THE EFFECT OF TEMPO TRANSFORMATIONS ON ESSENTIA'S BEAT TRACKING PIPELINES

Vykintas Čivas
Supervisors: Jaehun Kim , Cynthia Liem
EEMCS, Delft University of Technology, The Netherlands
v.civas-1@student.tudelft.nl, {J.H.Kim, C.C.S.Liem}@tudelft.nl

Abstract

Beat detection is an important MIR research area. Due to its growing usage in multimedia applications, the need for systematic ways to evaluate beat detectors is growing too. This research tests *RhythmExtractor2013*, a pipeline offered by *Essentia*, an open-source music analysis library used in research and industry. The annotated test samples, taken from four open-source datasets - GTZAN, Ballroom, SMC_MIREX and MDB Drums, had tempo transformations (uniform, randomized, incremental and decremental tempo changes) applied to them and put to test against the aforementioned extractor. F-measure was chosen to calculate the extractor's accuracy. The results show, that the accuracy is affected mostly by the presence of steady rhythm and drums, but also by the window size during the result calculation process, with the worst scores appearing when the samples are slowed down.

Keywords

beat tracking, Essentia, tempo transformations, robustness

1 Introduction

Music Information Retrieval (MIR) is a growing research field which tackles the problem of managing music in digital format and working with it [1]. MIR is an interdisciplinary area where theoretical and practical knowledge of music and engineering is required in order to perform various tasks. Nevertheless, despite MIR seemingly being a very niche field, the outcomes of researches are being used by industrial companies, such as Spotify for music retrieval and music recommendation [2], Shazam for audio recognition [3], Sony (Sony R&D Center) for noise cancelling or source separation [4] but also in other machine learning tasks, such as audio alignment, cover song identification, query by humming and query by tapping [5].

In order to represent music in a digital format to be able to use it in MIR tasks, features have to be extracted from audio files. Low-level features can be extracted from music clips to find out information, such as, loudness and MFCC's (Mel-frequency cepstral coefficients), and these can be consecutively used to extract high-level features, such as, genre, mood or tags. As there are not so many open-source music databases available (due to copyright issues) and the ones which are available are fairly small, researchers turn to crowd-sourced public datasets, such as AcousticBrainz, which contain acoustic information about all kinds of musical pieces [6]. Such platforms allow individuals to calculate these features offline and upload them for everyone to use. As this allows for duplicate music clips to be analysed, problems arise when the extracted features of identically sounding music files are represented differently and influences subsequent stages of various pipelines. This can happen due to a number of reasons, for instance, *(i)* different audio encodings (mp3, wav, ogg to name a few), as suggested by Liem and Mostert [7], *(ii)* skewed datasets where not so popular musical concepts appear rarer and therefore biases transfer downstream the pipeline when learning [8] and *(iii)* musical transformations which are imperceptable to a human ear, e.g., adding small amounts of noise. Indeed, Sturm et al. [9] have shown that even small perturbations already increase the probability of false negatives.

There are, nonetheless, transformations which should be reflected by the output. For instance, pitch estimation models will produce different results on pitch shifted samples. In general, some pipelines might drop important information or produce a lot of meaningless information about the input when it is affected by transformations of various magnitudes while very robust pipelines might not suffer so much. For example, some feature extractors might not capture important elements of music recordings (e.g., frequencies of various percussive instruments) which could then in turn affect beat extraction algorithms [10].

Kim et al. [11] already showed that certain transformations, such as tempo changes, pitch shifting and adding noise, reduce the performance of neural network architectures, in particular VGG-like. These results open a

new gate for more in-depth studies which take into account other music processing pipelines, more transformations and different magnitudes of these transformations. On top of that, to be able to weigh the effect of such transformations on processing pipelines, it is also necessary to understand the perceptual effect of the magnitudes of the transformations. It is necessary to identify meaningful relations between different genres and musical transformations which in turn requires an understanding of what features should be extracted from and therefore what variations of pipelines should be applied to an audio clip of a particular musical style. The remainder of this paper will focus on temporal transformations with the hope to shed more light on on what level they affect *Essentia's*[1] beat tracking pipelines and draw a relation between the transformations and the pipelines in order to strengthen the expectations about the outputs.

The paper adapts the following structure: section 2 will in depth describe research goals and rationale; section 3 will talk about methodology of the research and its setup. Section 4 will summarize and give the analysis of the results of the experiments. In section 5, a global discussion over the results will be given, including the limitations. Lastly, section 6 will talk about the integrity of the research followed by the conclusion and future work recommendations in section 7.

2 Research Goals

This research explores the topic of testing the robustness of music processing pipelines with inputs affected by musically meaningful transformations. Two classes of such transformations can be identified: supposedly relevant and irrelevant ones to the pipeline's objective. For example, adding volume is relevant for calculating dynamic complexity [12] but not for calculating tempo. The topic then poses a few main assumptions: *(i)* inputs having relevant transformations affect pipelines in a correct way and *(ii)* inputs having irrelevant transformations do not affect robust pipelines. In other words, the more robust the pipeline, the more invariant it is to the (irrelevant) transformations [8]. For instance, MusiCNN [13] audio tagging pipeline has an enforced pitch invariance in its inner workings and therefore does not get influenced by inputs affected by pitch shifts. Conversely, if an input having a relevant transformation is created for the purpose of testing, a robust pipeline will produce a *correct* output.

Musical beat, being the "basic rhythmic unit of a measure" [14], is one of the most important building blocks of a musical piece. In digital audio, beat is one the features which "encapsulates most of the meaningful information of an audio track" [15, p.2]. Nowadays beat detection is used in various ways and applications, such as video editing[2], rhythm games[3], synchronization with other media[4] and other MIR tasks: automated rhythm transcription [16], chord extraction [17] and music similarity [18]. Beat extraction is of high importance in MIR research and industry and therefore it is important to test the robustness of these pipelines as failing pipelines might propagate to the other parts of applications and make overall performance of them not satisfiable.

This research will explore the assumption *(i)* and will answer the question *how tempo input transformations affect Essentia's beat extraction algorithms?* This paper will focus on testing an algorithm provided by *Essentia*, namely *RhythmExtractor2013*[5]. It is an an open-source library for music analysis which nowadays is being used in industry[6] and offline research to extract various information about audio clips [19]. Two different methods of *RhythmExtractor2013* will be tested: *multifeature*[7], which was proposed by Zapata et al. [20] and *degara*[8], proposed by Degara et al. [21]. The research will try to answer the main question with the help of the following sub-questions: what musically meaningful transformations can be applied to inputs, what data can be used to test the extractor, what measure can be used to test the accuracy of the extractor, how does the accuracy change with regards to the musical transformations, what can be deduced from the results and what are their causes.

3 Methodology and Setup

This section will describe the methodology and setup of the experiment: what transformations are applied to the test samples, what data is used to test the *RhythmExtractor2013* and what evaluation strategy is applied to the results.

3.1 Musical Transformations

To understand what a musically meaningful transformations is, we can break down the term into two parts, namely, how an audio can be transformed and what is the musical meaning of it. The purpose of such distinction is to identify transformations which "accomplish a musically meaningful effect" [22, p.109]. As an example, Table 1 summarizes some of the possible transformations.

The whole experiment consists of four tempo transformations:

1. Tempo change of the whole clip. A clip is time-stretched by every value from the interval [0.5; 2.0] with steps of 0.01.

[1] Essentia, https://essentia.upf.edu/

[2] Filmora, https://filmora.wondershare.com/get-creative/edit-video-to-beat.html

[3] Beat Saber, https://beatsaber.com/; Bemuse, https://bemuse.ninja/; Bilter Fubble, https://test-bilter-fubble.herokuapp.com

[4] BeatSync, http://www.beatsynclights.com/

[5] Documentation available here: https://essentia.upf.edu/reference/std_RhythmExtractor2013.html

[6] List of companies using Essentia: https://essentia.upf.edu/applications.html

[7] Multifeature, https://essentia.upf.edu/reference/std_BeatTrackerMultiFeature.html

[8] Degara, https://essentia.upf.edu/reference/std_BeatTrackerDegara.html

2. Random changes in tempo throughout the audio in the interval [0.5; 2.0] of the original speed. The audio is cropped at every time stamp provided by the ground truths. The amount of random shifts in a clip is equal to the amount of time intervals in the ground truths.
3. Incremental tempo change throughout the song in the interval [1.0; 2.0] of the original speed. The clips are cropped at every time stamp provided by the ground truths. The steps of the tempo increase are equal to the interval size (2.0 - 1.0 = 1.0) divided by the amount of resulting clips.
4. Decremental tempo change throughout the song in the decreasing interval [1.0; 0.5]. The clips are cropped at every time stamp provided by the ground truths. The steps of the tempo decrease are equal to the interval size (1.0 - 0.5 = 0.5) divided by the amount of resulting clips.

In the rest of the paper, these transformations will be referred to as experiments #1 to #4. The transformations themselves are performed with the help of the *PyRubberband*[9], a Python wrapper, widely used by the community for audio stretching.

Transformation (effect)	**Musical meaning**
Pitch shifting	Modulation, transposition
Time stretching	Different BPM/tempo, *ritenuto*[10], *accelerando*[11]
Tremolo	Vibrato
Chorus	Number of voices
Echo, reverb	Performance acoustics
Gain	Dynamics

Table 1: Possible transformations and their musical meanings.

3.2 Dataset

To be able to perform the experiments, a special dataset was deliberately created. It includes samples from four open-source datasets, namely GTZAN [23], Ballroom [24], SMC_MIREX [25] and MDB Drums [26] and a new dataset was created (visualised in Figure 1). In total, the new dataset contains 364 audio clips split into two main categories. The first category contains samples which have a steady, regular rhythm, that allows for "easy" beat tracking. The second category has the opposite: 218 audio clips which have an unsteady rhythm and therefore pose difficulties for beat tracking algorithms. The existing research on the evaluation of beat tracking algorithms shows that it is important to have not only easy to track samples but also challenging ones, as

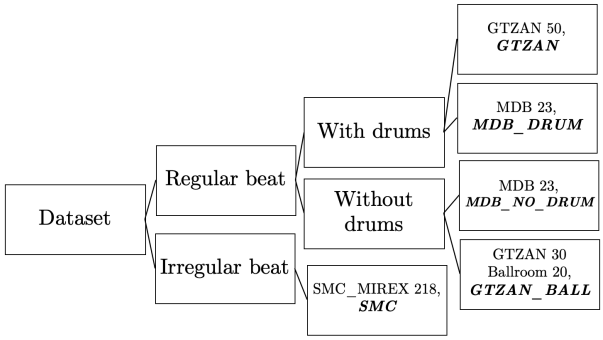

Figure 1: Visualization of the new dataset, the splits, the number of samples in each split and their abbreviations for references in later section.

is done, for instance, in [27]. Zapata et al. [28] suggests that testing pipelines on a dataset where the majority of the samples fall under the "easier songs" category might produce a result which is "optimistic" and does not reflect the general performance when applied to all music genres. This idea is reinforced by Davies and Böck [29], where they indicate that it can be difficult to understand the true performance of the pipeline if non-supervised[12] data sampling is done, where the proportion of non-trivial audio clips is unknown. In [30] it is also claimed, that "easy" samples are not enough to test the beat tracking pipelines as they do not provide tempo changes, which are important part of the research aimed at beat tracking algorithms.

The "difficult" (unstable rhythm) category contains audio samples taken from the SMC_MIREX dataset. The subset includes recordings of solo instrumental pieces, songs and orchestral works which have a great deal of rhythmical freedom and depend on a performer's interpretation. The SMC_MIREX was made to "add diversity to existing collections" [25, p.5] and is used as a "difficult" set in [31] and [29]. Therefore, it is also suitable for this experiment. The "easy" (stable rhythm) category contains songs taken from GTZAN, Ballroom and MDB Drums datasets. Chiu et al. [32] found that the best performance of the their proposed beat tracking model was achieved with samples which have drums in them. It is additionally suggested, that, because of this, "tailored trackers for percussive and non-percussive sounds" [32, p.4] would be beneficial to build. For this reason, the "easy" category is further split into two subcategories: one containing samples where beats are clearly expressed by drums and the other one containing samples where beats are mostly felt with harmonic changes and drums are not present. The exception is MDB Drums dataset, which, in addition to full mixes, provides separate audio stems. Having this extra freedom, 23 mixes without drums were produced and are used in tests against full mixes. The distinction between

[9] PyRubberband, https://github.com/bmcfee/pyrubberband

[10] Musical term for slowing down.

[11] Musical term for speeding up.

[12] In this case random, not checked manually, from untrustworthy sources.

drum and drumless samples allows to investigate whether the overall performance of the beat tracking pipelines on "easy" samples is affected by the presence of percussive instruments.

The regular beat subset was composed with the help of the *Mean Mutual Agreement* (MMA) score calculated in [28]. As an indication to how difficult it is for an algorithm to calculate beats of a song, MMA ranks musical genres from the most difficult (lowest score) to the easiest (highest score) ones. It is important to mention, that although MMA score is based on music genres, genre, as a metric, is disregarded in the experiment. However, the samples from the datasets were taken according to their music genre labels, namely, the set with drums includes disco, pop and hip-hop from GTZAN and the set without drums contains classical from GTZAN and waltz (which could be considered as a subgenre of classical) from Ballroom. MDB Drums dataset includes samples form a range of genres too and hence poses more real-world challenges. Nevertheless, in the beat tracking applications mentioned previously, these systems come early in the pipeline (e.g, chord extraction) or are expected to produce an output independently of genre (light synchronization), so genre labels were used only as a helping tool to create a new dataset. Even though "classical" falls below the MMA threshold of 1.5, [28] does not disclose what kind of samples were used for the experiment; hence it is difficult to judge whether the score was affected by the presence of rhythmical freedom (e.g, rubato) in the clips. To eliminate this doubt, only handpicked (by manual listening) "classical" clips with steady rhythm are used in this dataset.

The beat annotations for GTZAN samples were taken from [33], for SMC_MIREX - from [25], MDB Drums - from [26] and for Ballroom - from [34]. As far as MIR community is concerned, the validity of ground truth is a topic of discussions itself (e.g., [7]) as annotations made even by domain experts can be subject to inaccuracies and biases [35]. However, such ground truths with annotations are currently widely used in evaluation strategies and provide the most robust way of testing pipelines and so are trusted and used in this research too.

3.3 Evaluation

Even though in [28] it was found that *multifeature* method is more "accurate" (with regards to their proposed evaluation strategy) and *degara* runs much faster, this research will disregard the run-time of these pipelines because the purpose of the experiment is to evaluate the robustness of these methods towards musical transformations. From the beat tracking evaluation strategies summarized in [30], F-measure (or F-score; a measure of a test's accuracy) was chosen as it is widely used to evaluate not only the beat tracking systems but information retrieval systems in general. The measure is calculated from two metrics, precision and recall. The former metric corresponds to a fraction of how many beats were identified correctly from all of the beats calculated by the algorithm and the latter metric corresponds to the fraction of how many beats were correctly identified from the beats which are in the ground truths.

For humans too it is not all the time easy to exactly identify where a beat is. Time stamps need to be discretized from the continuous time domain and that brings its own challenges. Whilst producing ground truths, domain experts use various technologies, such as Sonic Visualizer [13], to try to extract exact times when ticks appear. That does not mean, however, that a beat time stamp extracted by an algorithm is not correct if it is not exactly equal to the ground truth. The results of [36] show, that when people are asked to tap in synchrony with a click, in general, the taps precede the click by $30 - 50$ ms. This means, that there always is a window in which if a beat falls it can perceptually still be considered correct. In this research two sizes of windows are considered: 70 ms, as chosen in [37] for beat tracker evaluation, and 40 ms (the interval average from [36]) as 70 ms tapping delay for a musical ear can already seem big and sound 'off time'.

4 Result Analysis

This section presents results acquired from the experiments described in section 3.

4.1 Experiment #1

The transformation in experiment #1 changes the tempo of the whole audio clip. Table 2 presents the means and standard deviations of F-measures per song subset across all time stretch values ([0.5; 2.0]) calculated with *multifeature* method and using 70 ms window:

Dataset	mean	SD
GTZAN	0.754	0.111
MDB_DRUM	0.732	0.121
MDB_NO_DRUM	0.526	0.075
GTZAN_BALL	0.509	0.043
SMC	0.374	0.065

Table 2: Experiment #1 results

The overall resulting scores across the experiments show a tendency where beats are recognized better from the samples with drums in them. For example, Figure 2 and Figure 3 show the results of the experiment #1 on the MDB dataset, where beats were calculated on samples with and without the drum stems respectively. The blue line indicates the average F-measure across all samples, the red dashed line is a fitted curve on the average, the light blue filling represents the standard deviation of the F-measure and the three remaining lines are plots of randomly picked result instances (the names of the audio samples are indicated in the legend). It can be seen from the plots, that the average F-measure is higher and the standard deviation is smaller of the samples with drums in them. Experiment #1 on GTZAN dataset with drums

[13] Sonic Visualizer, https://www.sonicvisualiser.org/

(Figure 4) provides somewhat similar results seen in Figure 2 and results of the GTZAN_BALL subset (Figure 5) show similar tendencies to Figure 3. Two different collections of the same type showing similar results reinforce the assumption that *RhythmExtractor2013* performs better on audio clips with drums in them with regards to the uniform tempo changes. The tracks which have most unstable curves in the plots also include amplified guitars, present in rock and metal genres, a lot of extra off-beat percussive noises, present in electronic music, are not of very high quality or have reverb, which leads to the distinction among noise and beats harder to identify.

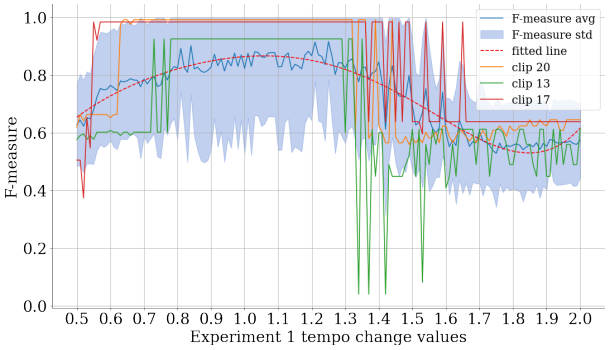

Figure 2: MDB_DRUM, multifeature, 70 ms

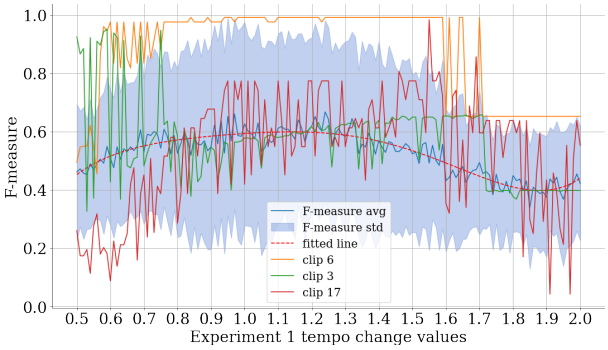

Figure 3: MDB_NO_DRUM, multifeature, 70 ms

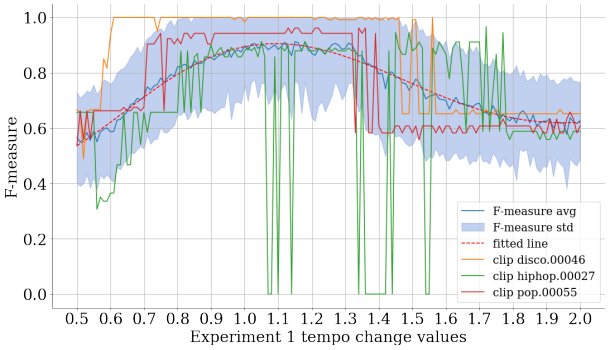

Figure 4: GTZAN dataset, multifeature, 70 ms

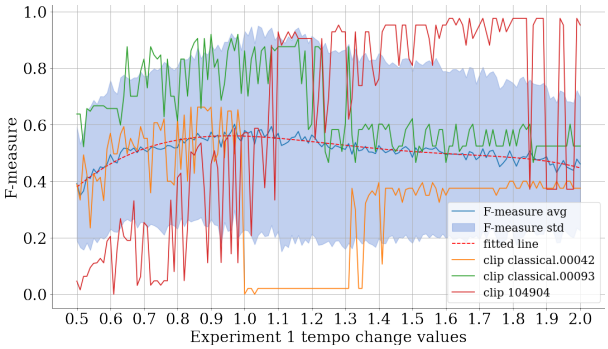

Figure 5: GTZAN_BALL, multifeature, 70 ms

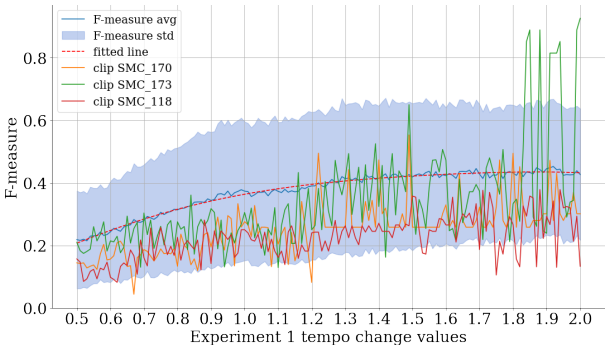

Figure 6: SMC dataset, multifeature, 70 ms

Results of songs with irregular rhythm are not optimistic. Figure 6 shows the plot of the experiment #1 on SMC dataset. It can be observed, that the average F-measure scores are relatively low compared to the previous figures, with untransformed audios ($\times 1.0$ speed) scoring on average $\approx$0.373, compared to $\approx$0.823 of MDB_DRUM, $\approx$0.627 of MDB_NO_DRUM, $\approx$0.897 of GTZAN and $\approx$0.579 of GTZAN_BALL.

4.2 Experiment #2

The transformation in experiment #2 makes many random tempo changes throughout the audio clip. Table 3 presents the means and standard deviations of F-measures per song subset calculated with *multifeature* method and using 70 ms window:

Dataset	mean	SD
GTZAN	0.433	0.101
MDB_DRUM	0.425	0.095
GTZAN_BALL	0.408	0.084
MDB_NO_DRUM	0.385	0.073
SMC	0.342	0.085

Table 3: Experiment #2 results

During this experiment the clips were randomly sped up or slowed down at various time stamps. That, as a consequence, increased their rhythmical irregularity. Results show, that all of the 'regular beat' subsets suffered

from this transformation and the accuracy of all subsets dropped significantly, almost 2 times in some cases compared to the scores of untransformed sets. Nevertheless, subsets with clips with drums still scored the highest, tendency present also in experiment #1.

4.3 Experiment #3

The transformation in experiment #3 increments the tempo throughout the song. Table 4 shows the means and standard deviations of F-measures per song subset calculated with *multifeature* method and using 70 ms window:

Dataset	mean	SD
GTZAN	0.832	0.137
MDB_DRUM	0.802	0.137
MDB_NO_DRUM	0.742	0.166
GTZAN_BALL	0.742	0.178
SMC	0.573	0.218

Table 4: Experiment #3 results

The results of the experiment #3 show that subtle additions of rhythmical instability do not drastically affect the scores, although it can negatively impact them, as can be seen for GTZAN dataset. The growth in scores (for SMC, GTZAN_BALL and MDB_NO_DRUM), compared to the scores of untransformed audios, could potentially be explained by the window size. When a song becomes fast, the beats can come very close to one another. Consecutively, when a window size is applied on ground truths during the calculations, the intervals might overlap. Falsely detected beats now have a higher chance to fall into one of those intervals and this leads to increase in true positives and decrease in false positives, which has a positive impact when calculating the F-measure. Finally, datasets with clips with drums in them scored the highest, as in both previous experiments.

4.4 Experiment #4

The transformation in experiment #4 decrements the tempo throughout the song. Table 5 displays the means and standard deviations of F-measures per song subset calculated with *multifeature* method and using 70 ms window:

Dataset	mean	SD
MDB_DRUM	0.778	0.163
MDB_NO_DRUM	0.706	0.190
GTZAN	0.688	0.176
GTZAN_BALL	0.646	0.230
SMC	0.400	0.181

Table 5: Experiment #4 results

Similarly to the results of the experiment #1, slowing down the songs and increasing rhythmical instability had the biggest impact on GTZAN set. The audio quality of the songs in the GTZAN dataset is originally somewhat poor and the experiment degraded it further. The process of clipping and stitching the songs has produced audible pops which could have influenced the beat tracker. On the other hand, the increase in scores for the MDB_NO_DRUM and GTZAN_BALL sets can also be related to the same reason. The sometimes audible on-beat pops could have helped the beat tracker to identify more true positives. Nonetheless, as with the other experiments, a set with drums (MDB) has scored the highest.

4.5 Window size

For the experiments #2, #3 and #4, result changes were observed when scores were calculated with a window size of 40 ms (*multifeature*). The pairs of the means and standard deviations of deltas (magnitudes only) across all datasets for these experiments turned out to be (0.061, 0.012), (0.037, 0.010) and (0.070, 0.044) respectively.

For the experiment #1, changing the window size made an observable difference too. Figure 7 shows the plot of the experiment #1 on the SMC dataset with 40 ms window. By comparing it to the results of calculations with the window size of 70 ms (Figure 6), it was found that the mean and standard deviation of deltas (magnitudes only) are 0.099 and 0.010 respectively. The results of the rest of the datasets are given in Table 6.

Dataset	mean	SD
GTZAN_BALL	0.117	0.036
SMC	0.099	0.010
GTZAN	0.077	0.070
MDB_NO_DRUM	0.075	0.017
MDB_DRUM	0.032	0.017

Table 6: Window size change effect on experiment #1, multifeature

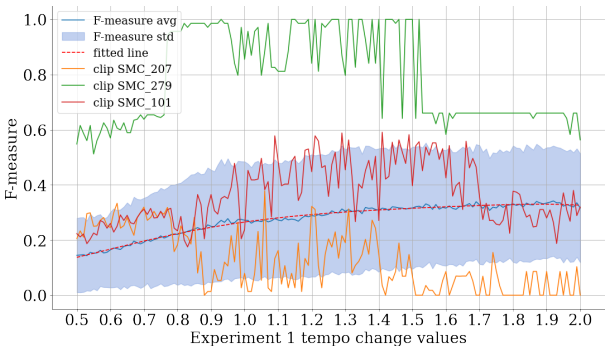

Figure 7: SMC dataset, multifeature, 40 ms

The results show that the window size is a crucial parameter in accuracy calculations and can affect the score by up to ±12% with high probability.

4.6 Parameter *method*

No significant changes were observed when comparing the results of *multifeature* and *degara* methods. The re-

sults show that, on average, not even 1% difference can be observed between the methods. Table 7 summarizes the means and standard deviations of deltas across all datasets per experiment:

Experiment	mean	SD
#1	0.005	0.027
#2	0.008	0.013
#3	0.002	0.014
#4	0.005	0.018

Table 7: Absolute values of means and standard deviations of deltas, 70 ms

5 Discussion and Limitations

Even though the results give insight into how *Essentia's* beat tracking algorithms perform on various music clips, they inevitably lack the ability to generalize perfectly. The experiments collectively show a high standard deviation, which indicates that the results might not be so reliable after all. This is the outcome due to a number of reasons. Firstly, the datasets used are quite small. Such experiments require a lot of samples in order to be able to draw very general conclusions, but only a fraction of the world's music is open-source and publicly available, let alone the annotated clips. Secondly, the dataset size led to an imperfect split. For instance, the MDB set contains only 23 samples of a variety of genres. Some of them, metal, jazz, fall below the MMA score of 1.5 despite them having a steady rhythm. On the other hand, some samples in SMC set have drums in them even though their rhythm is unstable. Overall, more distinctive subsets could be made, but that would result in a very few samples in each of them.

Apart from that, transformations affect the audio quality. When listening back to the transformed audios, some anomalies can be heard. When the songs are sped up by large magnitudes, incidental pops and distortions appear. These then can be falsely detected by the algorithm and influence the result. Moreover, when the songs are slowed down, the sound events become artificially stretched. This results in unwanted additional sounds and notes being prolonged, e.g., a simple quick sounding snare drum hit now becomes a longer, *tenuto*[14] like noisy sound. Beats themselves, therefore, are stretched and might not be detected due to their length. Figure 8 visualises how a beat is affected by a time stretch as time progresses.

Several other properties have an impact on the results. Firstly, having percussive sounds is advantageous for a beat tracker. It is clear that *RhythmExtractor2013* is more accurate with more confidence on samples with drums in them. Interestingly, this result is observed not only on the original audios, but also on the transformed ones, which reinforces the conclusion made in [32] and reveals a gap in the capabilities of the algorithm. Moreover, rhythm

[14] A musical term for sustaining the note for its whole length.

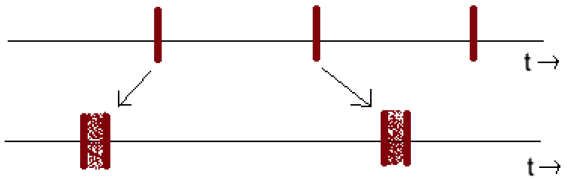

Figure 8: Beat sounds affected by time stretch. Top line: original audio, bottom line: slowed down audio

stability affects the beat tracking accuracy too. The results show that *RhythmExtractor2013* performed worst across all experiments on the SMC dataset, which contains songs with irregular rhythm. Finally, presence of timbral qualities similar to those of percussive sounds, e.g, amplified guitars, electronic effects, have negative influence. This should be taken into account when creating test sets.

6 Responsible Research

The research was conducted with respect to the principles defined in the Netherlands Code of Conduct for Research Integrity [38]. **Honesty** is preserved by objectively discussing the results and providing clear explanations about their validity. **Scrupulousness** is preserved by using scientific methods to conduct the research and experiments from start to end. The data for the experiment was carefully selected according to its design but it was in no ways fabricated or manipulated. However, the data has been acquired from a "secondary" source. This has been **transparently** indicated in the previous chapters and the implications of that have been discussed (sections 3.3.22-24 of the conduct). The research is, in essence, **independent** as it has no influence from non-scholarly, such as political or commercial, sources. The author of the paper claims full **responsibility** if plagiarism or conflict of interest is found.

The experiment can be easily replicated and work can be continued on it as data and the code can be acquired from *4TU.ResearchData*[15] (3.3.25). In addition, the paper has all of the references of papers from which inspiration for ideas was taken or statements were reinforced (3.4.29). A potential bias on the results can be identified in the lack of diversity of the music samples used for the experiments. Unfortunately, due to the legal matters of music distribution, only free, publicly available samples could be used. Besides, due to the limited availability of the ground truth annotations, which were an integral part of the whole research, the datasets stayed unavoidably small.

7 Conclusions and Future Work

This paper has presented the evaluation of *Essentia's RhythmExtractor2013* by conducting an experiment in

[15] The repository identifier (DOI): https://doi.org/10.4121/18274532

which tempo of audio samples was affected in various ways. The results across the experiments unanimously show, that the comfort zone of the *RhythmExtractor2013* contains samples which have steady rhythm and drums in them, steady rhythm, in fact, having more weight. However, whilst analysing the scores, a number of obstacles have been identified. First of all, the datasets used for the experiments are relatively small. More samples should be used to achieve more general and better results. The research has shown, that there is a need for collecting a large and very diverse dataset with beat annotations. Moreover, other parameters - genre of the song, window size during the calculations - can affect the results too.

Moreover, striving for the realistically most flawless evaluation system, several aspects need to be considered to a greater extent: identifying atomic properties of songs (percussion, rhythm, frequency range, instrumentation, etc.), grouping songs by these properties to create distinct subsets and understanding how third-party libraries can affect musical transformations (e.g., how the audio samples are time stretched, what parameters can be passed to algorithms to get the most realistic stretched version). To abide by Richard Rogers' thought, that 'there is no one-size-fits-all solution'[16], there could, perhaps, be separate beat trackers which individually are superior to each other with regards to these properties and could be used together to achieve the best results.

In addition to open-source, licence free libraries, such as *Essentia*, or crowd-sourced public datasets, such as AcousticBrainz, there is at least one more service which provides its users with audio features, namely Spotify Web API[17]. Developer terms and conditions[18] do not specify limitations on its academic usage[19], therefore it could be a valuable helping tool in evaluating other beat tracking systems. Before the API could be incorporated in any research, it is firstly necessary to discern how trustworthy and accurate the beat data is. If reliable, the API could potentially contribute to filling the gap of limited amount of ground truth annotations for beats. With this, also more musically diverse test datasets could be collected.

As the research reveals, *RhythmExtractor2013* is, to a certain degree, led astray by the tempo transformations. *Essentia* has more audio feature extractors which can also be confronted with various musically transformed audio samples. In addition to showcasing the results of this research, the paper also aims to spark the readers with creativity in designing evaluation systems for music analysis pipelines.

[16]Taken from https://www.brainyquote.com/quotes/richard_rogers_613230

[17]Documentation for audio features: https://developer.spotify.com/documentation/web-api/reference/#/operations/get-audio-analysis

[18]Developer terms and conditions: https://developer.spotify.com/terms/

[19]The paper does not claim responsibility for the legal usage of the API, this should be further investigated.

References

[1] Joe Futrelle and J. Stephen Downie. Interdisciplinary research issues in music information retrieval: Ismir 2000–2002. *International Journal of Phytoremediation*, 21(1):121–131, January 2003.

[2] Spotify. Get perfect song recommendations in the playlists you create with enhance, 2021. Accessed on: Nov. 8, 2021. [Online]. Available: https://newsroom.spotify.com/2021-09-09/get-perfect-song-recommendations-in-the-playlists-you-create-with-enhance/#:~:text=Here's%20how%20it%20works%3A,a%20max%20of%2030%20recommendations.

[3] Avery Li chun Wang and Th Floor Block F. An industrial-strength audio search algorithm. In *Proceedings of the 4 th International Conference on Music Information Retrieval*, 2003.

[4] Sony (Sony R&D). Audio & acoustics, 2021. Accessed on: Nov. 8, 2021. [Online]. Available: https://www.sony.com/en/SonyInfo/technology/about/rdc/tech-portfolio/audio-acoustics/.

[5] E. Gómez M. Schedl and J. Urbano. Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, 8(2-3):127–261, 2014.

[6] Alastair Porter, Dmitry Bogdanov, Robert Kaye, Roman Tsukanov, and Xavier Serra. Acousticbrainz: a community platform for gathering music information obtained from audio. In *16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, pages 786–792, Malaga, Spain, 26/10/2015 2015.

[7] C.C.S. Liem and C. Mostert. Can't trust the feeling?: How open data reveals unexpected behavior of high-level music descriptors. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, pages 240–247, 2020. Virtual/online event due to COVID-19 ; 21st International Society for Music Information Retrieval Conference, ISMIR 2020 ; Conference date: 11-10-2020 Through 15-10-2020.

[8] Janne Spijkervet Minz Won and Keunwoo Choi. *Music Classification: Beyond Supervised Learning, Towards Real-world Applications*. https://music-classification.github.io/tutorial, Nov. 2021.

[9] Corey Kereliuk, Bob L. Sturm, and Jan Larsen. Deep learning and music adversaries. *IEEE Transactions on Multimedia*, 17(11):2059–2071, 2015.

[10] Bob L. Sturm. A simple method to determine if a music information retrieval system is a "horse". *IEEE Transactions on Multimedia*, 16(6):1636–1644, 2014.

[11] Jaehun Kim, Julián Urbano, Cynthia C.S. Liem, and Alan Hanjalic. Are nearby neighbors relatives?: Testing deep music embeddings. *Frontiers in Applied Mathematics an Statistics*, 5:1–17, 2019.

[12] S. Streich. *Music Complexity a multi-faceted description of audio content*. PhD thesis, Universitat Pompeu Fabra, 2007.

[13] Jordi Pons and Xavier Serra. musicnn: Pre-trained convolutional neural networks for music audio tagging, 2019.

[14] The Editors of Encyclopaedia Britannica. beat, 1998. Accessed: Nov. 21, 2021. [Online]. Available: https://www.britannica.com/art/beat-music.

[15] A. Kratimenos, K. Avramidis, C. Garoufis, A. Zlatintsi, and P. Maragos. Augmentation methods on monophonic audio for instrument classification in polyphonic music. In *European Signal Processing Conference*, volume 2021-January, pages 156–160, 2021.

[16] Christopher Raphael. Automated rhythm transcription. In *ISMIR 2001, 2nd International Symposium on Music Information Retrieval, Indiana University, Bloomington, Indiana, USA, October 15-17, 2001, Proceedings*, 2001.

[17] Juan P. Bello and Jeremy Pickens. A robust mid-level representation for harmonic content in music signals. In *ISMIR 2005 - 6th International Conference on Music Information Retrieval*, ISMIR 2005 - 6th International Conference on Music Information Retrieval, pages 304–311, 2005. 6th International Conference on Music Information Retrieval, ISMIR 2005 ; Conference date: 11-09-2005 Through 15-09-2005.

[18] Daniel P. W. Ellis, Courtenay V. Cotton, and Michael I. Mandel. Cross-correlation of beat-synchronous representations for music similarity. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 57–60, 2008.

[19] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, O. Mayor, Gerard Roma, Justin Salamon, J. R. Zapata, and Xavier Serra. Essentia: an audio analysis library for music information retrieval. In *International Society for Music Information Retrieval Conference (ISMIR'13)*, pages 493–498, Curitiba, Brazil, 04/11/2013 2013.

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

[21] Norberto Degara, Enrique Argones Rua, Antonio Pena, Soledad Torres-Guijarro, Matthew E. P. Davies, and Mark D. Plumbley. Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):290–301, 2012.

[22] Xavier Amatriain, Jordi Bonada, [Agrave]lex Loscos, Josep Lluís Arcos, and Vincent Verfaille. Content-based transformations. *Journal of New Music Research*, 32(1):95–114, 2003.

[23] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[24] Fabien Gouyon, Anssi Klapuri, Simon Dixon, Miguel Alonso, George Tzanetakis, Christian Uhle, and Pedro Cano. An experimental comparison of audio tempo induction algorithms. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14:1832 – 1844, 10 2006.

[25] Andre Holzapfel, Matthew Davies, Jose Zapata, João Oliveira, and Fabien Gouyon. Selective sampling for beat tracking evaluation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20:2539–2548, 11 2012.

[26] Carl Southall, Chih-Wei Wu, Alexander Lerch, and Jason Hockman. Mdb drums – an annotated subset of medleydb for automatic drum transcription. 10 2017.

[27] C. N. Ramadhani, E. Suryawati Ningrum, and Z. Darojah. Music signals beat tracking based on bidirectional long-short term memory. In *IES 2019 - International Electronics Symposium: The Role of Techno-Intelligence in Creating an Open Energy System Towards Energy Democracy, Proceedings*, pages 485–489, 2019.

[28] J. R. Zapata, Andre Holzapfel, M.E.P Davies, J. Oliveira, and F. Gouyon. Assigning a confidence threshold on automatic beat annotation in large datasets. In *13th International Society for Music Information Retrieval Conference*, pages 157–162, Porto, Portugal, 08/10/2012 2012.

[29] Matthew E. P. Davies and Sebastian Böck. Evaluating the evaluation measures for beat tracking. In *ISMIR*, 2014.

[30] Matthew Davies, Norberto Degara Quintela, and Mark Plumbley. Evaluation methods for musical audio beat tracking algorithms. 10 2009.

[31] A. S. Pinto, S. Böck, J. S. Cardoso, and M. E. P. Davies. User-driven fine-tuning for beat tracking. *Electronics (Switzerland)*, 10(13), 2021.

[32] C. Chiu, A. W. Y. Su, and Y. Yang. Drum-aware ensemble architecture for improved joint musical beat and downbeat tracking. *IEEE Signal Processing Letters*, 2021.

[33] Magdalena Fuentes Matthew E. P. Davies, Sebastian B ock. *Tempo, Beat and Downbeat Estimation*. https://tempobeatdownbeat.github.io/tutorial/intro.html, Nov. 2021.

[34] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algo-

rithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.

[35] Olof Misgeld, Torbjörn L Gulz, Jūra Miniotaitė, and Andre Holzapfel. A case study of deep enculturation and sensorimotor synchronization to real music. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 460–467, Online, November 2021. ISMIR.

[36] Gisa Aschersleben and Wolfgang Prinz. Aschersleben, g. & prinz, w. synchronizing actions with events: the role of sensory information. percept. psychophys. 57, 305-317. *Perception & psychophysics*, 57:305–17, 05 1995.

[37] Simon Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58, 2001.

[38] KNAW, NFU, NWO, TO2-federatie, Vereniging Hogescholen, and VSNU. Netherlands code of conduct for research integrity (2018), 2018.