Exploring structural sections in EDM DJ-created radio mixes with the help of automatic music descriptors

Vincent Zurita Turk

Supervised by Perfecto Herrera
and Giuseppe Bandiera

Sound & Music Computing
Music Technology Group - Universitat Pompeu Fabra

Abstract

Disc Jockeys are the most likely the ultimate experts at mixing music to “move the crowd”. In this work, structural sections based on “levels of emotional experience” in electronic dance music (EDM) are investigated by analyzing DJ-created radio mixes using automatic music descriptors in order to get audio content information. Mixes are first manually annotated to categorise structural sections and then, their characteristics and patterns between them are investigated. Finally, various evaluations are performed to explore the accuracy with which a system could automatically detect, with the given information, the different sections within randomly chosen independent ten seconds excerpts. Results prove the reliability of the chosen descriptors to achieve a successful classification with moderate number of errors. Moreover, a solution for improving the prediction’s accuracy by taking in account patterns between structural sections is proposed.

Table of contents

Table of Figures

List of tables

List of equations

1. Introduction

Disc Jockeys are the ultimate experts to “move the crowd”. DJs maintain the musical pace and feel in a mix and know that all forms of music have an ingrained pattern of rhythm, tension and release that the dancers naturally follow and expect to hear. When those patterns are broken in a mix, it can seriously throw off the groove. Therefore, they use those patterns to their advantage to keep the dancefloor rocking. They first choose the songs and then decide how to mix them together in a seamless smooth way. They interplay with the crowd and have a whole set of production techniques and methods explicitly intended to build emotional peaks, thus intensifying the listeners’ emotional and bodily responses.

The first factor that we would think to be relevant in a DJ mix is that consecutive tracks need to somehow “work” together. If they “work” together, it means that there will be smooth concatenations and a continuous flow of music where we will probably not realize hops between tracks. In order to achieve that, we would intuitively think that they first need to be similar in tempo and also harmonically, i.e. compatible in tone. But there is more than that …

Background

This section is an introduction to some important concepts on EDM DJ-created mixes. Explaining from basic theory of EDM structure, DJ techniques and how these are correlated to peak emotional experiences. Theories about musical expectancy and tension are explained, and at the end, some state of the art methods to automatically detect structural sections are presented.

1.1 Basic Dance Music Theory

All music is divided into segments called “measures”. Most dance music is written in a “4/4” measure, meaning that there are four quarter notes that comprise a measure. That means that a part of a dance song should be a multiple of four counts of four before changing to another part of the song. The eight counts combine to form evenly divisible multiple segments of 16, 32, 64, 128, or more downbeats, where different elements of the song tend to kick in. No matter the complexity of the songs, 99% of popular dance music is consistent with an “eight count”, which basically is two 4/4 back to back.

Musical structures in EDM are often characterized as highly repetitive, where the basic unit consisting of 2 or 4 bars is repeated and developed throughout the track. The musical elements are introduced, changed or removed after 2, 4, 8, 16 or 32

bars. So that the groove expands in textural density, consisting of rhythmic and melodic structures built layer upon layer (with little variations).

1.2 Basic Beatmixing

Beatmixing is the process of mixing two songs matching their beats in order to make a single mix that flows together. It is achieved by adjusting the speed of the beats of each song so they are the same speed ("beatmatching"), then correctly lining up the beginning of the eight counts by 1 by 1, and running both songs together for a few eight counts during a song's break before bringing in the new song at full volume while simultaneously reducing the volume of your current song. It is usually performed within an even set of eight downbeats; and almost never over someone singing.

Francis Grasso was the first DJ to use headphones so he could hear the incoming track on one turntable and, using a fader knob, blend it into the outgoing track on the other. Until then, DJs were just playing one song after next, without mixing them. At the time, speeds couldn't be adjusted, so that there was no room for error. Without knowing it, Grasso invented the modern art of beatmatching, one of the most basic techniques in a today's DJ repertoire.

1.2.1 Beatmatching

"Beatmatching" is the process of matching tempo or speed of the drumbeats in the currently playing song along with the tempo of the drumbeats of the new song you would like to play, or "beatmix" with the current song.

In order to beatmatch songs with different tempos, the technique of time stretching is used. It refers to altering the length of the song by speeding up or slowing down the reproduction of the song. As a result it will output a slower (in rhythm), lower (in pitch) and longer (in time) version of the song, or a faster, higher and shorter version of the song. The evil side of time stretching is that depending on the EQ and instrumentation of a particular song, the time stretching process may introduce artificial digital artifacts that may sound distorted.

1.2.2 Four Percent Rule

Not all songs will beatmix well. The Ye Olde Chuck "Four Percent Rule" states that you should never attempt to mix songs with a tempo that is more than four percent faster or four percent slower than your current song (Vorobyev, 2012). Any more or less than four percent will usually cause a noticeable change in the "pitch" of a song, meaning the singer's voice or synthesizer parts will sound like they are artificially higher or lower than someone has heard the song before would recognize.

There are now ways to adjust a song's tempo without altering the pitch with hardware and software solutions, but in most situations due to constant song

repetition in the radio and clubs, most people are used to the real tempo and will notice a difference.

There are some exceptions. First, if you are using a turntable, CD player or a digital software with “time stretching”, you can get away up to a 16% tempo gain or reduction with some songs. Time stretching is a complex mathematical algorithm that implies changing the tempo of a song without changing the key nor adding audible artifacts (digital noise). This allows the DJ to speed up or slow down without it being as obvious. Secondly, you can use the pitch control (without time stretching) to change the pitch or key of the music if you are into beatmixing while key matching.

1.2.3 Syncing Beats in Phase

Sometimes a DJ beat-matches two songs perfectly, but the drums hit at slightly different times so instead of hearing a solid kick, we hear two kicks because there is a small gap between them. The solution for the problem is to phase the beats correctly. When two waves peak occur at the same time, there are considered “in phase”. When the peaks occur at different times, they are incoherent or “out of phase”. Phasing can be understood as waves that can either add up or cancel each other. This phenomenon is called superposition and phase cancellation.

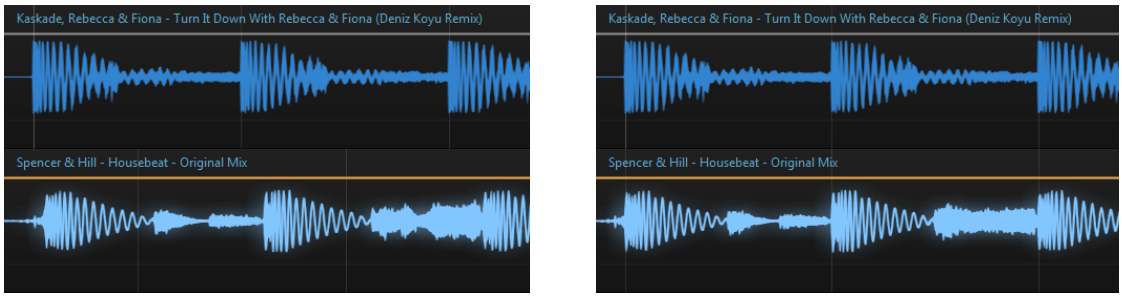

Figure 1: **Left: Different tempos. Right: Songs at the same tempo, beat-matched and in phase**

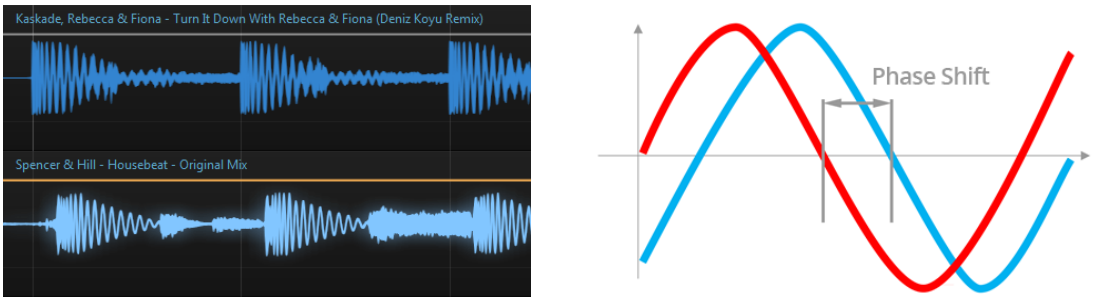

Figure 2: Left: Songs at same tempo, but out of phase. Right: phase shift

1.3 Harmonic Mixing

In 1986, a visionary named Stuart Soroka brought some real musicality into the sessions by introducing the DJ world to the concept of harmonic mixing. It refers to

mix songs that are most often either in the same key, or their keys are relative or in a subdominant or dominant relationship with one another.

1.3.1 Camelot Wheel

The Camelot Wheel, invented by Mark Davis, is a color-coded system that gives information about which keys are compatible, as any musical key shares a special harmonic relationship with certain other keys. Each key is assigned a code number from one to twelve. The wheel is comprised of two concentric circles; major keys are on the outer circle and minor keys are on the inner circle.

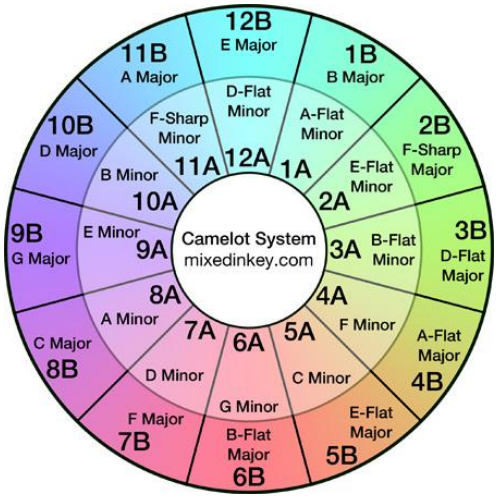

Figure 3: Camelot Wheel invented by Mark Davis, gives information about which keys are compatible.

For any key in the Camelot Wheel, the three immediately adjacent keys are compatible[1]. For example if mixing a song in the key of A minor, which is 8A on the Camelot Wheel; means that you can mix it with any other track in 7A, 8A, 9A or 8B, and feel confident that the result will be a smooth harmonic transition.

A DJ technique called Energy Boost Mixing is used to give some burst of excitement at any point in the set. It happens when mixing into a key that is one or two semitones higher than the current key. To go up one semitone using the Camelot Wheel, you just have to add seven to the number of the current track. To go up two semitones, just add two to the current number. Mixing in the opposite direction of an energy boost will slow down the dance floor, which is a good technique to use towards the end of the set, so that the audience get the idea that the set is about to end.

The exception is the confirmation of the rule, and therefore some other combinations work outside the harmonically compatible group, as songs are not always compatible, but might sound insanely cool when mixed together.

[1] http://www.mixedinkey.com/Book/How-to-Use-Harmonic-Mixing-2

1.4 Energy Level Mixing

In many ways, energy level mixing is a lot like harmonic mixing. Just as the Camelot Wheel assigns a key value to tracks that are harmonically compatible, your energy level values can also inform your choices for your DJ Set.

Good timing is the magic that keeps people dancing. It is necessary to know how to beatmatch, but phrase-matching is actually a more important DJ skill. Understanding phrasing helps to know when to start mixing out of the track you are playing and into a new one, so the mix sounds seamless. In general, it is a good idea to mix out of the chorus so the audience hears the most familiar part of the song, and then continue the momentum into the next track. It is smart to avoid overlaying a beat on the breakdown, as it gives people room to breathe. Mixing intros over outros is the simplest mix that can be done as it is straightforward and easy, but playing the full six minutes of each track might start to get people bored. Therefore, varying levels of intensity are essential in EDM. Most of EDM songs are structured according to this linear pattern (Vorobyev, 2012):

intro > normal intensity > high intensity > breakdown > high intensity > outro.

Some tips mixing from Vorobyev (2012) explained in his book *"Beyond Beatmatching: Take Your DJ Career to the Next Level"*, and also in mixedinkey.com[2] are the followings:

- Don't let the music play non-stop without a break. Let the breakdown play through and don't mix anything new over it; your audience needs a chance to rest.
- Don't mix from a high intensity section to an intro. You can lose momentum with your audience and bring them down too quickly.
- Don't mix from breakdown 1 into breakdown 2. This might work for a chillout room, but not the dance floor.
- Don't mix from a breakdown into the high intensity section of another song. It might sound too abrupt because it doesn't give your listeners enough transition time.
- The best approach is to let a breakdown play all the way through. When the high intensity portion of the song comes back, you can mix into the normal or high intensity portion of the next song. This will keep your energy going.

[2] http://www.mixedinkey.com/Book/Control-the-Energy-Level-of-Your-DJ-Sets

1.5 Correlation between peak emotional experiences and production techniques

Music has the capacity to induce intense emotional and bodily experiences. Producers and DJs are aware of it and have developed techniques explicitly intended to intensify emotional responses of the audience, and they use them to shape musical expectation, tension and intense reactions in EDM (Solberg, 2014). For decades, scholars from different academic disciplines -philosophy, anthropology, psychology, sociology, musicology- have been investigating why music evokes such reactions in us.

The studies on how music can shape emotional peaks and intense experiences are many. A point of departure for many of the scholars concerned with the phenomenology of musical listening has been the humanistic psychologist Maslow's term Peak Experience, defined as "the most wonderful experience of your life: happiest moments, ecstatic moments… from listening to music, or suddenly 'being hit' by a book or a painting, or from some great creative moment" (1968). Gabrielsson (2011) continued this research by collecting interviews and self-reports of Strong Experience with Music, with the purpose of "describing what reactions may occur in particularly strong experience with music, to explore which factors can elicit such reactions, and to consider what consequences the experience may have for the individual".

Scholars have often tended to theorise musical affect as an emotional "response", but using these latter psychological concept as a basis, Solberg (2014) conceptualises this as an emotional "experience", thus underlining the role of the culturally situated listener who is contributing, rather than merely responding, in the shaping of musical and emotional experiences.

These kind of strong emotional experiences are accompanied by physiological responses. Studies within music psychology show that certain musical features and musical passages seem to correlate and further intensify emotional experiences. For instance, increases in heart rate and galvanic skin response, experienced as goosebumps, chills, thrills or shivers, can be triggered when perceiving music. Studies suggest that changes or expansions in, for instance, dynamics, texture, structure and volume are associated with these types of physiological responses (see, for instance, Rickard 2004; Gomez and Danuser 2007; Guhn, Hamm and Zentner 2007; Dibben and Witek 2009; Grewe, Kopiez and Altenmüller 2009).

1.6 Musical expectancy and tension in EDM

Musical tension and expectancy can furthermore shape one's musical experience (Meyer 1956, Huron 2006). In *Emotion and Meaning (1956),* music theorist Meyer develops his theory on how musical expectation and tension shape musical meaning by evoking emotional reactions, thus marking the starting point of studies concerned

with positive and negative emotional responses to musical structures. The correlation between musical expectations and emotional responses has been further investigated and refined by Huron (2006), who coined the ITPRA theory of expectation, which attempts to explain how expectation is a source of musical pleasure.

This theory consists of five physiological and expectation-related emotion response systems -Imagination, Tension, Prediction, Reaction and Appraisal- and each response is related to physiological and psychological changes, and may cause changes in attention, arousal and motor movements. These responses subsequently follow each other, and can be grouped into “pre-outcome responses” and “post-outcome responses”.

The *imagination response* is the ability to *feel* an outcome as if the outcome has already taken place. The aim of the *tension response* is to prepare us for an approaching event by “tailoring arousal and attention to match the level of uncertainty and importance of an impending outcome”. The next level, the *prediction response,* deals with humans’ ability to have an accurate prediction of what is to come; motor responses will come more rapidly and be more accurate if one can manage to predict the outcome, and it will also convey positive emotional responses if one succeeds in predicting the outcome. The *reaction response* establishes if the outcome is pleasant or unpleasant.

The club experience stands out as a culturally dependent phenomenon and experience, some critical remarks can be made about the theory’s heavy reliance on biological processes, thus leading to a misunderstanding as Huron reduces complex musical emotion to a simple “cause and effect” pattern for musicians, producers and composers to use. Nonetheless, the responses Huron describes are applicable to EDM and serve as pertinent explanation model of the musical devices in “build up” and “drop”, which I find related to musical expectancy and biological and cultural processes combined.

Elements related to musical expectancy can be found in newer electronic dance music (EDM) tracks, and particularly in the sections *build up* and drop. These sections are filled with intensifying features, and DJs and producers are highly attuned to what production techniques “move the crowd”, as in causing pleasurable states such as euphoria and ecstasy. However, studies on EDM and the club experience have primarily been concerned with sociological and anthropological aspects of clubbing (see Reynolds 1998; Rietveld 1998; Fikentscher 2000; Jackson 2004). Some contributions regarding analysis of the musical design and aesthetics of EDM exist, with special focus on rhythm, sound, meter and structure (see, for instance, Hawkins 2003, 2008; Butler 2006; Snoman 2009; Zeiner-Henriksen 2010), but generally EDM has not been subject to much music analysis of its production

techniques, and especially not the production techniques in the sections “build up” and “drop”.

1.7 Structural sections & bodily peaks in EDM

As previously indicated, not much of the research on EDM gives a detailed account of the musical features. However, the moments of building intensity are addressed by several of the scholars concerned with the music-theoretical aspect of EDM (Fikentscher 2000; Butler 2006; Montano 2009; Snoman 2009; Zeiner-Henriksen 2010; Garcia 2011), and especially the terms *breakdown* and *build-up* seem quite established within this literature. However, there is less consistency around the terms used to describe the section following the *breakdown* and *build up*. Terms characterizing what the DJ does, such as “peaking the floor*”* (Fikentscher 2000), “peaking the crowd” (Montano 2009) “dropping (down) the bass” (Garcia 2011) and “dropping (down) the beat” (Butler 2006), occur frequently in this literature, but not a specific term defining the section as a whole. I therefore choose to refer to this section as “*the drop*”, after its most prominent and audible feature, namely the reintroduction of the bass and bass drum. This way of producing EDM has expanded post-2010 in newer EDM genres such as dubstep, trap and electro-house, and can also be found in EDM-inspired pop music, but nonetheless the previous literature review gives indications of these ways of building tension as not an entirely new phenomenon.

Dominant sections in EDM are the *breakdown, build-up* and *drop*. Where first the *breakdown section* breaks down the groove and intensity of the track, then the *build-up section* builds it up to a peak which is symbolised by *dropping* the bass and bass drum. Another common effect in newer EDM, is the *drum roll effect* where the rhythmical pattern becomes increasingly divided until the core, starting with quarter notes and finishing in a drum roll right before the bass drops and the drum returns.

A ***breakdown*** is characterized by the track’s texture becoming considerably thinner or even being entirely changed. Several instrument layers are removed, most importantly the bass and the bass drum— on which dancers rely and coordinate themselves. The ***build up*** gives strong indications of a massive musical, but also emotional and bodily peak ahead. The different instrument layers are built up one after one, layer upon layer, the rhythmic structures being constantly compressed, with the clubbers both hearing and sensing many upward and uplifting movements. This continues until the dance floor is bursting with anticipation and seemingly cannot tolerate this any longer. Yet, the DJ plays with them and pushes their boundaries of patience just a little further, before giving them the timely tension-resolving part. The bass and bass drum are **drop**ped down, and the main groove returns with its regular rhythmic and melodic structures, ideally leaving the dance floor more ecstatic than ever.

In EDM, tempo changes are seldom made; the dancers need a steady and predictable framework on which they can rely and improvise within, but by constantly dividing the note value, the DJ or EDM producer gives the illusion and the effect of a tempo increase, affording a greater intensity without the dancer being uncertain about the beat.

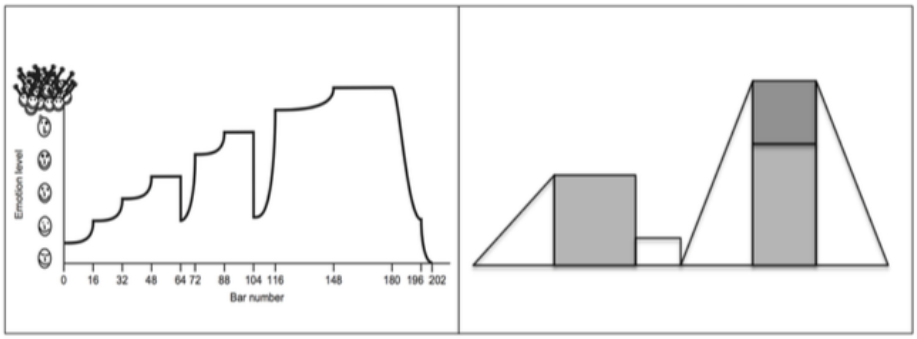

Figure 4: Left: Example of a typical song map including the development of the track and the intensity in the clubbers' emotional level. Right: The drop occurs where the graphs peak, meaning that the affective intensity of the clubbers peaks when they experience being dropped down into the beat (Solberg, 2014).

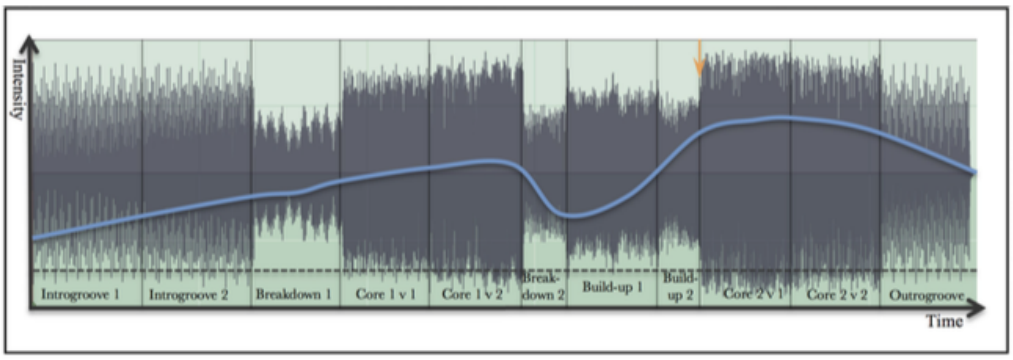

Figure 5: A Schematic model of 'tension' by Cinnamon Chasers (2013) with an overview of the track's sections combined with the track's amplitude graph. The blue line marks T. Solberg's (2014) interpretation of the intensity curve throughout the track, while the orange arrow indicates the drop.

1.8 Listening setting & bodily engagement

Solberg (2015) demonstrates that intense experiences of EDM can occur regardless of the listening setting and bodily engagement, by doing two experiments. The first simulated the club setting where an optical motion capture system tracked and recorded the body movements of a group of people. Each participant wore reflective markers and the group danced together to a continuous DJ mix of four EDM tracks. The second experiment simulated the home listening setting and three physiological responses - electrodermal activity, heart rate level and respiration - were measured on a different group of people. Each participant sat alone while they listened to five different EDM excerpts using headphones. Both studies included the same questionnaire, gathering information about the participants' self-reported pleasure and bodily and affective appraisal of the music, and additionally they described the specific musical characteristics causing affective arousal and desire to move. Some

of the features that reinforced their desire to move, that were related to the large structural and dynamical changes in music were the followings:

- *Extensive use of sounds and effects comprising upward pitch movements*
- *Compression of rhythmical structures*
- *Large changes in frequency spectrum and textural density*
- *Removal and reintroduction of the bass and bass drum*
- *New or changed element occurring after drop.*

1.9 Automatic detection of structural sections

Bello (2016) is one of the many researches that studied the structural segmentation of songs, whose main goal is to automatically identify the large-scale, non-overlapping segments of a given audio signal. The task is divided in two subproblems: the boundary detection and structural grouping. The former identifies the beginning and end times of each music segment within a piece, and the latter labels these segments based on their acoustic similarity.

The variability that characterizes drops in EDM is huge, and therefore, automatic *drop* detection is a challenging task. It has been shown that the importance of this event is reflected in the number of users who leave comments in online audio distribution platforms. Yadati (2014) proposes a method to automatically detect *drops* in EDM recordings, which initial experimental results are promising. It consists of a two-stage approach, that first models the sound characteristics during *drop* events and then incorporates drop-related social comments on the *SoundCloud* platform as weak reference to improve drop detection.

Aljanaki (2014) also employs a multimodal approach combining three sources of data: noisy labels collected through crowdsourcing, timed comments from SoundCloud and audio content analysis. They predict the correct labels from the noisy labels using the majority vote and employ timed comments from *SoundCloud* users to count the occurrence of specific terms near the potential *drop* event, and finally, they conduct an acoustic analysis of the audio excerpts. They obtained the best results when both annotations, metadata and audio were combined, though the differences between them are not significant.

2. Motivation

Knowing that production techniques exist and successfully intensify emotions and arousal of the listener -regardless of listening setting and bodily engagement (Solberg, 2015) we can start understanding DJ decisions (rational and intuitive decisions) and making assumptions about the musical structures in a session. Audience know they will receive their build ups, breakdowns and drops, but the uncertainty and the surprising element lies within the questions of precisely *when* and *how* it will be done, and what characterizes each of the sections.

The aim of this project is to investigate structural sections in EDM radio DJ-created mixes through audio feature extraction and examine features representing timbre, loudness, tempo, etc… Knowing the characteristics of the different sections would allow understanding their differences and similarities, and so, create the possibility of automatically discriminate them by training a classifier system.

Based on previous research on the topic, -I defined four levels- four different structural sections, with which I believe it is possible to describe all levels of “intensity of emotional experience” within a mix. This investigation would prove that my classification is reliable if the system is able to automatically classify them with accurate results.

3. Automatic music analysis techniques: state of the art

In order to extract information from big catalogues of music, tools exist that can automatically extract relevant data from the audio tracks. Research is being done in order to collect information about the music from different perspectives, resulting in four different categories: *music content, music context, user properties* and *user content.*

Music content deals with data directly inferred by the audio signal (such as melody, timbre, loudness, rhythm) while *music context* refers to aspects that are not directly extracted from the signal (such as artist, genre, year of release, label). *User context* and *user properties* refer on data regarding to the user preferences. The former deals with aspects that change frequently (such as mood or social context), while the latter refers to aspects that are considered constant or slowly changing (such as musical preferences).

3.1 Audio Content Analysis

This kind of analysis focuses on extracting useful information directly from the signal itself through some algorithms (or library algorithms). The type of content information can vary very much in relation to the need of the research. There are global descriptors, which analyse the whole musical piece and therefore relate to contextual information, and local descriptors, which analyse smaller portions of the track (window size) and give non-contextual information. Content can be divided in four categories: *timbral information, temporal information, tonal information* and *inferred semantic information.*

- *Timbral information* is the acoustic feature that is neither pitch nor intensity, but the features that make a guitar sound different to a piano even if the play the exact same note. It is directly related to the overall quality and color of the sound.
- *Temporal information* refers to rhythmic aspects, such as tempo or length of measures.
- *Tonal information* is directly linked to the frequency analysis and pitch. It can describe what notes are being played and the tonality of a song.
- *Inferred semantic information* attempt to give a more defined shape to the data collected through machine learning techniques using data from the previous categories. An example of this kind of descriptors may be genre or mood.

Information extracted through these techniques gives information on different levels of abstraction and are classified as low-, mid- and high-level descriptors. These refer (in order of complexity) to acoustical, sensorial, perceptual, structural and expressive concepts.

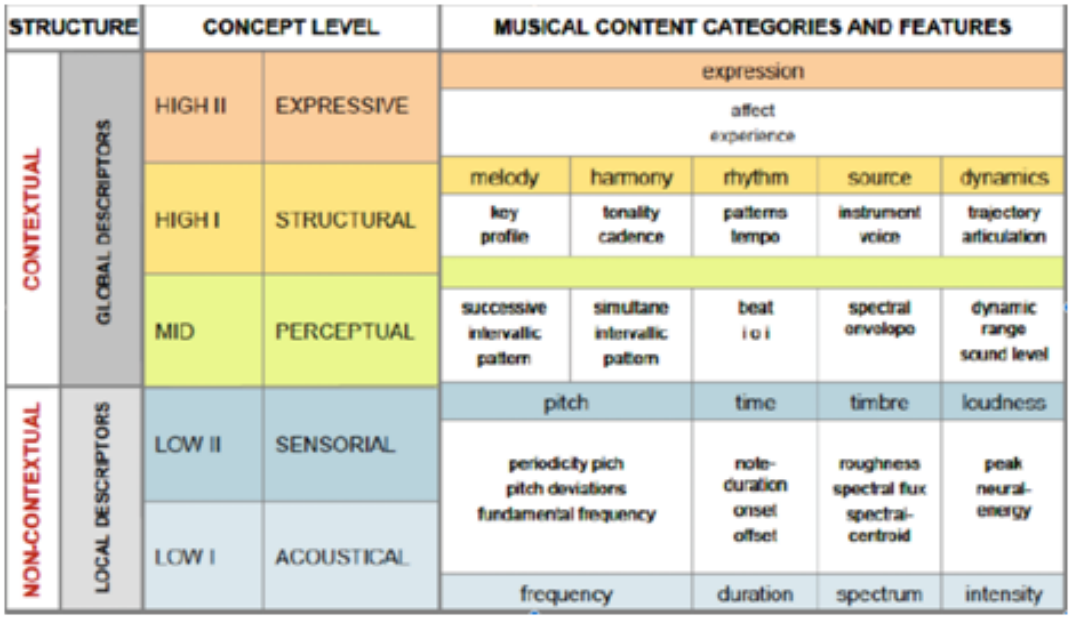

Figure 6: Features classification by levels of abstraction by Lesaffre (2005).

- Low level data (signal-centered): has little sense for the user as it has no musical meaning. Examples of this kind of descriptors are Mel Frequency Cepstral Coefficients (MFFCCs) and Zero Crossing Rate (ZCR).
- Mid level data (object-centered): has musical meaning but that is related to low level music features. This category mainly includes temporal and tonal descriptors.
- High-level (user-centered): is related to inferred semantic information.

Many studies on music similarity computation are done focusing only on low-level and timbral information, because it is proven to bring acceptable results with proper similarity measures -(Schnitzer, 2007). However, more recent studies have shown evidence of advantages using also high-level descriptors (Barrington 2007, Lamere 2007); and the most advanced systems use data from all the categories.

When computing low and mid-level descriptors, the procedure requires the following operations:

- Conversion of the signal from stereo to mono, so that we compute all the descriptors for just one signal

- Down-sampling of the signal to improve the performance while computing the descriptors
- Segmentation of the signal into frames, short segments (usually from 512 to 2048 audio samples). Consecutive frames are not disjoint, as the hop-size determines the hop of samples between the beginning of a frame and the next one, and is normally half or a quarter as big as the frame size
- Computation of Fast Fourier Transform, with an appropriate prior windowing technique, for descriptors that rely on frequency analysis of the signal.

The computation of descriptors is then performed on each frame, and finally a single value for each descriptor is computed by means of specific statistical analysis. Mean, median, variance and covariance are the most used statistical tools for calculating representative global values.

Figure 7: Procedure preliminary to the extraction of audio features. Bandiera (2015)

3.2 Low Level descriptors

3.2.1 MFCCs

Mel-Frequency Cepstral Coefficients (MFCCs) are strongly related to human auditory system, as is a set of critical bandpass filters with overlapping bands that try to replicate auditory filters. The term *critical band* indicates a range of frequencies around a specific one that may not be perceived in a totally independent way if played together to this reference frequency. The Mel bands are based on the mel frequency scale, which is a perceptual scale of pitches judged by listeners to be equal in distance from one another. Mel frequency scale is linear at low frequencies (below 1000 Hz) and logarithmic above. The following formula can be used to convert a frequency *f* in Hertz to *m* Mels:

$$m = 2595\, log_{10}(1 + \frac{f}{1000})$$

Equation 1: conversion of a frequency in Hertz to Mels

Although there is not a standard procedure for computing MFCCs values, steps follow approximately this procedure:

1. Computation of Fourier Transform for each frame

2. Mapping of the powers obtained in step 1 onto mel bands. Then, the logarithm of each power for each mel band is computed, to approximate the cochlea more closely.
3. Computations of the discrete cosine transform to eliminate unnecessary redundancies.
4. Extraction of the MFCCs as amplitudes of the resulting spectrum.

Differences in the procedure can be the shape of the windows used for mapping the spectrum into mel bands, pre-filtering the signal after step 1 or the total number of MFCCs to output.

3.2.2 Bark bands

The Bark scale was proposed by Eberhard Zwicker in 1961, and is a psychoacoustical scale that tries to improve the mel scale, where each "Bark" stands for one critical bandwidth. There are 24 Bark bands, and are described as bands over which masking phenomenon and the shape of cochlea filters are invariant, which is strictly not true. To convert a frequency *f* to *B* Barks we can use the following formula:

$$B = 13a\tan\left(\frac{f}{1315.8}\right) + 3.5a\tan(\frac{f}{7518})$$

Equation 2: conversion of frequency in Hertz to Barks

The published Bark bands (given in Hertz) are[3]:

[0, 100, 200, 300, 400, 510, 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000, 15500]

with corresponding band centers at:

[50, 150, 250, 350, 450, 570, 700, 840, 1000, 1170, 1370, 1600, 1850, 2150, 2500, 2900, 3400, 4000, 4800, 5800, 7000, 8500, 10500, 13500]

Again, width of frequency bands grows slowly below 1000 Hz, while showing an exponential growth at higher frequencies. One advantage between Bark bands values over MFCCs is that they are intuitively more understandable, as they directly represent energy presence, whereas MFCCs values are just abstract numbers not directly related to our perceptual experience.

[3] https://ccrma.stanford.edu/~jos/bbt/Bark_Frequency_Scale.html

3.3.3 Spectral centroid

Spectral centroid refers to the measure that indicated where the “center of mass” of the spectrum is. Perceptually, it has a robust connection with with the impression of “brightness” of a sound, and therefore is used to characterise musical timbre. It is calculated as the weighted mean of the frequencies present in the signal, with their magnitudes as the weights.

3.3.4 Other relevant timbre descriptors

There are a big number of timbre descriptors. Many of them are barely intelligible, but are somehow related to perceptive aspects. Three low-level intelligible descriptors that are often used are:

- Loudness: refers to the perception of the energy of an audio signal. It corresponds to its energy raised to the power of 0.67. The formula was proposed by Stevens (1957), in an attempt of providing a relationship between the magnitude of a physical stimulus and its perceived intensity of strength.
- Dissonance: refers to the measure of perceptual roughness of the sound and is based on the roughness of its spectral peaks, as it depends on the distance between the partials measured in critical bandwidth. Any simultaneous pair of partials of about the same amplitude that is less than a critical bandwidth apart produces roughness associated with the inability of the basilar membrane to separate them clearly.

3.2 Mid level descriptors

Rhythm

Several notations for tempo exist in traditional music, such as BPM (bits per minute), MPM (measures per minute). Others exist that are expressed by semantic notations indicating a range of BPM, such as *presto* (168-200 BPM), *andante* (84-90 BPM) or *allegro* (120-128 BPM).

MIR systems need to use the most accurate notations, therefore semantic annotations are disregarded, and precise notations such as BPM or Onset Rate (OR) are used.

3.2.1 Onset Rate

Onset refers to the beginning of a new musical event, and onset rate is therefore defined as the number of onsets in a time interval. As not all the sounds have long and clear attacks, many difficulties are involved in the process of detecting onsets. For polyphonic music one of the main problems is that it might have simultaneous notes spread over tens of seconds.

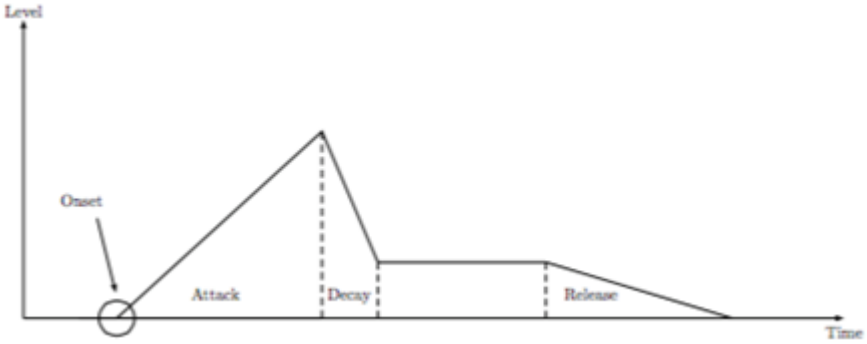

Figure 8: Onset in a sound envelope

Several ways of computing an onset detection have been developed. It may be performed in time domain (when looking for significant changes in the overall energy), or in frequency domain (if looking for event regarding a specific range of frequencies), phase domain or complex domain. Important algorithms for this task are:

- *HFC*, the High Frequency Content detection that looks for significant changes on the highest frequencies. It is useful for percussive events.
- *Spectral Flux*, that decomposes the complete range of frequencies into bins, measures changes in magnitude in each bin, and then sums all the positive changes across all bins.
- *Complex-Domain spectral difference function*, which detects note onsets as a result of significant change in energy in the magnitude spectrum, and/or a deviation from the expected phase values in the phase spectrum, caused by a change in pitch.

3.2.2 BPM

The *beat* or *pulse* is not necessarily the fastest or slowest component of the rhythm but the one that is perceived as the most basic. More precisely, *beat* defines any event or accent, while *pulse* is a repeating series of periodic short-duration stimuli perceived as points in time. It can be defined as those instants when the listeners tap their foot while listening to a track. The algorithms for detecting BPM greatly rely on onset detection functions. The approach is to look for some time-pattern that may explain the distribution of onsets over time, and hence derive the BPM. Usually more than one onset detection function is used to achieve good results.

Since drums tend to drive the rhythmic pulse of a song it makes sense to aim to synchronise most closely with the percussive elements of the music. Fitzgerald (2010) applied median filtering to a spectrogram to separate the percussive and the harmonic components of a signal. This technique for percussive separation is motivated by the observation that percussive features tend to consist of wide band noise across all frequencies and appeal as vertical lines in the spectrogram, while

harmonic components appear as horizontal lines. Median filtering is applied in both the frequency and time direction to create separate percussive-enhanced and harmonic-enhanced spectrograms.

A beat tracking algorithm that achieves very high reliability is *TempTapDegara*, developed by N. Degara et al. (2012). This algorithm models the time between consecutive beat events and exploits both beat and non-beat signal observations, estimating not only beats position but also the expected accuracy. It analyses the input music signal and extracts a beat phase (the location of a beat with respect to the previous beat) and a beat period salience observation signal, and then computes the beat period (regular amount of time between beat events) from these values.

The beat tracking probabilistic model then takes as input parameters the phase observation signal and the beat period estimation, returning the set of beat time estimates. The quality of the beat period salience observation signal is finally assessed and a k-nearest neighbour algorithm is used to measure the reliability of the beat estimates. A *complex spectral difference* method is used for computing the beat phase observation signal that will allow the computation of all the other features. This onset function has shown good behaviour for a vast range of audio signals and has been used in other beat tracking systems (Davies, 2007).

3.2.3 Tonality

Many studies have been done in order to improve techniques for detecting tonality or harmonic content of a song. Most part of the research is not oriented toward the computation of similarity between tracks, but for source separation, that is the task of isolating a specific instrument from the whole.

From a musical point of view, in western music an octave is made of 12 pitches or semitones and seven different notes take place. According to the pitch assigned to each note, we may have different *keys* that are a combination of the *tonic* (central pitch) and the mode. *Major* and *minor* are the most popular modes.

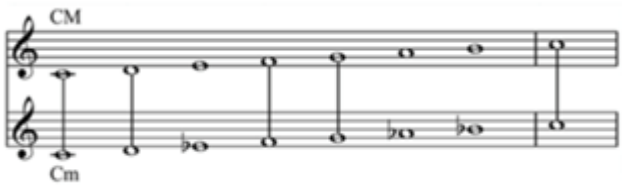

Figure 9: Major and minor modes of C

Harmony refers to the simultaneous combination of notes, called *chords*, and over time, *chord progressions.* One of the most relevant descriptor for extracting information about the tonality is called Harmonic Pitch Content Profile (HPCP), and

is also called chromagram. It recognizes a chord without even precisely detecting what notes are being played, and tonality can also be obtained.

An *HPCP* is a 12*k* size vector that indicated the level of energy for each pitch class profile. If *k* = 1, the *HPCP* represents the intensities of the twelve semitones pitch classes, otherwise it shows a subdivision of these which can be extremely useful when dealing with non-chromatic scales, which are very popular in eastern music. Tonality features can be extracted on different temporal scales, such as *instantaneous* which refer to a single frame (low level of abstraction), and *global*, which are features related to a wider audio segment or the whole song, and represent a higher level of abstraction.

The general approach for computing *HPCP* can be summarized as follows:

- The first step is usually done to decrease the computational cost of HPCP without affecting its output. Therefore, a pre-processing of the audio signal can be performed, and a transient detection algorithm can be used in order to remove noise.
- Then, once the signal is segmented into frames and a proper windowing function is applied, Fast Fourier Transform (FFT) is computed to get the frequency spectrum.
- Subsequently, frequencies corresponding to local maxima are found applying a peak-picking algorithm. Usually, they only run on an interval between 100 Hz and 5000 Hz, as it is the most predominant range, as outside it much noise is added due to percussion and instrumental noise. [12]
- Finally, *HPCP* is computed. Many approaches have been developed based on Fujishima's [13] pitch content profile algorithm. First, a mapping of each frequency bin of the FFT to a pitch class is needed (Gómez, 2006), and then, amplitudes inside each region are summed up and divided by the number of bin inside that region. So that bins referring to the same note, but in a different octave, are collapsed in a single bin for that note indicating the overall energy of it in the frame. The *HPCP* value of the *n*-th *HPCP* bin is calculated as:

$$HPCP(n) = \sum_{i=1}^{nPeaks} w(\mathrm{n}, f_i) a_i^2$$

Equation 3: Calculating the HPCP value of the n-th HPCP bin

Where a_i are the magnitude and f_i the frequency of the *i*th peak, *nPeaks* is the number of spectral peaks considered, and $w(\mathrm{n}, f_i)$ is the weight of the frequency bin f_i when considering the HPCP bin *n.*

HPCP values are usually normalized in order to store the relative importance of the *n*th *HPCP* bin:

$$HPCP_{normalized}(n) = \frac{HPCP(n)}{Max_n(HPCP(n))}$$

Equation 4: HPCP normalized values

Once the *HPCP* is computed, tonality and chord estimation can be computed by correlating the values obtained and a matrix of empirically computed *HPCP* profiles corresponding to different keys or chords.

Pitch salience was designed as quick measure of tone sensation. Pitch salience is given by the ratio of the highest autocorrelation value of the spectrum of the non-shifted autocorrelation value. Unpitched sounds (non-musical sound effects) and pure tones have an average pitch salience value close to 0 whereas sounds containing several harmonics in the spectrum tend to have a higher value. This algorithm may give better results when used with low sampling rates (i.e. 8000) as the information in the bands musically meaningful will have more relevance.

3.3 High level descriptors

The motivation behind high-level descriptors is to translate low and mid-level descriptors into a semantic explanation, through machine learning or statistical analysis. Research in MIR has been done for classifying music genres, artist identification and mood detection. Music genre classification is performed on a machine-learning basis: a classification algorithm is trained with low-level data, such as timbre or rhythm descriptors. The quality of the system will depend not only on the quality of the low-level data, but also on the variety of music in the training set. If the system is trained with *Pop* music it will perform badly on *Classical* music for instance. Furthermore, the process of labelling songs can have also some difficulties, as labels can be misunderstood or not very clear.

An interesting task targeted by the MIR community is the automatic segmentation of audio, where songs are split into their main parts (such as intro, refrain or verse). Approaches are generally based on performing self-similarity in order to locate points of significant change measured on the basis of low-level descriptors regarding rhythm, melody or timbre.

3.3.1 Danceability

Developed by Streich and Herrera (2005), is a high level semantic descriptor that gives information about how danceable a section is, derived from the detrended

fluctuation analysis (DFA) exponent that was first proposed by Peng et al. and further Jenning et al. (2003) used it for music classification.

4. Methodology

In order to explore structural sections in EDM, to be able to inform about their characteristics and understand meaningful information to define them, I first created a collection of EDM radio DJ-created mixes, which was first analysed and annotated manually. The pre-process of analyzing the tracks and reading about already existing research on structural sections in EDM and levels of emotional experience, helped me to decide how to create my taxonomy with which I would annotate the mixes.

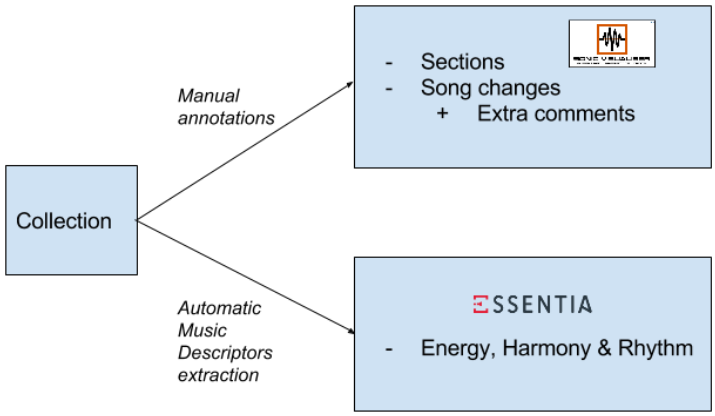

Figure 10: Basic scheme about the collection analysis

Further, audio content was extracted with the help of automatic music descriptors, in order to provide information about the energy, harmony and rhythm of the mixes. Finally, an evaluation was performed to asses the quality of the used music descriptors to accurately classify and discriminate between structural sections. The methodology can be viewed as an empirical musicological approach aimed to allow classifying structural sections in EDM DJ-created mixes.

4.1 Collection

First of all, first things first, I needed to create a collection of EDM radio DJ mixes to work with and to be used as the data-to-be-analysed to perform this study. Many of the DJ mixes that I found on the internet and music platforms were recorded from live shows in nightclubs or festivals where the presence of an audience could be found. As previously discussed, DJs interact with the audience giving them the power to take part on their decisions on what songs to play next or on how to approach the following minutes in the session. Therefore, I needed somehow to avoid the presence of these audiences, in order to avoid third parties affecting the decisions of the DJs. So, I thought about radio DJ sessions, where there is no

audience (only the sound engineer and maybe some presenters and radio station workers). In this context I could analyse how a DJ structures a session by his own means and thoughts, just by getting information purely about the audio itself. If there was an audience, I would have had to put some cameras in the nightclub or festival and analyse the crowd behaviour in respect to the music, and vice versa; and so, study the DJ decisions as a consequence of the audience behaviour, which would have lead to another kind of project.

Therefore, I used as the source data for this investigation, the British Broadcasting Corporationś Essential Mix radio program, considered one of the most reputable and influential radio programs in the world. Broadcast since 1993, the Essential Mix showcases exceptional DJs of various genres of electronic dance music (EDM). Presented by Pete Tong, most part of the studio DJ mixes are two hours long, and some of the live DJ mixes are one hour long.

Essential mixes present many pros for this project, such as:

- Studio DJ mixes have the same duration: 2 hours
- DJs are important/known: which implies that they create trends and are reliable examples
- Same studio implies same master: which avoids statistical differences in values of extracted automatic music descriptors
- Studio radio DJ mixes implies that there is no interaction between DJ and audience

The only contra is that the range of EDM genres is wide, but I used mixes of similar genres to build the collection. Mixes used for this research are the followings:

2015.10.24 - Essential Mix - Loco Dice
2015.09.19 - Essential Mix - Seth Troxler
2015.09.05 - Essential Mix - Lee Burridge
2015.05.09 - Essential Mix - Scuba
2015.04.25 - Essential Mix - Claptone
2015.03.28 - Essential Mix - Four Tet x Jamie XX
2015.02.28 - Essential Mix - Max Cooper
2015.10.31 - Essential Mix - Dave Clarke

4.2 Annotations & Taxonomy

The collection was annotated in respect to a taxonomy based according to the psychological perception of the “intensity of emotional experience”, independently of the actual part of the song (e.g. intro, verse, chorus…). Divided in four different levels it was annotated in respect to the following characteristics:

- **Drop:** when highest emotional experience is achieved. The bass and bass drum are dropped down and the main groove returns.
- **Build Up:** high emotional experience, but still having the expectation of something bigger coming in. The different instrument layers are built up one after another, giving massive musical and the sensation of strong emotional and bodily peak.
- **Build Up Filtrado:** softer emotional experience than Build Ups, because of presence of filters.
- **Breakdown:** lowest tension is achieved and seems like a “breath” within the mix. It is characterized by the track’s texture becoming considerably thinner or even being completely changed. Several instrument layers are removed, most importantly the bass and the ass drum.

The annotation was done with Sonic Visualiser, which is a free software for viewing and analysing the contents of music audio files. Developed at the Centre for Digital Music at Queen Mary, University of London, it is particularly interesting for musicologists, signal-processing researchers and anyone wanting to take a look at what lies inside the audio files. It allows to install feature-extraction plugins and has already-integrated audio visualisations such as spectrograms views with interactive adjustment of display parameters. Moreover, it allows annotating audio data by adding labelled time points and defining segments, point values and curves, which can be imported and exported to various external text file formats.

In this case, a comma-separated values (.csv) file was exported for each of the 2 hours long mixes, giving the point in time where a new section started and the previous finished. I also aggregated extra information, such as when a new song started and some specific comments when I thought it was interesting to explain or highlight something (see Appendix A to get the list of 40 hours of mixes annotated in respect of the point in time when a new song start). However, only the starting (and ending) point of each structural section was used for this project.

Thus, 16 hours were annotated, where 12 hours were used to train the system and 4 hours as an independent test set to perform the evaluation. Within the 14 hours-training-data, 637 different sections were annotated. The following number of each section annotated was:

- Drop: 72
- Build up: 305
- Filtered Build up: 199

- Breakdown: 61

4.3 Music descriptors extraction

Once the process of manually annotating was finished, automatic music descriptors were extracted in order to empirically get useful information to characterise each of the four different structural sections defined above. Timbral, temporal, tonal and some high-level inferred semantic information were thought to be needed in order to accurately describe each of the sections.

The automatic music descriptors were extracted using Essentia. It is an open-source C++ Library of algorithms for audio analysis and audio-based MIR. It has been developed at the Music Technology Group, in Universitat Pompeu Fabra, Barcelona. It contains a large collection of temporal, spectral, tonal, and high-level music descriptors, algorithms for audio input/output functionality, standard digital signal processing blocks and statistical tools. The library can be complemented with Gaia (Serra et al. 2013), a C++ library to apply similarity measured and classification on the result of the analysis. Each processing block is offered as an algorithm, and has three different types of attributes: inputs, outputs and parameters. So that, different blocks may be linked in order to perform the required processing task.

The chosen music descriptors to classify the structural sections were the *onset rate,* as it detects presence of percussive elements, the *loudness, energy* and *RMS* descriptors that refer to the perception of energy of the audio signal. Moreover, the *Bark bands* that represent directly the energy presence within the 27 bands and *MFCCs* that represent the power of the spectrum in 13 bands. Also, the *pitch salience* was used as it gives information about the tone sensation, and finally the *danceability* descriptor which informates about how a danceable a section is.

Extraction parameters & duration of structural sections

In order to define the parameters with which I was going to extract the automatic music descriptors, I decided to do a previous study on the durations of each of the sections. I found out that the shorter sections tended to be at least 10 or 15 seconds long, up to more than two or three minutes in the longest cases.

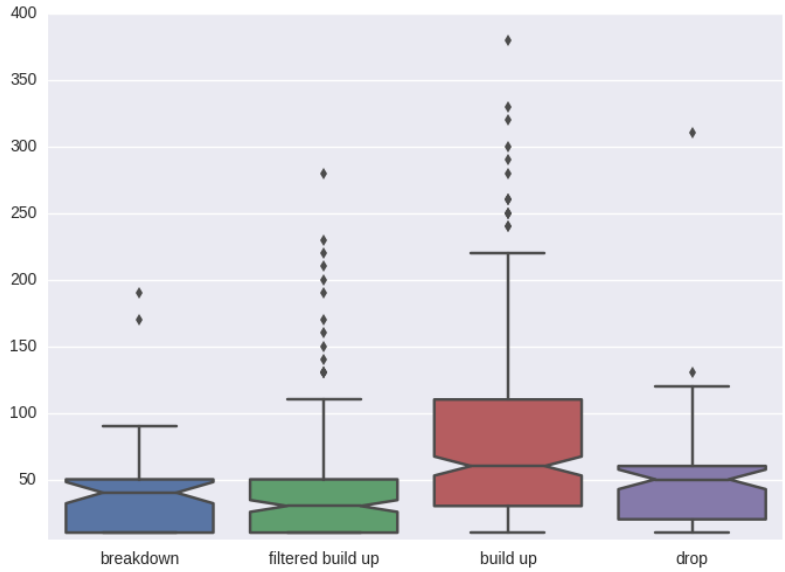

Figure 11: Duration in seconds of each of the structural sections

	Breakdown	Filtered Build Up	Build Up	Drop
median	40s	30s	60s	50s
stdev	34.375	65.289	65.289	51.806

Table 1: median and standard deviations of durations of structural sections

Therefore, having a quite vast range of section-durations and knowing that mixes were two hours long, I decided to perform the automatic music descriptors extraction by taking slices of 10 seconds, that I would call “time blocks”. The values of each of the different descriptors were averaged in order to get a single value for each of the time blocks. I also calculated the standard deviation and the slope for each of the time blocks, for *loudness, energy, RMS, spectral centroid, pitch salience, onset rate* and *danceability* descriptors by averaging its frame values within the time block. Each time block represented a row in the final .csv file, where the information of the different descriptors lied in each of the columns.

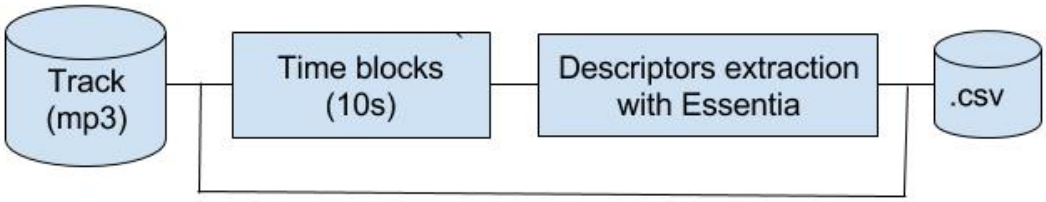

Figure 12: Basic schema of the extraction of descriptors, where track is segmented in time blocks of ten seconds and then descriptors are extracted.

All descriptors, except the *onset rate, BPM* and *danceability*, were extracted generating frames of 1024 samples with hops of 512 samples. For each of the frames *Hann* windowing is applied and FFT is performed to get a spectral representation of the signal. The *BPM* and *onset rate* were computed using the *BeatTrackerMultiFeature* algorithm, that computes a number of onset detection functions and estimates beat location candidates from them using *TempoTapDegara* algorithm.

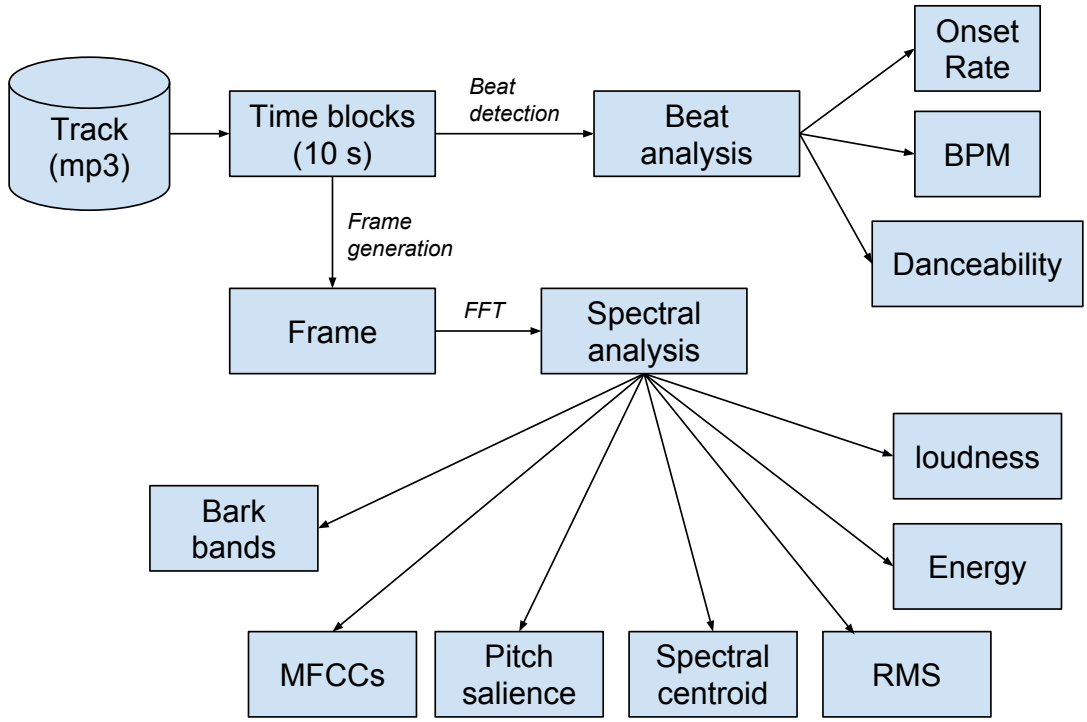

Figure 13: scheme of music descriptors extraction through beat and spectral analysis

Once both .csv files (the annotations and the descriptors values), for each mix were produced, they were put together in a single file. The annotations files consisted on two columns, where the first was the time label precisely written in seconds with three decimal numbers and the second column was the actual annotation. The descriptors files consisted in more than 50 columns, one for each descriptor and its standard deviations (if calculated), and around 720 rows, each representing a “time block” (10 seconds) for the two hour long mixes (i.e. 7.200 seconds long mixes).

In order to match both annotations with the descriptors values, the values of the points in time of the annotations were rounded to the closest multiple of ten. Knowing that the shortest sections found were longer than 10 seconds, there was no room of error to miss any of the annotations. Using the *vlookup* function in excel allowed to put them together by matching the time labels.

4.4 Analysis of structural sections

4.4.1 Extracted music descriptors analysis

Once a .csv file was produced for each of the mixes, having both the annotations, representing the structural sections, and the values of the extracted descriptors, an analysis was performed in order to observe similarities and differences between the sections, and understand what elements made a specific section distinctive.

Boxplots are a convenient way of graphically representing groups of numerical data through their quartiles. The bottom and top of the box are always the first and third quartiles, and the band inside the box represents the second quartile (median). They may have lines extending vertically from the boxes, called whiskers, which indicate the variability outside the upper and lower quartiles, where the lower end will represent the minimum and the upper end the maximum. Outliers, represented by points outside the range of the box, are observations that lie at more than 3/2 times upper or lower the quartiles. Below we can observe Boxplots representing the characteristics of each of the four structural sections (*Drop, Build Up, Filtered Build Up & Breakdown*) for the most relevant descriptors used.

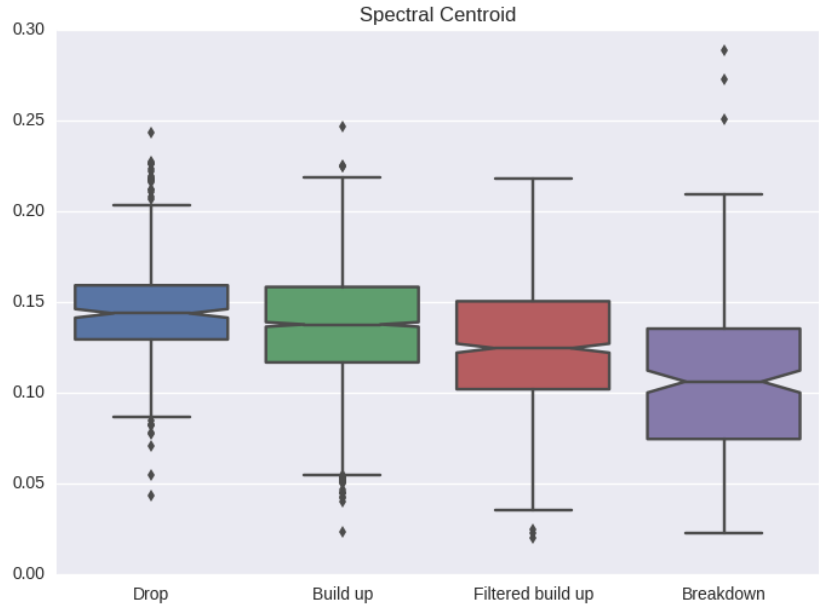

Figure 14: Boxplot representing the spectral centroid range for each of the structural sections

It can be observed that for the *spectral centroid* the *drop* presents the highest spectral centre of mass and that the breakdown has the lowest values. Some overlapping between all the sections is present, especially between both highest levels of emotional experience -*drop* and *build up*.

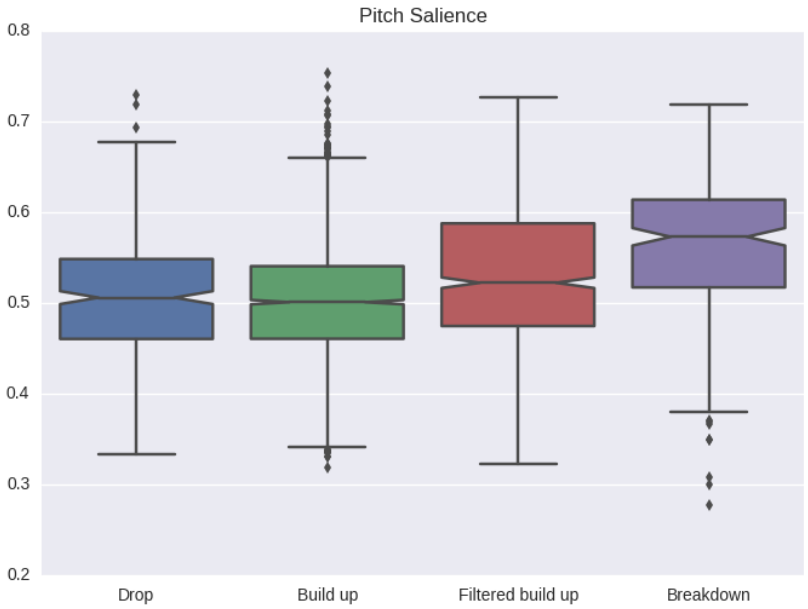

Figure 15: Boxplot representing the pitch salience range for each of the structural sections

Pitch salience discriminates well the *breakdowns* and *filtered build ups* (still with some overlapping), as the median is quite higher, unlike drops and *build ups,* which are very similar. A lot of overlapping can be observed between *build ups* and *drops*, as *build ups* are inside the range of the *drop*, which would lead to lower the accuracy of the classification.

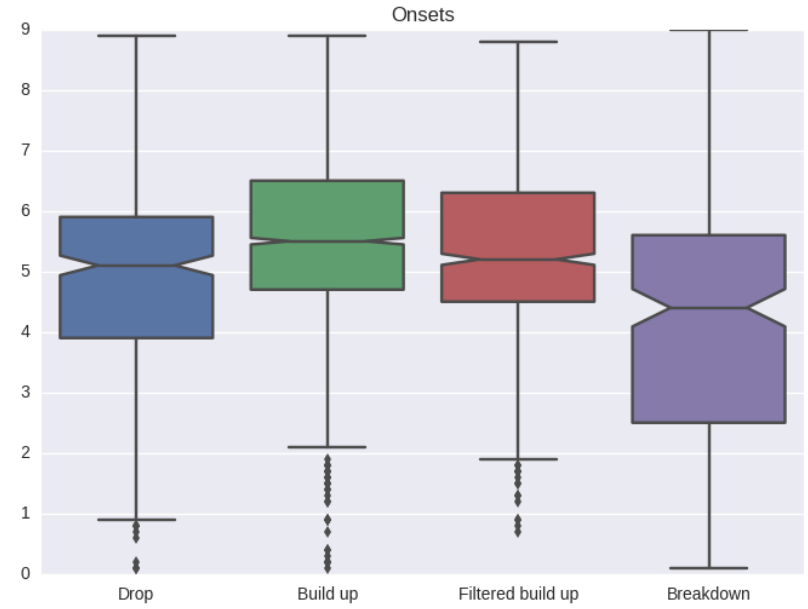

Figure 16: Boxplot representing the onset rate range for each of the structural sections

The highest *onset rate* is shown to be in the *build ups* and in the *filtered build ups*, where the values are quite similar, being a bit higher on the *build ups*. Moreover, it can be observed that the *breakdowns* present the lower *onset rate*. This descriptor would be useful to discriminate *breakdowns* and *drops*, but less accuracy would be achieved discriminating between *build ups* and *filtered build ups*.

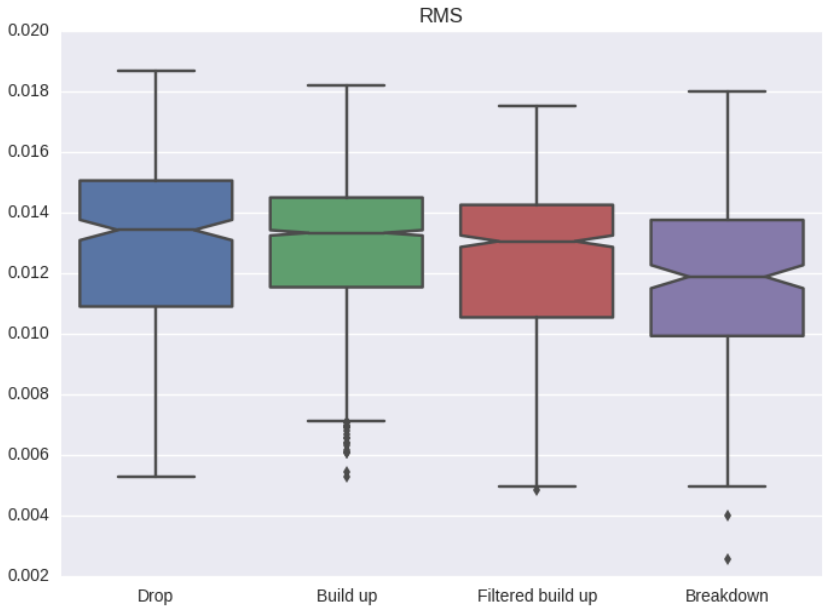

Figure 17: Boxplot representing the RMS range for each of the structural sections

RMS, loudness and *energy* present very high correlation, so that showing only one of the plots is enough. It can be observed than the range of the *build ups* lie inside the range of the *drop*, which would make the discrimination difficult. *Filtered build ups* mainly lie inside the drop range, but it goes a bit lower. *Breakdowns* present lower values, still with some overlapping, but enough to be reliable for classification.

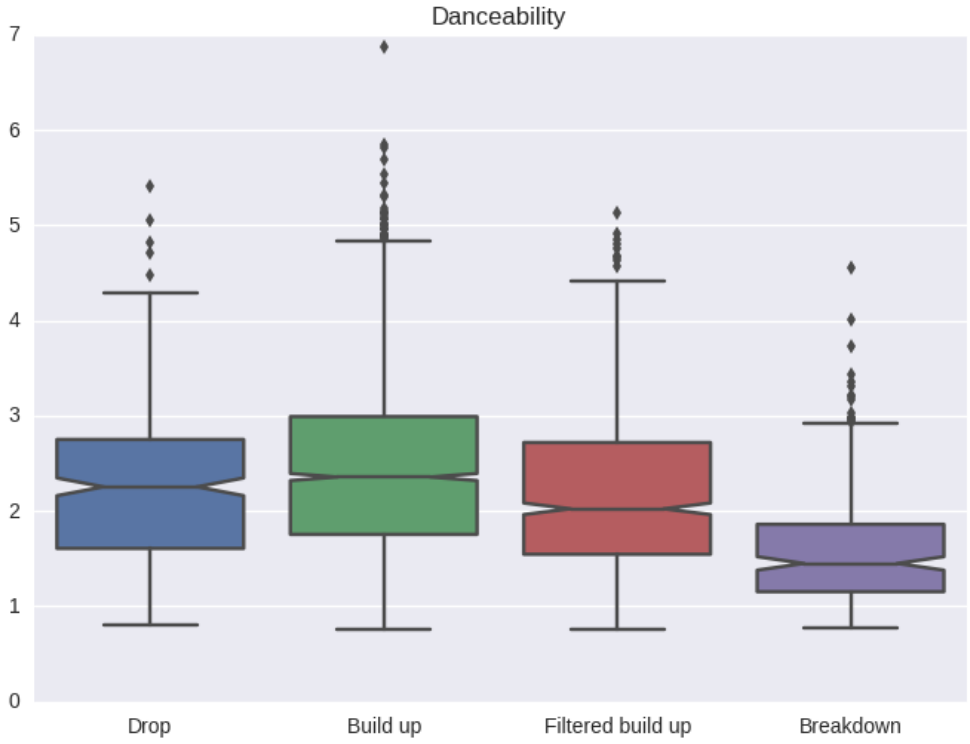

Figure 18: Boxplot representing the danceability range for each of the structural sections

Based on the *danceability* descriptor, the most danceable section seems to be the *build ups* and the *breakdowns* the less danceable. A lot of overlapping is present between the *drops*, *filtered build ups* and *build ups*, but still showing different median values, so that it would be meaningful and reliable for the classification.

The plot below shows the four first *Bark bands* for each of the four different sections. Where the first *Bark band* goes from 0-50 Hz, the second from 50-100 Hz, the third from 100-150 Hz and the fourth from 150-200 Hz.

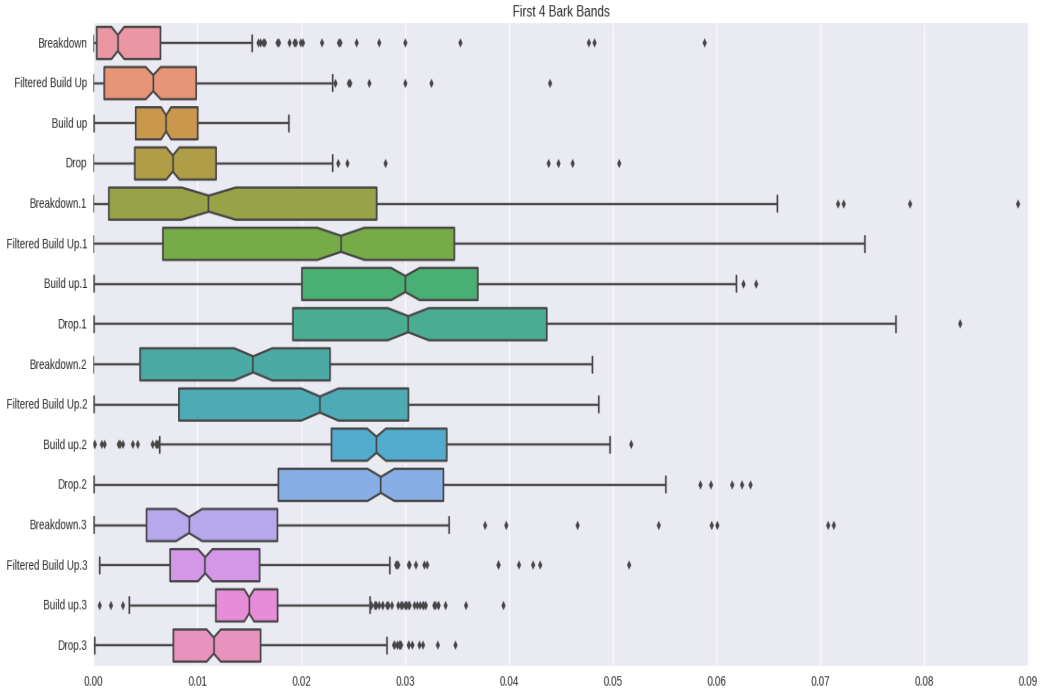

Figure 19: Boxplot representing the first four Bark bands ranges for each of the structural sections

In the first three bands it can be observed that the *build up* and *drop* overlap a lot, making the discrimination difficult. Unlike in the fourth band that we observe less overlapping and the median is quite lower in the drop. *Breakdown* always present the lowest values, followed by the *filtered build ups* as the second lower. This proves that *Bark bands* would be an important descriptor to discriminate between the sections, but it will show some difficulties discriminating between *drops* and *build ups.*

Looking at these boxplots we can conclude that, as they present many differences between structural sections, the descriptors used are reliable for a classification and some level of accuracy could be achieved. In many cases it could be observed that drops and *build ups* have very similar characteristics, which would be an inconvenience when classifying just based on these descriptors. This means that many drops and *build ups* do not differ much musically and could even be the exact same or very similar, and therefore some other elements have to be taken in account to properly discriminate between both sections.

4.4.2 Precedence of structural sections analysis

Analysing the ordering and looking for patterns of the structural sections that are often repeated, can give useful information to inform about the possibilities of having a specific section next. Hence, some statistics on the sequences of different sections were produced. When analysing audio content based information about some of the sections we do not observe much differences. The most clear situation can be seen between *build ups* and *drops*, which have very similar characteristics. Therefore knowing which are the precedent sections, can give significant information to predict and classify the actual one.

The psychological sensation emotional experience increase is directly related to the difference of tension between two consecutive parts. In some cases, it can be observed that the precedent sections have a high level of tension and the next part has been annotated as a build up, instead of as a *drop*. Otherwise, in some other cases it can be observed that the precedent section present low level of tension, so that when it achieves a higher tension, the psychological sensation of the intensity of the emotional experience is higher, because the difference of tension is bigger, and it is annotated as a *drop*.

Based on our annotated collection, statistics about the sequences and ordering of the structural sections were obtained. The tables show the normalized percentage of the sections appearing before the actual one. The first table show the probabilities of having specific sections as the first precedent ones and the second table, the probabilities of sections appearing as the second precedent ones.

1st Precedent	Breakdown	Filtered Build Up	Build Up	Drop
Breakdown	0	0.295	0.639	0.065
Filtered Build Up	0.074	0	0.806	0.118
Build Up	0.138	0.707	0	0.154
Drop	0.112	0.450	0.436	0

Table 2: Normalized percentage of the sections appearing as the first precedent section from the actual one

2nd Precedent	Breakdown	Filtered Build Up	Build Up	Drop
Breakdown	0.098	0.508	0.314	0.081

Filtered Build Up	0.118	0.682	0.092	0.105
Build Up	0.079	0.098	0.747	0.079
Drop	0.112	0.253	0.338	0.297

Table 3: Normalized percentage of the sections appearing as the second precedent section from the actual one

Often repeated patterns can be observed as there are sections with high percentages of appearing as the first and second precedent sections before the actual. For example, in the case of the filtered build up, there is a 80.6% of probability of having a build up just before, and again a filtered build up (68.2%) as the second precedent section. In the case of the build ups, there is a 70.7% of probability of having a filtered build up just before, and again having a build up (74.7%) as the second precedent section. It can be concluded that there is an often repeated pattern that goes from build ups to filtered build ups and viceversa.

4.5 Evaluation

An evaluation in order to assess the quality of the chosen descriptors when classifying structural sections in EDM is performed, and at the same time test if the taxonomy used can properly be described with automatic music descriptors. In this evaluation, randomly chosen independent time blocks are taken for classification. Moreover, improving the classification by giving extra information to the system, such as precedent sections, is investigated. This would be in an ideal case where the system could read the predicted precedent sections from the actual one, and use them to make predictions not only based on extracted descriptors information.

Many different evaluations were performed with different classifiers, parameters and training methods using Weka. Thus, the three most prominent evaluations are explained in this section. The following section briefly explains what Weka is and give some insights about the machine learning algorithms and evaluation methods used.

4.5.1 Evaluation tools and methods

4.5.1.1 Weka

Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data pre-processing, classification, clustering, association rules, and visualization among others. It is also well-suited for developing new machine learning schemes. It gives detailed information about the accuracy of each section, such as:

- TP Rate: rate of true positives (instances correctly classified as a given class)
- FP Rate: rate of false positives (instances falsely classified as a given class)
- Precision: proportion of instances that are truly of a class divided by the total instances classified as that class
- Recall: proportion of instances classified as a given class divided by the actual total in that class (equivalent to TP rate)
- F-Measure: A combined measure for precision and recall calculated as 2 * Precision * Recall / (Precision + Recall)
- The ROC area measurement, which is one of the most important values output by Weka. An "optimal" classifier will have ROC area values approaching 1, with 0.25 being comparable to "random guessing" in this case, as there are four possibilities.

It should be noted that the "balance" of the data set needs to be taken into account when interpreting results. Unbalanced data sets in which a disproportionately large

amount of instances belong to a certain section may lead to high accuracy rates even though the classifier may not necessarily be particularly good.

4.5.1.2 J-48 Tree classifier

A decision tree is a predictive machine-learning model that decided the target value (dependent variable) of a new sample based on various attribute values of the available data. The internal nodes of a decision tree denote the different attributes, the branches between the nodes tell us the possible values that these attributes can have in the observed samples, while the terminal nodes tell us the final value (classification) of the dependent variable. The attribute that is to be predicted is known as the dependent variable, since its value is decided by the values of all the other attributes. The other attributes, which help in predicting the value of the dependent variable, are known as the independent variables in the dataset[4].

The J-48 decision tree classifier follows the following simple algorithm. In order to classify a new item, it first needs to create a decision tree based on the attribute values of the available training data. So, whenever it encounters a set of items (training set) it identifies the attribute that discriminates the various instances most clearly. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign to it the target value that we have obtained.

For the other cases, we then look for another attribute that gives us the highest information gain. Hence, we continue in this manner until we either get a clear decision of what combination of attributes gives a particular target value, or we run out of attributes. In this case, or if we cannot get an ambiguous result from the available information, we assign this branch a target value that the majority of the items under this branch possess.

Now that we have the decision tree, we follow the order of attribute selection as we have obtained for the tree. By checking all the respective attributes and their values with those seen in the decision tree model, we can assign or predict the target value of this new instance.

[4] http://www.d.umn.edu/~padhy005/Chapter5.html

4.5.1.3 Support Vector Machine (SVM) & Sequential Minimal Optimization (SMO)

Support Vector Machine (SVM) is a supervised machine-learning algorithm, which can be used for classification problems. In this algorithm, each data item is plotted in a n-dimensional space (where n is the number of features) with the value of each feature being the value of a particular coordinate. Then, classification is performed by finding the hyper-plane that differentiates the two classes very well. Support vector are simply the coordinates of individual observation, and support vector machine is a frontier, which best segregates the two classes[5].

SMO, implemented by the LIBSVM tool (Chih, 2011) is one way to solve the SVM training problem that is more efficient than standard quadratic programming (QP) solvers[6], which are a special type of mathematical optimization problem (for minimizing or maximizing). SVM training methods were much more complex and required expensive third-party QP solvers. SMO uses heuristics to partition the training problem into smaller problems that can be solved analytically. It globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default. In that case the coefficients in the output are based on the normalized data, and not the original data[7]. Whether or not works well depends largely on the assumptions behind the heuristics (working set selection).

4.5.1.4 Cross validation

Cross validation is a model evaluation method that is better than than residuals[8]. The problem with residual evaluations is that they do not give an indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training. Some of the data is removed before training begins. Then, when training is done, the data that was removed can be used to test the performance of the learned model on “new data”. This is the basic idea for the whole class of cross validation methods for model evaluation.

The holdout method is the simplest kind of cross validation. The data is separated into two sets, called the training set and the testing set. The system is trained with the training set only and it is asked to predict the output values in the testing set, which it has never seen the output values before. The K-fold cross validation is one way to improve over the holdout method. The data is divided into *k* subsets, and the holdout method is repeated *k* times. Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set. Then, the

[5] https://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/
[6] http://stats.stackexchange.com/questions/130293/svm-and-smo-main-differences
[7] http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html
[8] https://www.cs.cmu.edu/~schneide/tut5/node42.html

average error across all *k* trials is computed. The advantage of this method is that it matters less how the data get divided, because every data point get to be in a test set exactly once, and gets to be in a training set *k-1* times.

5. Results

Two pre-evaluations and a final evaluation is presented in this section. The first two evaluations are performed using cross-validations of 5 folds using a J-48 tree classifier and a SMO classifier. In both evaluations it is tested how would the prediction improve in an ideal case when the system knows the two sections that precede the actual one. Finally, the last evaluation is performed using the SMO method and an independent set, not used for training but for evaluation only.

The .csv file introduced in Weka consisted on 6 DJ-created mixes, with a total duration of 12 hours, in which the 4236 time blocks present represented:

- Drop: 373
- Build up: 2701
- Filtered build up: 917
- Breakdown: 245

(Note that 12 hours represent 8640 time blocks, so that 4404 time blocks were removed as they did not represent any of the structural sections used in the taxonomy. In this parts we could find commentator talks during the session -every 15 or 20 minutes-, introducing the artist at the beginning and a final talk at the end of the set).

An issue that appeared when performing the first evaluations, as the classes are somehow unbalanced, meaning that some classes have much more instances than others, the classificator tended to choose the most common class the most part of the time. Therefore, a pre-prefiltering was done, using the *SpreadSubSample* filter in Weka and defining the *MaxCount* parameter to the double of the number of times that the rarest section appeared. This filter allows you to specify the maximum "spread" between the rarest and most common class[9]. In this case, *MaxCount* was defined to 490, which is the double of the number of times that *Breakdowns* were present, which was the rarest section.

Also note that, for some unknown reason, standard deviations and slopes calculated for the values of the descriptors between time blocks were removed for the evaluation, as they did not allow to use the *SpreadSubSample* filter.

[9] http://weka.sourceforge.net/doc.stable/weka/filters/supervised/instance/SpreadSubsample.html

5.1 First evaluation: J-48 Tree classifier & Cross-validation with 5 folds

The first evaluation was performed using a J-48 Tree classifier and a cross-validation with 5 folds. There were 47 different attributes (47 columns in the same .csv), where we find: *RMS, Loudness, Energy, Spectral Centroid, Pitch Salience, Onset Rate, Danceability,* 13 *MFCC* bands and 27 *Bark* bands. We add two more variables to represent the 1st precedent & 2nd precedent sections, getting a total of 49 attributes. The .csv file consisted on a total of 50 columns counting the annotated actual section used as the ground truth.

The *minimum number of instances per branch* was manipulated with the idea of reducing the complexity of the classification without loosing much accuracy on the prediction. It was decided to set the *minimum number of instances per branch* at 20, as it considerably reduced the complexity of the tree and did not affect in excess the accuracy of the system. Increasing this parameter continuously reduced the accuracy of the system without anymore reducing much the complexity of the tree.

Min. # instances per branch	2 (default)	20
Correct Classified Instances	52.112%	50.876%
# of leaves	236	42
Size of tree	471	83

Table 4: Correlation between the minimum number of instances per branch and the accuracy obtained

The table above shows the difference in accuracy of correctly classified instances, number of leaves and size of the tree between both classifiers, one with the default value of minimum number of instances per branch, which is 2, and the further modified tree with 20 as the minimum number of instances per branch.

Moreover, using only the audio content extracted from the descriptors 50.876% of correctly classified instances was achieved. The figure below shows more detailed information about the classification for each section:

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.545	0.245	0.496	0.545	0.519	0.293	0.701	0.493	build up
	0.473	0.217	0.492	0.473	0.482	0.260	0.678	0.470	build up filtrado
	0.469	0.072	0.542	0.469	0.503	0.422	0.799	0.463	breakdown
	0.534	0.144	0.529	0.534	0.531	0.388	0.742	0.490	drop
Weighted Avg.	0.509	0.186	0.510	0.509	0.508	0.325	0.718	0.480	

The confusion matrix below shows accuracy of the predicted instances just using music descriptors. Each column of the matrix represents the instances in a predicted

section while each row represents the instances in the actual class, being 1598 the number of instances.

	Build up	Filtered build up	Breakdown	Drop
Build up	257	102	27	94
Filtered build up	143	232	55	60
Breakdown	38	69	115	23
Drop	90	69	15	199

Table 5: Confusion matrix of first evaluation using J-48 classifier and music descriptors

In the ideal case that the system could use the predicted precedent parts and use them to improve the classification, by taking the the 1st precedent part, it would improved considerably, to the point of achieving a 64.64% of correctly classified instances. Moreover, taking in account both 1st and 2nd precend parts improved a bit more the classification to a 66.58% of correctly classified instances. The confusion matrix below shows the improvement on the accuracy of the predicted sections using not only audio music descriptors, but also both 1st and 2nd precedent parts.

	Build up	Filtered build up	Breakdown	Drop
Build up	368	44	18	60
Filtered build up	47	340	38	65
Breakdown	17	64	129	35
Drop	77	53	16	227

Table 6: Confusion matrix of first evaluation using J-48 classifier and both music descriptors and precedent parts

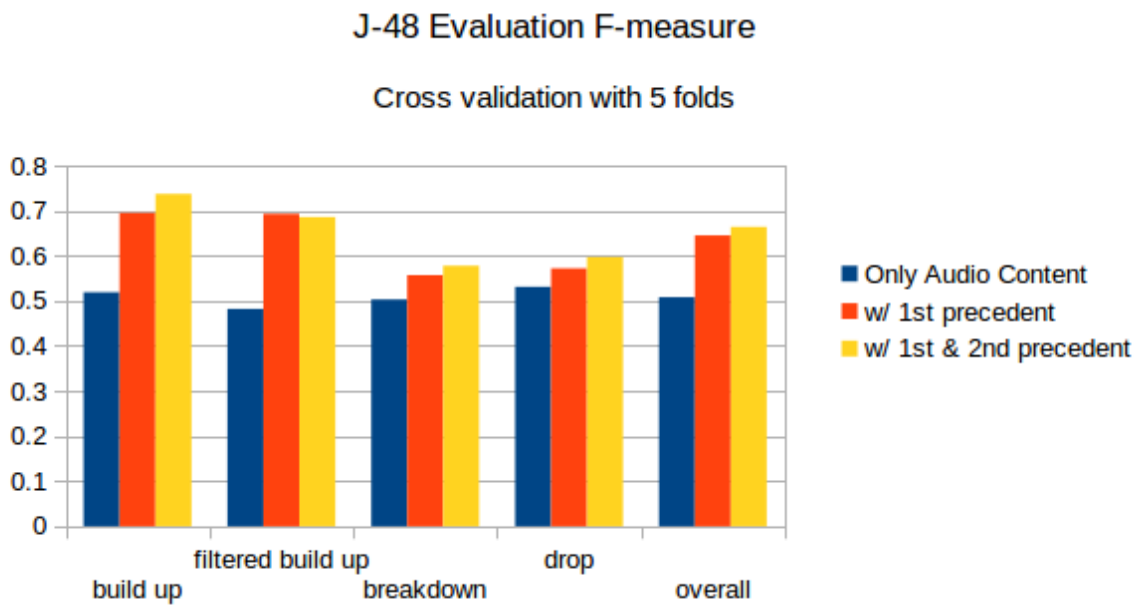

Figure 20: F- measure representing accuracy improvement adding precedent parts using J-48 classifier

5.2 Second evaluation: SMO with a cross validation of 5 folds

The second evaluation was performed using a SMO classifier and a cross-validation with 5 folds. There were 47 different attributes (47 columns in the same .csv), the same as in the previous evaluation. Further, by adding two more variables to represent the 1st precedent & 2nd precedent sections, getting a total of 49 attributes, plus the actual section, the .csv file consisted on a total of 50 columns.

Taking in account only the audio descriptors for the classification, a 51.627% of correctly classified instances was achieved. See the figure below for detailed prediction accuracy information for each of the sections:

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.604	0.329	0.448	0.604	0.514	0.257	0.669	0.412	build up
	0.500	0.218	0.504	0.500	0.502	0.283	0.666	0.425	build up filtrado
	0.469	0.038	0.693	0.469	0.560	0.510	0.807	0.489	breakdown
	0.453	0.095	0.593	0.453	0.514	0.396	0.749	0.450	drop
Weighted Avg.	0.516	0.196	0.537	0.516	0.517	0.336	0.708	0.437	

The confusion matrix below shows accuracy of the predicted 1598 instances just using music descriptors.

	Build up	Filtered build up	Breakdown	Drop
Build up	296	120	7	67
Filtered build up	171	245	35	39

Breakdown	47	73	115	10
Drop	147	48	9	169

Table 7: Confusion matrix of second evaluation using SMO classifier and music descriptors

In an ideal case, taking in account the 1st precedent part improved from 51.627% up to a 69.211%, and using the 1st and 2nd precedent parts improved the classification a bit more, obtaining a 70.776%.

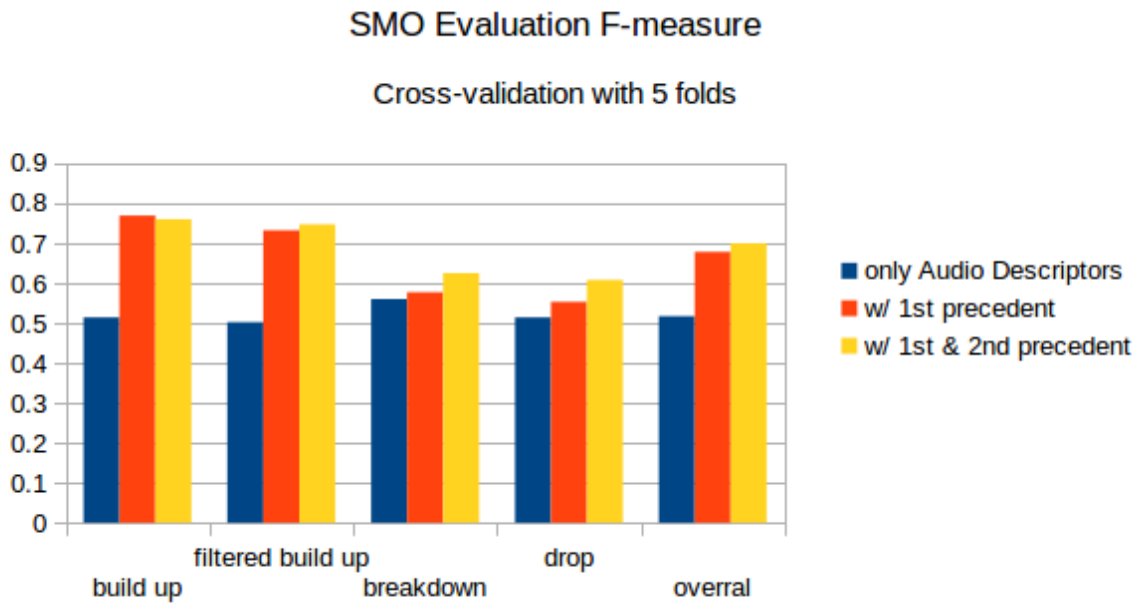

Figure 21: F- measure representing accuracy improvement adding precedent parts using SMO classifier

5.3 Third evaluation: SMO with independent test set

For the last and most reliable evaluation, the SMO method and an independent test set was used. Testing with an independent test set was needed as it is completely new data for the system, and therefore it will give more reliable information for someone willing to test the system with his own data.

The training and evaluation data consisted again of a .csv file with 47 different attributes, where we found the calculated mean averages for each time block for the following descriptors: *RMS, Loudness, Energy, Spectral Centroid, Pitch Salience, Onset Rate, Danceability,* 13 *MFCC* bands and 27 *Bark* bands. The test data was obtained from two hours of DJ-created mixes, specifically:

- 2015.02.28 - Essential Mix - Max Cooper
- 2015.10.31 - Essential Mix - Dave Clarke

The independent test contains 1353 time blocks or instances, where number of sections are the following:

- Drop: 254
- Build up: 520
- Filtered build up: 419
- Breakdown: 160

The accuracy achieved was 41.093% of correctly classified instances, which represents around 10% less accuracy than both previous evaluations performed with the same data to train and test the system's accuracy. The detailed accuracy for each section obtained was:

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.390	0.277	0.468	0.390	0.426	0.118	0.567	0.401	build up
	0.296	0.115	0.537	0.296	0.382	0.223	0.630	0.397	build up filtrado
	0.250	0.003	0.909	0.250	0.392	0.449	0.743	0.439	breakdown
	0.744	0.414	0.293	0.744	0.421	0.258	0.690	0.273	drop
Weighted Avg.	0.411	0.220	0.509	0.411	0.407	0.216	0.630	0.380	

With a total number of instances of 1353, the confusion matrix shows accuracy of the predicted instances. Each column of the matrix represents the instances in a predicted section while each row represents the instances in the actual class.

	Build up	Filtered build up	Breakdown	Drop
Build up	203	46	2	269
Filtered build up	143	124	1	151
Breakdown	39	46	40	35
Drop	49	15	1	189

Table 8: Confusion matrix of third evaluation using SMO classifier and music descriptors tested with an independent set

5.4 Discussion

The taxonomy was defined based on previous multidisciplinary studies on EDM. However, a new structural section never utilised before was defined as *Filtered build up*, which as the name indicates, is a filtered version of the -commonly used and established within the literature- *Build up.* DJs tend to play around with filters during the *Build ups* in order to manipulate the intensity of it (e.g: usually low frequencies are filtered before a new layer is added, in order to increase the intensity when it comes in). Therefore, this new section labeled as *Filtered build up* is a way of dividing *Build ups* in two different sections.

The first two evaluations, using the J-48 tree and the SMO classifiers with cross validation method with 5 folds, did not present huge differences. By general means the SMO method worked a bit better, obtaining a 51.6% of correct classified instances, compared to the 50.8% achieved by the J-48. However, the precision-recall curve (PRC) showed that J-48 did more accurate predictions for *Build ups*, *Filtered build ups* and *Drops*, and SMO showed higher accuracy for *Breakdowns*. For both methods, the ROC values lied higher than 0.7, which reflects the reliability of the used descriptors as it is far above the 0.25, which would represent random guessing.

The final evaluation, using an independent test set, showed lower accuracy than previous evaluations as it was expected, but still achieving 41.09% of correctly classified instances. *Build ups* tended to be classified as *Drops*, which presents the lower PRC value at 0.273. Moreover, *Filtered build ups* tended to be classified as *Build ups* and *Drops*. The overall ROC value was 0.63, which again reflects the reliability of the used descriptors as it is far above the 0.25, which would represent random guessing.

The variability that characterises the structural sections, particularly the *Build ups* and *drops*, is huge, and therefore the prediction based on music descriptors is a challenging task. As still predictions errors when discriminating between structural sections were found, extra information was given to the classifier classification apart of the extracted music descriptors. As shown in the state of the art, solutions for *Drop* detection have been proposed (see Yadati 2014; Aljanaki 2014), such as using drop-related comments on the Soundcloud platforms, or noisy labels collected through crowdsourcing. A solution was proposed for this study, which in an ideal case where the system evaluated linearly from the beginning of the mix to the end, and could use its previous predictions to incorporate them in a correction system, showed that the improvement would be significant. For the J-48 tree classification the improvement would go up to 66.58% of correct classified instances taking in account both the first and second precedent sections, and for the SMO classifier up to 70.77%.

6. Conclusion

The taxonomy based on previous studies on the topic and the new section that I created -the *Filtered build up*- showed reliability as statistical differences were consistent. The variability of the structural sections makes the classification a challenging task, but this study proves that they can be described and discriminated with music descriptors extracted from the audio itself, as the performance of the predictions showed some level of accuracy considerably higher than 0.25 ROC values, which implies random guessing as there are four different sections. However, there is still room for improvement and a solution is proposed. It is shown that, in an ideal case of having a system with an integrated correction system that could read the previous predictions and use them to take part on the decision for the next classification, would significantly improve the system's accuracy.

7. Future Work

- Research could be done on improving the classification by using a higher number of music descriptors.
- Instead of analysing the mixes in respect to time, it could be done in respect to the bars or beats, as we know that EDM structures rely on layers which are added on top of each other every multiples of 4 bars.
- Creating a correction system would be an interesting way to improve the classification, as many different information could be given to take part on the prediction decisions, such as:
    - Length of structural sections: it has been shown that different sections tend to present different durations
    - Precedent sections: there are structural patterns that are often repeated
- Create a descriptor for each of the classes

8. Reproducibility

The material used for this study can be found in Github with all the permissions of use. Download it in the following link:

https://github.com/vincentzurita/MasterThesisMaterial

There are 5 folders:

- **Essential mixes**: The collection in mp3
- **Starting point of songs Annotations**: .csv file for each 20 DJ mixes (40 hours) with annotations giving the point in time when a new song stars.
- **Structural sections Annotations:** .csv file for each of the 8 DJ mixes (16 hours) with the annotations giving the point in time when a new structural sections starts.
- **Music descriptors x Structural sections Annotations:** .csv file of each of the 8 DJ mixes (16 hours) with the values of the extracted descriptors and the annotations giving the points in time when both new structural section & new song starts.
- **Scripts**:
  - **read.py**: extracts BPM, pitch salience, spectral centroid loudness, energy, RMS, danceability & onset rate, averaging its values every 10 seconds.
  - **barkbands.py**: extracts 27 bark bands, averaging its values every 10 seconds.
  - **mfcc.py**: extracts 13 MFCC bands, averaging its values every 10 seconds.

9. Bibliography

Aljanaki, A., Soleymani, M., Wiering, F., & Veltkamp, R. C. (2014). MediaEval 2014: A multimodal approach to drop detection in electronic dance music. In CEUR Workshop Proceedings.

Bandiera G., “A Content-Aware Interactive Explorer of Digital Music Collections : The Phonos Music Explorer,” 2015.

Barrington L., D. Turnbull, D. Torres, and G. Lanckriet, “Semantic similarity for music retrieval,” Int. Symp. Music Inf. Retr. ({ISMIR’07)}, 2007.

Bello, J. P. (2016). Systematic Exploration of Computational Music.

Butler, Mark J. 2005. “Hearing Kaleidoscopes: Embedded Grouping Dissonance in Electronic Dance Music”. twentieth-century music 2(2): 221–43. <http://dx.doi.org/10.1017/S1478572206000272>.

Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Dancecult: Journal of Electronic Dance Music Culture 1(1): 81–93. http://dx.doi.org/10.12801/1947-5403.2009.01.01.05

Davies M. E. P. and Plumbley M. D. , “Context-dependent beat tracking of musical audio,” IEEE Trans. Audio, Speech Lang. Process., vol. 15, no. 3, pp. 1009–1020, 2007.

Degara N., E. A. Rua, A. Pena, S. Torres-Guijarro, M. E. P. Davies, and M. D. Plumbley, “Reliability-Informed Beat Tracking of Musical Signals,” IEEE Trans. Audio. Speech. Lang. Processing, vol. 20, no. 1, pp. 290–301, Jan. 2012.

Dibben, Nicola and Maria Witek. 2009. “An Exploratory Study of Physiological and Emotional Responses to Groove-Based Music”. Durham Music Department (Durham): International Conference on Music and Emotion.

Fikentscher, Kai. 2000. “You Better Work!”: Underground Dance Music in New York City. Hanover, N.H.: University Press of New England.

Fitzgerald, “Harmonic / Percussive Separation using Median Filtering Harmonic Percussive Separation using Median Filtering,” 2010.

Gabrielsson, Alf. 2011a. Strong Experiences with Music. Oxford: OUP Oxford.

Garcia, Luis-Manuel. 2011.“‘Can you Feel it, Too?’: Intimacy and Affect at Electronic Dance Music Events in Paris, Chicago, and Berlin”. Ph.D. Dissertation (Ethnomusicology), University of Chicago.

Gómez E., “Tonal Description of Polyphonic Audio for Music Content Processing,” INFORMS J. Comput., vol. 18, no. 3, pp. 294–304, Aug. 2006.

Gomez, Patrick and Brigitta Danuser. 2007. “Relationships between Musical Structure and Psychophysiological Measures of Emotion”. Emotion 7(2): 377–87. <http://dx.doi.org/10.1037/1528-3542.7.2.377>.

Grewe, Oliver, Reinhard Kopiez and Eckart Altenmüller. 2009. “Chills as an Indicator of Individual Emotional Peaks”. Annals of the New York Academy of Science 1169: 351–4. <http://dx.doi.org/10.1111/j.1749-6632.2009.04783.x>.

Guhn, Martin, Alfons Hamm and Marcel Zentner. 2007. “Physiological and Musico-Acoustic Correlates of the Chill Response”. Music Perception 24(5): 473–83.

Hawkins, Stan. 2008. “Temporal Turntables: On Temporality and Corporeality in Dance Culture”. In Musicological Identities: Essays in Honor of Susan McClary, ed. Steven Baur, Raymond Knapp and Jacqueline Warwick, 121–34. Aldershot: Ashgate.

Huron, David. 2006. Sweet Anticipation: Music and the Psychology of Expectation. Cambridge, Mass.: MIT Press.

Jackson, Phil. 2004. Inside Clubbing: Sensual Experiments in the Art of Being Human. Oxford, New York: Berg

Jennings H. D. et al., “Variance fluctuations in nonstationary time series: comparative study of music genres,” Condensed Matter (2003). http://xxx.lanl.gov/abs/cond-mat/0312380

Lamere P. and West K., “A model=based approach to constructing music similarity functions,” EURASIP J. Adv. Signal Process., vol. 2007, 2007.

Maslow, Abraham H. 1968 [1962]. Toward a Psychology of Being. New York: D. Van Nostrand.

Meyer, Leonard B. 1956. Emotion and Meaning in Music. Chicago: University of Chicago Press.

Micheline. (2005). Music Information Retrieval Conceptual Framework, Annotation and User Behaviour.

Montano, Ed. 2009. “DJ Culture in the Commercial Sydney Dance Music Scene”.

Reynolds, Simon. 1998. Energy Flash: A Journey Through Rave Music and Dance Culture. London: Picador

Rickard, Nikki S. 2004. “Intense Emotional Responses to Music: a Test of the Psychological Arousal Hypothesis”. Psychology of Music 32(4): 371–88. http://dx.doi.org/10.1177/0305735604046096

Schnitzer, D, “MIRAGE–High=Performance Music Similarity Computation and Automatic Playlist Generation,” Master’s thesis, Vienna Univ. Technol., vol. 17, pp. 135–135, 2007.

Serra X., M. Magas, E. Benetos, M. Chudy, S. Dixon, A. Flexer, E. G´omez, F. Gouyon, P. Herrera, S. Jorda, O. Paytuvi, G. Peeters, J. Schlu¨ter, H. Vinet, “Roadmap for Music Information Research.” [Online]. Available: https://www.academia.edu/8710899/Roadmap_for_Music_Information_Research. [Accessed: 22-Apr-2015].

Snoman, Rick. 2009 [2004]. The Dance Music Manual: Tools, Toys and Techniques. Oxford: Focal Press.

Solberg R. T. , "'Waiting for the Bass to Drop': Correlations between Intense Emotional Experiences and Production Techniques in Build-up and Drop Sections of Electronic Dance Music," Dancecult: Journal of Electronic Dance Music Culture, vol. 6, no. 1. pp. 61–82, 02-Jun-2014.

Solberg R. T., "'Moved by the music': Affective arousal, Body Movement and Musical Features of Electronic Dance Music," in Ninth Triennial Conference of the European Society for the Cognitive Sciences of Music (ESCOM2015), 2015.

Stevens S.S., "On the psychophysical law.," Psychol. Rev., vol. 64, no. 3, pp. 153–181, 1957.

Streich, S., & Herrera, P. (2005). Detrended Fluctuation Analysis of Music Signals: Danceability Estimation and further Semantic Characterization.

Vorobyev, Y., & Coomes, E. (2012). Beyond beatmatching : take your DJ career to the next level. Mixed In Key.

Yadati, K., & Larson, M. (2014). Detecting drops in electronic dance music: Content-based approaches to a socially significant music event. Proceedings of the 15th ..., (Ismir), 143–148

Zeiner-Henriksen, Hans T. 2010. The Poumtchak Pattern: Correspondences Between Rhythm, Sound, and Movement in Electronic Dance Music. Ph.D. Dissertation (Musicology), University of Oslo.

Appendix A

The list below shows the .csv annotated files that give the starting point of new songs coming in:

2014.08.09 - Essential Mix - Skream in Ibiza.csv
2015.08.15 - Essential Mix - Jose Padilla.csv
2015.05.09 - Essential Mix - Scuba.csv
2015.03.28 - Essential Mix - Four Tet x Jamie XX.csv
2015.03.21 - Essential Mix - Kryder.csv
2015.02.21 - Essential Mix - Redlight.csv
2015.10.24 - Essential Mix - Loco Dice.csv
2015.09.05 - Essential Mix - Lee Burridge.csv
2015.01.24 - Essential Mix - Tale Of Us.csv
2015.08.29 - Essential Mix - Rudimental.csv
2015.10.17 - Essential Mix - Ed Banger Special.csv
2015.04.25 - Essential Mix - Claptone.csv
2015.10.31 - Essential Mix - Dave Clarke.csv
2015.01.31 - Essential Mix - Joris Voorn.csv
2015.10.03 - Essential Mix - Flume.csv
2015.07.31 - Essential Mix - Hot Since 82.csv
2015.05.02 - Essential Mix - Julio Bashmore.csv
2015.09.19 - Essential Mix - Seth Troxler.csv
2015.02.28 - Essential Mix - Max Cooper.csv
2015.10.31 - Essential Mix - Dave Clarke