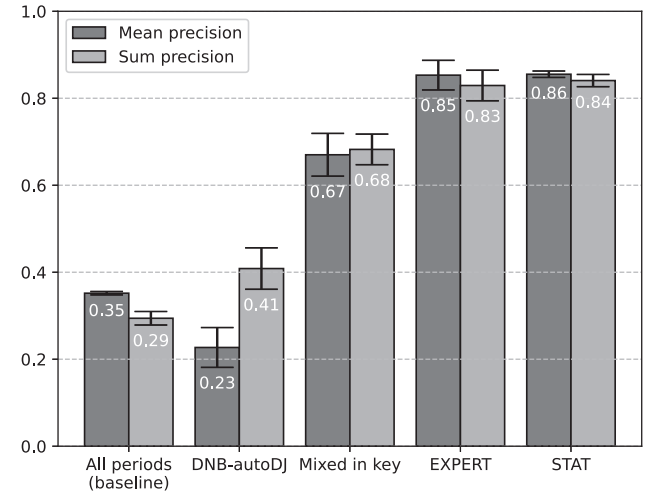
In Figure 5, we report the performance of STAT and EXPERT along with that of three other techniques:

1. "All periods" is a heuristic that selects all the points on the estimated period boundaries (described in the Offset Detection section). For this reason, we use it as a baseline for comparison.
2. "DNB-autoDJ" is a method introduced specifically to find switch points in drum-and-bass tracks (Vande Veire and De Bie 2018). For our experiment, we changed the parameters of this method to accept tracks in the tempo range of our dataset (i.e., adapting original the range 160–190 bpm to the range 99–148 bpm).
3. Mixed in Key is a commercial software application for identification of cue points (https://mixedinkey.com); no algorithmic description of the system is available.

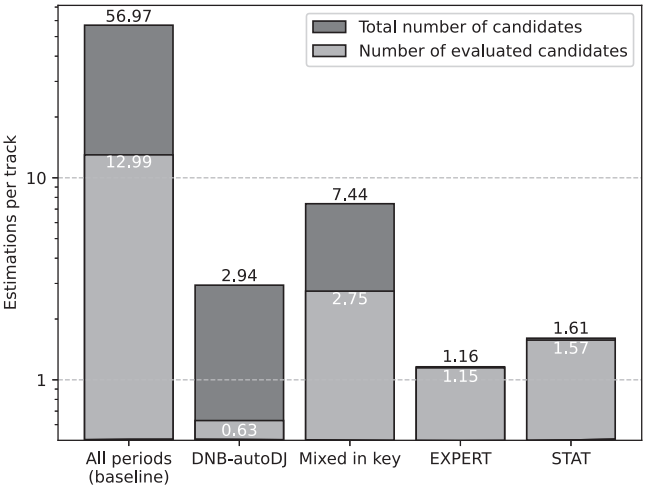
Our evaluation does not include the algorithms described by Bittner et al. (2017) nor those by Schwarz, Schindler, and Spadavecchia (2018), because their code is not publicly available and the available information does not provide enough detail to reproduce the work.

The results are based on a three-split cross-validation strategy: The training for STAT was performed on two subsets resulting from the split, and the testing was done on the third subset. For the other methods, for which no training was required, testing was also done on one subset at a time to keep the results comparable. In all cases, the results

Figure 5. Objective comparison of the algorithms for the generation of switch points. Mean and sum precision for the different approaches on our dataset with error bars for standard deviation (a); total number of candidates and number of evaluated candidates per track, using a logarithmic scale (b).



(a)



(b)

for the three subsets were averaged and displayed together with standard deviation (Figure 5a).

The mean and sum precision for the baseline were 35 percent and 29 percent, respectively. The mean precision of DNB-autoDJ was lower than the baseline; this was due to the small number of candidates that this method returns within the annotated portion of the tracks, leading to many tracks having no candidates at all. The sum precision of this method was better than the baseline, but was still lower than any other algorithms. This suggests that the design of DNB-autoDJ, specific to drum and bass, does not transfer to the genres present in our dataset. The precision achieved by Mixed in Key was 67 percent and 68 percent, whereas that of both EXPERT and STAT was about 85 percent. This indicates that our methods performed noticeably better than the others.

Figure 5b presents the average per track of both the total number of candidates and the number of evaluated candidates (i.e., those within the annotated portion of the track). In contrast to the other methods, the vast majority of candidates generated by EXPERT and STAT fall within the annotated portion. This is due to the fact that the other methods do not take the DJ search space into consideration, as do EXPERT and STAT. Moreover, on average, STAT generates more candidates than EXPERT, while retaining the same precision, thus offering more mixing opportunities.

Subjective Evaluation

Given that the choice of switch points is a subjective task (ultimately a matter of taste and function), and that no dataset can therefore capture the ground truth in its entirety, we also conducted a subjective evaluation to assess the candidates generated by our two approaches on 30 new tracks. (Of the initially chosen tracks, two were discarded because the beat-detection algorithm failed.) This subjective evaluation was carried out by three annotators out of the five involved in the curation of the dataset; these three listened independently to all of the candidates and judged whether or not they were suitable switch points. If they were not, the annotators also had to provide their reasoning.

Figure 6 compares the precision for STAT and EXPERT, based on the number of candidates labeled as switch points by at least one annotator. As was the case for the objective evaluation, STAT and EXPERT achieved almost identical results, around 90 percent. The subjective nature of this task can be appreciated when precision is computed over those candidates that were labeled as switch points by more than one annotator. In fact, when requiring two or all three annotators to agree, the mean precision of STAT

Figure 6. Subjective evaluation of the candidates generated by STAT and EXPERT.

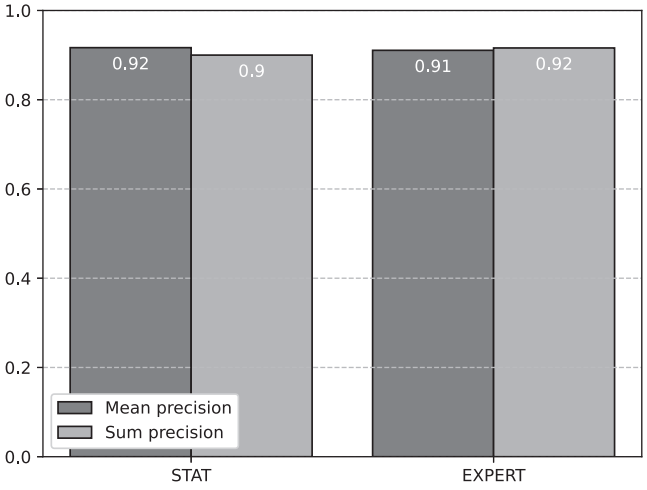
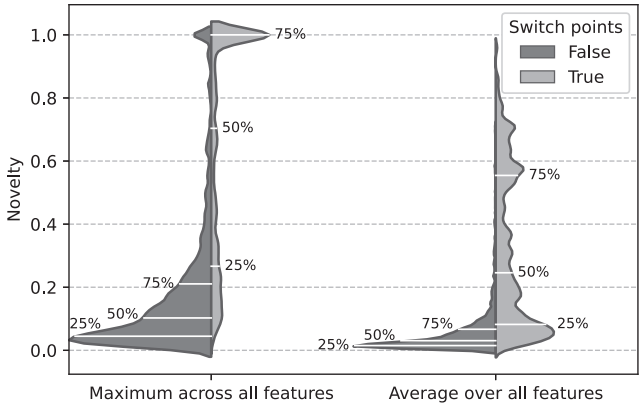


Figure 7. Violin plots representing the maximum novelty across and the average novelty over all features. The 25th, 50th, and 75th percentiles for each area are represented with horizontal lines.



dropped from 92 percent to 76 percent to 65 percent, respectively; the mean precision of EXPERT dropped from 91 percent to 71 percent to 62 percent.

Finally, we analyzed the reason why precision did not reach 100 percent. To this end, we looked at those candidates that no expert considered to be a valid switch point and, more specifically, at the reasoning they provided. We found that the annotators agreed that such candidates were not switch points because, despite being points of novelty, they did not immediately precede a sufficiently noticeable or prominent section. In other words, the segment directly following the candidate was not considered interesting enough to be used after the switch in a DJ mix.

Feature Novelty and Switch Points

We conclude by presenting a study we carried out to determine if there was a correlation between feature novelty and the switch points. For each track of our dataset, we collected all the annotations, as well as the points four bars apart according to the offset detection (as discussed in the Offset Detection section). The annotations constitute the ground truth, and all the other points come from Rule 2 (period alignment). Then, for each point of this set, we constructed a vector containing the novelty value for the seven different features that had been considered (see the Novelty Detection section). Finally, we extracted the maximum and average of each vector. The "violin plots" (a technique to display a probability density) in Figure 7 present the distribution of the maximum and the average novelty: In each plot, the area to the right contains the annotations (i.e., switch points) and the area to the left contains all other points.

By looking at the right-hand portion of the plot of maximum novelty, we observe that the majority of switch points have at least one feature with a novelty above 0.7. This condition is not sufficient to identify the switch points, however, as the left portion of the left-hand plot shows (see the small bump at the top). This observation is supported by the fact that not all features are equally important in the context of a DJ mix. For instance, a change in rhythm and energy is typically more significant than a change in harmony. The fact that high novelty in certain specific features often led to switch points confirms the intuition that stands behind the EXPERT approach, which effectively estimates only a few false positives and thereby attains high precision.

When looking at the plot of average novelty, we see that, if a point has an average above 0.2, it is almost certainly a switch point. In fact, almost no points are found above that threshold on the left-hand area of that plot. There are, however, switch points with an average lower than 0.2.

Obviously, this happens either when one or a few of the features have a high novelty value while the others have little or no novelty at all, or when many or all the features have some but limited novelty. These two observations support the design choices behind STAT, which is less strict than EXPERT and estimates fewer false negatives (i.e., it produces more candidates without reducing precision).

Conclusions

As part of our pursuit of automatic DJ mixing of EDM, we considered the problem of detecting switch points of individual music tracks. These are the points in a track where a DJ would likely transition from one track to the next. By means of the insights collected from professional and semiprofessional DJs, we produced a small set of rules that a point must satisfy to be considered a switch point, and we used these rules to annotate a dataset of 150 EDM tracks. Then we proposed two different algorithmic approaches for the automatic identification of switch points: STAT and EXPERT. The former uses the dataset for estimation, whereas the latter is inspired by common practice among DJs. Our evaluation suggests that these two methods perform similarly to one another and better than all the other approaches we evaluated.

In the future, we aim to exploit the generation of switch points for the automation of DJ mixing. To this end, we will also take into account factors such as compatibility between tracks (or portions of tracks) and the identification of the core portion of a track. The former will influence the mixing style whereas the latter will improve the estimation of the DJ search space.

References

Bittner, R. M., et al. 2017. "Automatic Playlist Sequencing and Transitions." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 472–478.

Böck, S., F. Krebs, and G. Widmer. 2016. "Joint Beat and Downbeat Tracking with Recurrent Neural Networks." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 255–261.

Butler, M. J. 2003. "Unlocking the Groove: Rhythm, Meter, and Musical Design in Electronic Dance Music." PhD dissertation, Indiana University, School of Music, Bloomington.

Cliff, D. 2000. "Hang the DJ: Automatic Sequencing and Seamless Mixing of Dance-Music Tracks." Technical report. HP Labs Technical Report No. HPL-2000-104.

Davies, M. E. P., et al. 2014. "AutoMashUpper: Automatic Creation of Multi-Song Music Mashups." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22(12):1726–1737. 10.1109/TASLP.2014.2347135

Driedger, J., M. Müller, and S. Disch. 2014. "Extending Harmonic–Percussive Separation of Audio." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 611–616.

Foote, J. 2000. "Automatic Audio Segmentation Using a Measure of Audio Novelty." In *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 452–455. 10.1109/ICME.2000.869637.

Gebhardt, R., M. Davies, and B. Seeber. 2016. "Psychoacoustic Approaches for Harmonic Music Mixing." *Applied Sciences* 6(5):Art. 123. 10.3390/app6050123.

Hirai, T., H. Doi, and S. Morishima. 2016. "MusicMixer: Automatic DJ System Considering Beat and Latent Topic Similarity." In Q. Tian et al., eds. *MultiMedia Modeling*. Berlin: Springer, pp. 698–709.

Jehan, T. 2005. "Creating Music by Listening." PhD dissertation, Massachusetts Institute of Technology, Media Arts and Sciences, Cambridge, Massachusetts.

Kaiser, F. 2012. "Music Structure Segmentation." PhD dissertation, Technical University Berlin, Electrical Engineering and Computer Science, Berlin, Germany.

Kaiser, F., and G. Peeters. 2013. "A Simple Fusion Method of State and Sequence Segmentation for Music Structure Discovery." In *Proceedings of the International Conference on Music Information Retrieval*. Available online at archives.ismir.net/ismir2013/paper/000106.pdf. Accessed June 2023.

Kim, A., et al. 2017. "Automatic DJ Mix Generation Using Highlight Detection." In *Extended Abstracts for the Late-Breaking Demo Session of the International Conference on Music Information Retrieval*. Available online at www.researchgate.net/publication/322007163. Accessed June 2023.

Kim, T., et al. 2020. "A Computational Analysis of Real-World DJ Mixes Using Mix-to-Track Subsequence Alignment." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 764–770.

Kronland-Martinet, R., S. Ystad, and M. Aramaki, eds. 2021. *Perception, Representations, Image, Sound, Music*. Berlin, Germany: Springer.
Lin, H.-Y., et al. 2009. "Music Paste: Concatenating Music Clips Based on Chroma and Rhythm Features." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 213–218.
Lin, Y.-T., et al. 2015. "Audio Musical Dice Game: A User-Preference-Aware Medley Generating System." *ACM Transactions on Multimedia Computing, Communications, and Applications* 11(4):1–24.
McFee, B., and D. P. Ellis. 2014a. "Learning to Segment Songs with Ordinal Linear Discriminant Analysis." In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5197–5201. 10.1109/icassp.2014.6854594.
McFee, B., and D. P. W. Ellis. 2014b. "Analyzing Song Structure with Spectral Clustering." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 405–410.
McFee, B., et al. 2015. "Librosa: Audio and Music Signal Analysis in Python." In *Proceedings of the Python in Science Conference*, pp. 18–24. 10.25080/Majora-7b98e3ed-003.
Nieto, O., and J. P. Bello. 2016. "Systematic Exploration of Computational Music Structure Research." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 547–553.
Pedregosa, F., et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12:2825–2830.
Peeters, G. 2021. "The Deep Learning Revolution in MIR: The Pros and Cons, the Needs, and the Challenges." In Kronland-Martinet, Ystad, and Aramaki (2021), pp. 3–30.
Rocha, B., N. Bogaards, and A. Honingh. 2013. "Segmentation and Timbre Similarity in Electronic Dance Music." In *Proceedings of the Sound and Music Computing Conference*, pp. 754–761.
Schwarz, D., and D. Fourer. 2021. "Methods and Datasets for DJ-Mix Reverse Engineering." In Kronland-Martinet, Ystad, and Aramaki (2021), pp. 31–47.
Schwarz, D., D. Schindler, and S. Spadavecchia. 2018. "A Heuristic Algorithm for DJ Cue Point Estimation." In *Proceedings of the Sound and Music Computing Conference*, pp. 259–264.
Serra, J., et al. 2014. "Unsupervised Music Structure Annotation by Time Series Structure Features and Segment Similarity." *IEEE Transactions on Multimedia* 16(5):1229–1240. 10.1109/TMM.2014.2310701
Vande Veire, L., and T. De Bie. 2018. "From Raw Audio to a Seamless Mix: Creating an Automated DJ System for Drum and Bass." *EURASIP Journal on Audio, Speech, and Music Processing* 2018(1):Art. 13. 10.1186/s13636-018-0134-8.
Vogl, R., G. Widmer, and P. Knees. 2018. "Towards Multi-Instrument Drum Transcription." In *Proceedings of the International Conference on Digital Audio Effects*, pp. 57–64.
Vogl, R., et al. 2017. "Drum Transcription via Joint Beat and Drum Modeling Using Convolutional Recurrent Neural Networks." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 150–157.
Yadati, K., et al. 2014. "Detecting Drops in Electronic Dance Music: Content Based Approaches to a Socially Significant Music Event." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 143–148.
Zehren, M., M. Alunno, and P. Bientinesi. 2019. "M-DJCUE: A Manually Annotated Dataset of Cue Points." In *Extended Abstracts for the Late-Breaking Demo Session of the International Conference on Music Information Retrieval*. Available online at archives.ismir.net/ismir2019/latebreaking/000025.pdf. Accessed June 2023.
Zehren, M., M. Alunno, and P. Bientinesi. 2021. "ADTOF: A Large Dataset of Non-Synthetic Music for Automatic Drum Transcription." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 818–824.