University of Derby

Department of Electronics, Computing & Mathematics

A project completed as part of the requirements for

BSc (Hons) Computer Games Programming

Entitled

Audio Beat Tracking: An Analysis of Beat Tracker Accuracy in Audio Streaming Scenarios

By

Ben Strutt

April 2017

Abstract

This study looked at the effects of audio streams on the accuracy of several beat trackers. Past research on beat tracking typically tested beat trackers on individual audio clips in isolation. This doesn’t evaluate beat tracker accuracy on streams of concatenated audio clips. This study took the publicly available Essentia Multifeature Beat Tracker, INESC-Porto Beat Tracker, Madmom Beat Tracker, Madmom Beat Detector, DBN Beat Tracker, and CRF Beat Detector and evaluated their accuracy in streaming and non-streaming scenarios. The Ballroom, GTZAN, and SMC audio datasets were used as a source of beat annotated audio files for both streaming and non-streaming tests. Comparing accuracy in streaming and non-streaming scenarios showed that all beat trackers had a lower accuracy on audio streams than on individual audio clips. The INESC-Porto Beat Tracker performed the worst in both scenarios, whilst the DBN Beat Tracker performed the best. The Madmom Beat Detector and CRF Beat Detector performed well in the non-streaming scenarios but their accuracy dropped significantly in the streaming scenario. Looking at beat tracker accuracy over time showed the INESC-Porto Beat Tracker and Madmom Beat Detector steadily lost accuracy over time during audio streams, whilst the other beat trackers maintained roughly constant accuracy. Comparing the DBN Beat Tracker to the Madmom Beat Tracker showed that the improved Dynamic Bayesian Network of the DBN Beat Tracker improved its accuracy in both streaming and non-streaming scenarios. Comparing the CRF Beat Detector to the Madmom Beat Detector showed that the CRF Beat Detector’s probabilistic post processing step improved accuracy significantly in streaming scenarios.

Acknowledgements

Many thanks to Dave Voorhis for his assistance throughout the course of this study.

Table of Contents

1. Introduction

In 2008 Shazam released its signature app on the app store, which could identify songs from a simple audio recording. In 2014 the company announced that its app had been used to identify 15 billion songs since its release (Nguyen, 2014). In the same year as Shazam's release, the game Audiosurf was released on the online videogame store Steam (Store.steampowered.com, 2008). This game could scan any audio file on a player's computer to identify its tempo/beat. It would then create a racetrack that matched the pacing of the song, allowing players to experience their music library in new ways. Audiosurf was released in February 2008 and became the bestselling game on Steam that month (Bramwell, 2008). Both of these highly popular pieces of software were only possible due to research into Music Information Retrieval, specifically in the areas of audio fingerprinting (Shazam.com, 2016) and frequency analysis respectively (Wilburn, 2008).

Music Information Retrieval, or MIR, is an area of research that focuses on algorithmically extracting information from audio files. This includes basic musical information such as extracting a song's tempo, key, or beat, as well as more complex analyses that attempt to extract the base melody, chords, or vocals from an audio file. Research in various areas have led to the creation of several useful tools such as the previously mentioned Shazam app. Other applications have included chord estimation (Mauch and Dixon, 2010), automatic music transcription (Bello-Correa, 2003), and enhanced videogaming by allowing the player's music library to directly influence the game, as in Audiosurf (Fitterer, 2008), Melody's Escape (Dansart, 2016), and Crypt of the Necrodancer (Clark and Wilson, 2015)).

1.1. Project Rationale

MIR is a very broad topic, covering many different potential areas of research. The subset of MIR research that this study focused on was that of beat tracking, a task that is the equivalent of a human tapping or nodding their head to the beat. Whilst humans find this a very simple, almost trivial task, the same cannot be said for computers. Research into beat tracking has been going on at least since 1990 with (Allen and Dannenberg, 1990), and may have started even earlier in 1976 with (Longuet-Higgins, 1976). Despite its long research history, beat tracking is still far from a solved problem. This is evidenced by the yearly Music Information

Retrieval Evaluation eXchange (aka MIREX) running a beat tracking task to test accuracy of recently developed beat trackers (Music-ir.org, 2016).

Looking at more recent research into beat trackers has shown that there are many different techniques that have been used when extracting beats. These include the use of multi-agent techniques (Zapata, Davies and Gomez, 2014), tempo induction using comb filters (Böck, Krebs and Widmer, 2015), and Dynamic Bayesian Networks (Böck, Krebs and Widmer, 2014) just to name a few. Regardless of underlying method, most beat trackers were evaluated by examining their accuracy on individual audio clips. Whilst this provides general information on a beat tracker's effectiveness, it doesn't give any indication of how it well it can handle a stream of sequential audio clips. This limitation of standard beat tracker tests has been avoided in the past through the use of "*stream testing*". Stream testing has previously been performed in (Oliveira et al., 2012) and (Collins, 2006) and involves running a given beat tracker on a single audio file that contains several back to back audio clips. The intent in (Oliveira et al., 2012) was to use stream testing to measure how well the INESC-Porto Beat Tracker handles song transitions. However, this form of testing could also be useful to obtain general information about how a beat tracker performs in streaming scenarios.

1.2. Project Aim and Objectives

The primary aim of this study was to see how accurate currently available beat trackers are when operating on audio streams. A secondary aim was to see if any beat trackers are currently suitable for extracting beats from audio streams. The final aim was to observe how audio streams can affect a beat tracker's accuracy. To achieve these aims several tasks were completed. First a set of beat trackers were obtained, alongside several datasets containing audio clips with annotated beat times. Then a set of stream files were created from the datasets to allow the beat trackers to be tested in a streaming scenario. Finally, the beat trackers were run on the audio clips and audio streams, and their outputs were evaluated to determine the beat tracker's accuracy.

The beat trackers were obtained by looking at the last six years of submissions to the MIREX Beat Tracking task. From those submissions six publicly available beat trackers were selected for further testing. The chosen beat trackers were the INESC-Porto Beat Tracker (Oliveira et al., 2012), the Essentia MultiFeature Beat Tracker (Zapata, Davies and Gomez, 2014), the Madmom Beat Tracker (Böck, 2016), the DBN Beat Tracker (Böck and Krebs, 2016), the

Madmom Beat Detector (Böck, 2016), and the CRF Beat Detector (Böck and Korzeniowski, 2016). These beat trackers were used to extract beats from audio clips in the GTZAN, Ballroom, and SMC audio datasets. The accuracy of the extracted beats was determined by evaluating them against the dataset's annotated beat times. A set of streaming audio files were then created from the audio clips provided by the datasets by concatenating groups of five randomly selected audio clips into a single audio stream. The beat trackers were then run on these stream files and evaluated to determine their accuracy in streaming scenarios. The streaming and non-streaming accuracies were then compared to determine how the streaming scenario affected each beat tracker. The accuracies were also used to compare the beat trackers to each other and determine which would be best suited to extracting beats from an audio stream. The streaming accuracies were also broken down by audio clip position within the stream to see how each beat tracker's accuracy varied over time.

This study is split into the following sections. Section 2 looks at papers describing various beat trackers to get an overview of different beat tracking methods and how they were evaluated. Section 3 describes the beat trackers, datasets, and evaluation methods used in this study. It also details the creation of the audio streams and the process that was followed to determine the accuracy of the beat trackers. Section 4 looks at the results of the study, presenting the accuracies of the beat trackers in various scenarios and examining their significance. It also includes analysis of the significant results, and comparisons between the beat trackers. Section 5 discusses the methodology and results, examining the potential limitations of this study. Section 6 summarises the study, providing a quick breakdown of the previous sections and suggesting areas for further research.

2. Literature Review

2.1. Introduction

In the past, beat trackers took a list of note onsets as inputs, occasionally augmented by pitch information or an initial tempo, rather than a raw audio signal. These note onsets were then used to try and identify the period and phase of the beats (Allen and Dannenberg, 1990). However, more recent beat trackers operate directly on the audio signal itself, typically in .wav format, and attempt to extract audio features, such as the note onsets and initial tempo, internally. This section looks at a variety of contemporary beat trackers, focusing on their implementations and evaluations. This was done to get a broad overview of the different methods of beat tracking that are currently being researched. The discussed beat trackers were found by looking at the last six years of submissions to the MIREX Beat Tracking task, and are prefaced by a quick description of MIREX itself.

2.2. MIREX Submissions

The Music Information Retrieval Evaluation eXchange (MIREX) is an annual event focused on evaluating the effectiveness of various developments in music information retrieval. This event allows participants to submit one or several programmes to a set of tasks, including tempo estimation, melody extraction, beat tracking, and many others (Music-ir.org, 2016). These programmes are then run, their performance at their dedicated task is evaluated, and the results are made publicly available on the MIREX website. This is very useful for looking at the current state of Music Information Retrieval research, as it provides a centralised location where modern solutions to MIR problems are described and evaluated.

Looking specifically at the MIREX Beat Tracking task, submitted beat trackers are expected to be accompanied by a brief pdf describing their implementation (Music-ir.org, 2016). Often these pdfs reference a longer paper which describes the beat tracker in more detail. It is these pdfs and papers that this literature review uses to examine each beat tracker that was submitted to MIREX in the last six years. MIREX also evaluates these beat trackers using three audio datasets, the SMC, MAZ, and MCK datasets (Music-ir.org, 2016). These datasets contain a selection of audio clips accompanied by annotated beat times, and the beat trackers are evaluated by comparing their outputs to these annotations. The comparisons are made using a variety of evaluation measures that are briefly described in section 3.5, and described in detail in (Davies, Degara and Plumbley, 2009).

2.2.1. SB Beat Trackers

The SB beat trackers, officially called BeatDetector and BeatTracker, have been submitted to the MIREX competition by Sebastian Böck every year since 2010. The latest versions were submitted to MIREX 2016 as SB8 and SB9 respectively. A brief description of the beat trackers can be found at (Böck, 2016) and (Böck, 2016), and reference implementations are included in the madmom library (Böck, 2013). Both beat trackers are very similar, being based on the main beat tracker described in (Böck and Schedl, 2011) but using comb filters instead of autocorrelation for tempo induction, as described in (Böck, Krebs and Widmer, 2015). The beat trackers work by passing an audio signal through an onset detection function to obtain three mel spectrograms and the first order differences for each. These are then passed into a recurrent neural network which outputs the probability of a beat occurring in the audio at any given time. This is passed into a bank of comb filters to determine the dominant tempo of the audio, and this tempo is used in conjunction with the RNN output to locate the final beats. The only difference between BeatDetector and BeatTracker is that BeatDetector assumes constant tempo throughout the piece. BeatTracker works slightly differently by only passing the 6 seconds following the last detected beat to the comb filters. The dominant tempo from those 6 seconds is used to estimate the next beat position, and then the process is repeated. This is done until all beats have been found, and it allows BeatTracker to handle varying tempo.

The original paper evaluated the beat trackers by submitting them to the 2010 MIREX beat tracking task (Böck and Schedl, 2011). The results showed that both beat trackers scored quite highly, with BeatTracker performing slightly better than BeatDetector overall. Since the 2016 submission uses a comb filter to estimate tempo rather than the original autocorrelation method, this modification is likely to have affected the beat trackers' performance. The comb filter modification is evaluated in (Böck, Krebs and Widmer, 2015) and shows a performance increase over the original autocorrelation function. Looking at the MIREX 2016 results ((Nema.lis.illinois.edu, 2016), (Nema.lis.illinois.edu, 2016), (Nema.lis.illinois.edu, 2016)) shows that the most recent SB submissions score quite well compared to the other beat trackers, only performing worse than the BK submissions that year.

2.2.2. BK Beat Trackers

The BK beat trackers were submitted by Sebastian Böck, Florian Krebs, Filip Korzeniowski, and Gerhard Widmer in 2014, 2015 and 2016. The main BK submissions were the DBN Beat Tracker and CRF Beat Detector. These two beat trackers are slightly modified versions of the SB submissions, Beat Tracker and Beat Detector. Reference implementations of all four beat trackers are available in the madmom library (Böck et al., 2015), to be found at (Böck, 2013). This section examines the BK 2016 submissions BK1 (DBN Beat Tracker) and BK3 (CRFBeatDetector).

The BK1 submission, described in (Böck and Krebs, 2016), works by taking the beat tracker outlined in (Böck, Krebs and Widmer, 2014), and replacing its Dynamic Bayesian Network with the DBN described in (Krebs, Böck and Widmer, 2015). The resulting beat tracker operates by first processing the audio stream with three parallel short time fourier transforms. The output is then passed to a group of Recurrent Neural Networks, each of which determines the probability of a beat being present at each time frame. Each neural network is specialised on a specific musical style, so that for any audio input there is likely to be a neural network specialised on that input's style. A separate neural network is trained on a variety of styles to create a more general RNN that the other RNNs can be compared to. This general RNN is intended to produce reasonably accurate beats for any audio input, with accuracy only surpassed by the one RNN specialised in that audio's musical style. This way, the correctly specialised RNN is found by choosing the specialised RNN with results closest to those of the general RNN. The idea is that the general RNN will perform much better than RNNs specialised in the wrong musical style, and only slightly worse than the RNN trained on the correct musical style. Hence, the correct RNN will produce results similar to the general RNN whilst the incorrect RNNs will not. Once a specialised RNN has been selected its output is passed to a Dynamic Bayesian Network which calculates the most likely tempo and beat phase at regular time intervals. The calculated beat phases and the list of beat probabilities output by the RNN are then used to determine the most likely beat positions, which are returned.

The accuracy of the returned beats was tested in both (Böck, Krebs and Widmer, 2014) and (Krebs, Böck and Widmer, 2015) using various standard evaluation measures across several datasets of audio clips. (Böck, Krebs and Widmer, 2014) compared the results to some other beat trackers and found that its beat tracker performed the best in most cases. One of the datasets included the scores of a human tapper for comparison, and the authors of (Böck, Krebs and Widmer, 2014) concluded that their beat tracker may outperform a human listener in certain scenarios. Since the beat tracker described in (Böck, Krebs and Widmer, 2014) does not match the BK1 beat tracker perfectly, its evaluation must be examined alongside the evaluation in (Krebs, Böck and Widmer, 2015). This latter evaluation shows that the inclusion of the modified Dynamic Bayesian Network improves performance over the original bar pointer model. Looking at the MIREX 2016 beat tracking results ((Nema.lis.illinois.edu, 2016), (Nema.lis.illinois.edu, 2016), (Nema.lis.illinois.edu, 2016)) shows the BK1 beat tracker consistently scoring very highly in comparison to the other beat trackers, often scoring within the top three.

The BK3 submission, described in (Böck and Korzeniowski, 2016), is quite similar to the beat tracker described in (Korzeniowski, Böck and Widmer, 2014), but using comb filters instead of autocorrelation to detect dominant tempo as described in (Böck, Krebs and Widmer, 2015). This beat tracker works approximately the same as BK1, passing the audio through several fourier transforms into a Recurrent Neural Network which produces a set of beat probabilities for each frame of the audio signal. These are passed into a bank of comb filters as described in (Böck, Krebs and Widmer, 2015) to estimate the dominant tempo. This is then passed into a Dynamic Bayesian Network which uses the calculated tempo and the output of the RNN to calculate the most likely beat positions.

Since the CRFBeatDetector is a combination of both (Korzeniowski, Böck and Widmer, 2014) and (Böck, Krebs and Widmer, 2015) the evaluations of both are examined to determine the effectiveness of the beat tracker. The evaluation of (Korzeniowski, Böck and Widmer, 2014) showed that the proposed beat tracker scored quite highly, with scores higher than those of several alternative beat trackers. Meanwhile the evaluation in (Böck, Krebs and Widmer, 2015) showed that replacing autocorrelation with the comb filter method of tempo estimation improves the results of a beat tracker similar to the one described in (Korzeniowski, Böck and Widmer, 2014). Looking at the MIREX2016 results ((Nema.lis.illinois.edu, 2016), (Nema.lis.illinois.edu, 2016), (Nema.lis.illinois.edu, 2016)) also shows promising performance, with the BK3 beat tracker scoring highly alongside BK1 and the SB submissions.

2.2.3. DZ1 Beat Tracker

The DZ1 beat tracker was submitted to the MIREX beat tracking task by Bruno Di Giorgi, Massimiliano Zanoni, and Augusto Sarti in 2014. It is described in (Di Giorgi, Zanoni and Sarti, 2014) as using a complex spectral difference onset detection function and a "tempo path" to detect beats. It is a multi-agent system, using a group of beat tracking agents that all try to find the next beat given the previous beat, dominant tempo, and the output of the onset detection function. All agents are based on a forward search which attempts to maximise a cost function. This cost is calculated by looking at how close the beat interval matches the tempo, and the magnitude of the onset detection function at that point. The cost of each beat is added together to calculate a total cost for that tracker, and the one with the highest cost at the end is chosen as the most accurate set of beats. Looking at this beat tracker's MIREX results on the SMC (Nema.lis.illinois.edu, 2014), MAZ (Nema.lis.illinois.edu, 2014), and MCK (Nema.lis.illinois.edu, 2014) datasets shows that this beat tracker's performance is low compared to other tested beat trackers. Unfortunately, no public implementation of this beat tracker could be found.

2.2.4. ES/EWFS Beat Trackers

The ES1, ES3, and EWFS1 beat trackers were submitted to MIREX in 2013 by Florian Eyben, Felix Weninger, Giacomo Ferroni, and Björn Schuller. All 3 beat trackers share the same accompanying pdf, to be found at (Eyben et al., 2013). From the title of the short description it seems that this beat tracker would have similarities to the BK3 beat tracker, which also uses LSTM Neural Networks and Comb Filters. However, unlike the BK3 submission these beat trackers perform quite poorly in the MIREX Beat Tracking event, on all tested datasets (Nema.lis.illinois.edu, 2013),(Nema.lis.illinois.edu, 2013),(Nema.lis.illinois.edu, 2013).Unfortunately, a longer paper describing this beat tracker in more depth could not be located and no public implementation was listed.

2.2.5. MD2 Beat Tracker

The MD2 beat tracker was submitted in 2014 by Michelle L. Daniels, and is described in (Daniels, 2014). It is another multi-agent system which runs multiple beat tracking agents simultaneously and then estimates the final beats from their outputs. Each tracker works by estimating the dominant tempo of the next five or ten (depending on the agent) seconds of music, and they all use varying input features and tempo ranges. Different methods were employed to extract tempo from the input feature, including comb filters, fourier transforms, and autocorrelation. This tempo and the input feature are then used to estimate the location of the next beats. Each agent then determines how confident it is that its tempo and beat estimates are accurate. The agents are then grouped by beat location/tempo, and the group with the highest overall confidence value is selected as the correct group. The correct group's output beat and tempo values are then averaged and outputted as the final beat and tempo. As a result, the final beats are not taken directly from any one beat agent, but are instead taken as the average of the most confident beat agents. Looking at this beat tracker's MIREX results ((Nema.lis.illinois.edu, 2014), (Nema.lis.illinois.edu, 2014), (Nema.lis.illinois.edu, 2014)) showed that this beat tracker has very lacklustre performance overall. No public implementation of this beat tracker could be located.

2.2.6. ODGR Beat Tracker

The ODGR beat tracker was submitted in 2012 by João Lobato Oliveira, Matthew E. P. Davies, Fabien Gouyon, and Luís Paulo Reis. Each of the submissions ODGR 1 to 4 are the same underlying beat tracker run in slightly different modes (Oliveira et al., 2012). The beat tracker itself is the INESC-Porto Beat Tracker, or IBT, and is freely available under GPL licensing and described in detail in (Oliveira et al., 2012). IBT, based on the BeatRoot beat tracker (Oliveira et al., 2010), is a multi-agent system whose agents each store a hypothetical beat phase and tempo. The audio signal is passed through a spectral flux onset detection function to the agent inducer, which estimates tempo possibilities using an autocorrelation function. Then for each possible tempo a set of hypothetical beat phases is calculated, and whichever best fits the spectral flux is chosen as that tempo's corresponding phase. Finally, these tempo-phase pairs are scored and ranked before being added to the pool of agents. This process, known as "agent induction", is intended to initialise the agent pool with a variety of possible tempo-phase pairs that will be used to estimate beat positions. IBT's various modes change the frequency at which the agents get re-induced, as allowing the agents to be re-induced more often allows IBT to better handle tempo changes within the audio signal. Once these initial agents have been created they propagate beat predictions according to their internal tempo and phase. By looking at onset events in the spectral flux and comparing them to the predicted beat times of a given agent it is determined how close the agent was to predicting the event. If the agent was very close then it is slightly modified to better match the event. If the agent was completely off then nothing happens. If the agent was reasonably close then three child agents that are slight variations of the original agent are added to the agent pool. This way a change in tempo or beat phase can be accounted for by the new agents, but false positives do not throw the system off. An agent referee continually evaluates the agents by how closely their predictions fit the spectral flux data, and the beats of the current best agent are output as the beats estimated by the system.

In (Oliveira et al., 2012) this beat tracker is evaluated when running on individual audio clips as well as audio streams. This was done to evaluate the beat tracker's overall accuracy as well as to test its ability to handle song transitions. The streaming results showed AMLt scores averaging around 90%, meanwhile the online/offline results showed reasonable AML and CML results, being only slightly lower than the SDP beat tracker (Oliveira et al., 2012). Looking at this beat tracker's MIREX results ((Nema.lis.illinois.edu, 2012), (Nema.lis.illinois.edu, 2012), (Nema.lis.illinois.edu, 2012)) shows that it performs quite badly compared to the other beat trackers tested at the time, but the MIREX evaluations only test beat trackers in a non-streaming scenario, so it may perform comparatively better in streaming scenarios.

2.2.7. JZ Beat Tracker

The MultiFeature Beat Tracker was submitted to the MIREX beat tracking event every year from 2012 to 2016 by José R. Zapata under the acronym JZ (ZDG before 2014). Each time it was submitted twice, once when running in INF mode (JZ1) and once in REG mode (JZ2). The MultiFeature Beat Tracker is publicly available as part of the Essentia framework (Zapata, 2016), which can be downloaded from (Essentia.upf.edu, 2013). The 2016 submission is based on the beat tracker described in (Zapata, Davies and Gomez, 2014), but with a different selection of Onset Detection Functions, and using the REPET algorithm (Rafii and Pardo, 2013) as a pre-processing step before onset detection (Zapata, 2016). Whilst most beat trackers pass the audio input through a single onset detection function, this beat tracker passes the audio to several onset detection functions simultaneously. Each onset detection function's output is then passed to the Degara beat tracker described in (Degara et al., 2012). Each output is analysed independently to produce a series of beat times for each onset detection function, then one of these series is selected to be returned by the MultiFeature Beat Tracker. The selection process looks at the level of "mutual agreement" between the produced beat series and chooses the series with the highest level of mutual agreement. Mutual agreement is calculated by evaluating the similarity between the beat sequences, which is done slightly differently in JZ1 to JZ2. JZ1 compares the outputs using the Information Gain measure from (Davies, Degara and Plumbley, 2011), whereas JZ2 uses the Regularity Function described in (Marchini and Purwins, 2011).

The evaluation present in (Zapata, Davies and Gomez, 2014) showed that this beat tracker scored quite highly compared to other beat trackers from 2012, even scoring competitively against the multi-agent beat tracker that inspired the paper. The results present in (Zapata, 2016) also show a slight increase in performance across the years, with the MIREX 2016 submission scoring higher than the 2012 submission in the AMLc and AMLt categories against the Mckinney dataset.

2.3. Conclusions

From looking at the MIREX submissions of the last six years it seems that many of them are multi-agent beat trackers, some have publicly available implementations, and many were submitted several times across multiple years. Some of the beat trackers have only been evaluated through the MIREX Beat Tracking task, and of those that were also evaluated independently most followed a similar evaluation procedure to MIREX. Only the INESC-Porto Beat Tracker (Section 2.2.6) was evaluated differently, by running it on audio streams as well as individual audio clips. Even in this case the stream testing was only performed using audio clips the beat tracker scored 100% on in the non-streaming scenario, and only the AML evaluation was used.

This shows a clear lack of extensive stream testing in current beat tracker research, as most of the examined beat trackers have only been tested in non-streaming scenarios such as those provided by MIREX. On top of this the only beat tracker that was tested in streaming scenarios was not tested extensively, but was only tested on a few audio files and only using a small subset of the evaluations employed in standard non-stream scenarios. This study intends to fill this research gap by testing several of the previously examined beat trackers in both streaming and non-streaming scenarios, using a wide array of audio clips and evaluation measures.

3. Research Methodology

3.1. Introduction

This section describes the work done during this study, as well as describing the beat trackers, evaluation measures, and datasets that were used. The tested beat trackers were found by looking at the submissions to the MIREX Beat Tracking tasks of the last 6 years (described in Section 2). The chosen beat trackers were then used to extract beats from a large array of audio files including both individual audio clips, and audio streams. The audio stream files were created by combining several audio clips into a single audio file, which allowed the streaming and non-streaming results to be directly compared as they both operated on the same audio clips. The standalone clips were each accompanied by a set of annotations detailing the correct beat times, which were used to determine the accuracy of each beat tracker on that specific clip.

Each beat tracker's performance was calculated using several of the standard evaluation measures used in the MIREX competitions, those being the F-Measure, Cemgil, Goto, Pscore, CML/AML, and D evaluations. These each judge the accuracy of a beat tracker by comparing its output beat times to a set of ground truth annotations, typically produced by hand. Each measure produces a floating-point value, typically a percentage, representing how close the beat detections were to the annotations (Davies, Degara and Plumbley, 2009). The evaluation measures all compare the beat detections to the annotations in different ways, each of which is covered in section 3.5.

The final results were obtained by averaging the evaluations of both non-streaming and streaming results for each beat tracker. This allowed for a clear comparison between the beat trackers' accuracy in both scenarios. The streaming averages were then subdivided into several categories to allow for an examination of how the overall makeup of a stream can affect the accuracy of each beat tracker, and how each beat trackers accuracy varies over time within the stream.

This methodology is organised as follows. Section 3.2 lists the beat trackers that were chosen for further testing and describes the installation process of each. Section 3.3 describes the audio datasets that were used to create the audio stream files, as well as being used to test the beat trackers' non-streaming accuracy. Section 3.4 describes the procedure used to produce the audio streams and Section 3.5 describes the standard beat tracking evaluation measures that were used to determine beat tracking accuracy. Finally, Section 3.6 describes how the beat trackers, datasets, and evaluations were used to obtain the final results.

3.2. Tested Beat Trackers

The beat trackers chosen for further testing were taken from the MIREX beat tracking submissions between 2012 and 2016. Of the beat trackers submitted during this time the ODGR1/4, JZ1, BK1/3, and SB8/9 submissions were selected for further testing. These submissions were chosen based on their public availability. All beat trackers were installed, tested, and run on Linux Mint 17.3.

3.2.1. INESC-Porto Beat Tracker

Of the four versions of the INESC-Porto Beat Tracker that were submitted as ODGR1->4, only the non-causal (ODGR1) and causal (ODGR4) modes were tested. Testing both should allow for a comparison of causal and non-causal methods of beat detection, whilst also allowing for more direct comparisons to beat trackers which only operate causally or non-causally. The implementation itself was downloaded from (Smc.inesctec.pt, 2012) as a standalone executable built for 32bit Linux. It was tested on a two minute and fifteen second long, stereo, 44.1KHz wave audio file to confirm the implementation worked as expected.

3.2.2. Essentia MultiFeature Beat Tracker

The Essentia MultiFeature Beat Tracker, last submitted to MIREX in 2016 as JZ1 and JZ2, was tested due to its public availability as part of the Essentia c++ library for audio analysis and information retreival(Essentia.upf.edu, 2013). Unfortunately, the Essentia framework only contains the 2013 version of the beat tracker, rather than the improved version submitted in 2016. The framework itself was installed and compiled following the instructions on (Essentia.upf.edu, 2013), and the MultiFeature Beat Tracker executable was found within the Essentia directory under "*build/src/examples/streaming_beattracker_multifeature_mirex2013*". This executable was copied into a separate directory and tested as a standalone application to confirm its successful execution.

3.2.3. Madmom Beat Trackers & Beat Detectors

The BK1/3 and SB8/9 beat trackers submitted to MIREX in 2016 are all included as example beat trackers within the madmom audio signal processing library, available at (Böck, 2013). Before installing the madmom library *python-pip* and *python-dev* were downloaded using the "*apt-get install*" terminal command. The madmom dependencies of *numpy 1.12.0*, *scipy 0.18.1*, *cython 0.25.2*, and *nose 1.3.7* were then installed using pip. Finally, the madmom library itself was downloaded and installed following the instructions at (Madmom.readthedocs.io, 2015). Once the library was installed the DBNBeatTracker (BK1), CRFBeatDetector (BK2), BeatDetector (SB8), and BeatTracker (SB9) python scripts were found in the "*madmom/bin*" folder and tested to confirm the installation was successful.

3.3. Datasets

To properly test the beat trackers a series of datasets containing audio clips and annotated beat times was required. The datasets were found online using MIR dataset lists such as (Music-ir.org, 2016), (Lerch, 2017), (Holzmann, 2009), and (Raffel, 2015). The datasets in these lists were examined to locate datasets that included annotated beat times, and then narrowed further to datasets of reasonable size that were publicly available. The datasets that were ultimately chosen and used were the GTZAN, Ballroom, and SMC datasets.

3.3.1. SMC Dataset

The SMC Dataset (Holzapfel et al., 2012) has been used in the MIREX Beat Tracking competitions since 2012. It is made up of 217 audio clips, each 40 seconds long, in 16 bit mono 44.1KHz wave format. Files 278-289 are marked as "easy" for beat detection whilst the rest are "hard" to extract beats from. Each audio file also came with a set of tags which identify why the file may have been difficult to annotate and a number representing how confident the annotator was that the annotations are correct. These tags were not used in this study because the accuracy of the annotations doesn't affect the comparison between streaming and non-streaming results, as they both use the same annotations.

3.3.2. Ballroom Dataset

The Ballroom dataset (Krebs, Böck and Widmer, 2013) contains 698 audio clips between 30 and 31 seconds long in 16 bit mono 44.1kHz wav format. These audio files are organised by musical genre, specifically ChaChaCha, Jive, Quickstep, Rumba (american), Rumba (international), Rumba (miscellaneous), Samba, Tango, Viennese Waltz, and Waltz. This dataset's audio files and annotations are not available in the same location online, so they had to be accessed and downloaded separately. The annotations themselves took the form of simple text files with each line storing the bar position and time of a given beat. Since only the beat times were required in this case the annotations were modified to remove the bar positions.

3.3.3. GTZAN Dataset

The GTZAN dataset (Marchand and Peeters, 2015) consists of 1000 30 second long audio files of 16 bit mono 22.05KHz in .au format. These files were reformatted to .wav files to make sure the beat trackers would be able to operate on them. The annotations took the form of xml files storing the time, bar position, tatum, and various other pieces of information about each beat. These files were converted to text files and modified to remove all information and formatting other than the beat times. This was done to make sure all dataset annotation files used a consistent format.

3.4. Stream Audio Files

Whilst the datasets provide many audio clips for non-stream testing they do not include any stream files. As such a set of stream files was created from the audio clips present in each dataset. Each stream file was created by selecting five random audio clips and stitching them together using the Sox command line utility (installed through apt-get). The name and duration of each audio clip within the stream was also noted in an accompanying text file. 100 stream files were created for each dataset, containing audio clips taken from within that dataset. On top of this another 100 stream files were created from audio clips within each audio sub-category of each dataset. This was done because it can be expected that audio clips within a sub-category or genre will be more similar than audio clips from different genres. Hence, stream files within a sub-category may contain less variation than stream files that cover entire datasets. Creating both sets of stream files was intended to allow comparison of beat tracker performance on audio streams with low internal variation to performance on streams with high internal variation.

3.5. Evaluations

To determine how accurate a given beat tracker's output was the set of standard evaluation measures used in the MIREX Beat Tracking task were used to compare its output to the beat annotations that came with each dataset. The evaluations used were the F-measure, Cemgil, Goto, PScore, CMLc, CMLt, AMLc, AMLt, and D evaluations (Music-ir.org, 2016). A variety of evaluations were used because they each use slightly different characteristics of a beat tracker's beat detections to determine their accuracy. Each evaluation is briefly described in this section but for more detailed descriptions see (Davies, Degara and Plumbley, 2009). The evaluation descriptions are followed by a description of the implementations used in this study.

3.5.1. F-Measure Evaluation

The F-measure evaluation uses a short time window around each beat annotation to identify the number of correct, incorrect, and missing beat detections in a beat tracker's output. A score is calculated by comparing these quantities to the total number of beats, intended to reflect the overall accuracy of the beat tracker as a percentage. This evaluation measure is simple enough to implement and gives reasonable idea of how accurate a beat tracker is. Unfortunately, it doesn't handle partially correct beat detections very well. Since each detection is treated as either correct or incorrect, all detections within the time window are treated as correct regardless of how close they were to the annotation. At the same time, any beat detections that were placed on the off beat will always be treated as incorrect even though the correct beat period may have been determined by the beat tracker. These two problems mean this evaluation measure may score beat trackers higher or lower than they deserve.

3.5.2. Cemgil Evaluation

The Cemgil measure (originally used in (Cemgil et al., 2000)) does away with the binary correct/incorrect beat detection scoring system. Instead, a Gaussian error function is used to measure each individual detection's accuracy between 0 and 1, 0 being least accurate. These scores are then added up and divided by the average of the total number of beat detections and the number of annotations, resulting in a percentage representing the accuracy of the detections. Whilst this solves the problem present in the F-measure of all beats being treated as either correct or incorrect, it still has the problem of detections present on the off beat being marked as incorrect.

3.5.3. P-Score Evaluation

The P-score, defined in (McKinney et al., 2007), takes a very similar approach to the F-measure by looking at how many detections are correct compared to the annotations and normalising the result by the number of annotations (or the number of detections if it is larger). The main difference between the F-measure and the P-score is that the P-score ignores the first five seconds of detections. This was done to minimise the initialisation effects the beat tracker may have on its detections that could decrease their final score due to a warm up period (McKinney et al., 2007).

3.5.4. Goto Evaluation

The Goto evaluation (Goto and Muraoka, 1997) is unique among all the evaluation measures in that it returns either true (as 1) or false (as 0), rather than a floating-point value representing accuracy. This is because the Goto evaluation was created to determine if a beat tracker's output converges to a given beat sequence over time or not. It does this by calculating a beat error sequence by looking at the time distance between each annotation and its nearest detection. The error is normalised using the beat period around the selected annotation, resulting in a range of lowest error (0) to highest error (1). Then a beat error subsequence is extracted, taking all errors values below 0.35. Finally, a set of rules described in (Goto and Muraoka, 1997) is applied to this sequence to determine if the detections are, on average, correct or incorrect.

3.5.5. CML/AML Evaluations

The CML and AML evaluations, developed in (Hainsworth, 2004) and (Klapuri, Eronen and Astola, 2006) are a family of evaluations which look at the length of segments of continuously correct detections. Sections of correct detections are located by looking at each detection's proximity to an annotation and considering the previous detection's proximity to the previous annotation. Then the length of the longest continually correct segment (CMLc) or the total length of all correct segments (CMLt) are compared to the full length of the annotations to get an accuracy percentage. The AMLc and AMLt evaluations follow the same principle, but whilst the CML evaluations only treat beats at the annotated metrical level as correct, the AML evaluations also allow for correct tapping at half or double the annotation's metrical level (Davies, Degara and Plumbley, 2009).

3.5.6. D Evaluation

The D evaluation (Davies, Degara and Plumbley, 2009) follows a similar process to Goto, obtaining the beat error sequence of the detections given the annotations. However, it also looks at the inverse, finding the beat error of the annotations given the detections. Both beat error sequences are then normalised to between -0.5 and 0.5 and converted into a histogram of 41 bins for which the information gain is calculated. The lowest information gain of the two histograms is then used as the output, and ranges from 0 to log2(41). The main benefits of this evaluation measure are that it avoids the two problems the F-measure evaluation has. By examining a histogram of beat error, if the beats are consistently off by the same amount this suggests that the beat tracker detected the right tempo, and this evaluation would score this reasonably well. At the same time, the beat error grades each detection on a scale rather than using F-measure's binary correct or incorrect system.

3.5.7. Implementation

Each of these evaluations were calculated using the mir_eval python library, available at (Raffel, 2014), which provides implementations of all these standard evaluations. The mir_eval library's outputs were modified slightly to normalise/un-normalise certain evaluations, to bring them in line with the range used in the MIREX beat tracking task. This was done to allow the evaluation results to be directly compared to MIREX results, to confirm that the beat trackers and evaluations were all operating correctly. This comparison was done using the beat tracker's accuracy on the SMC dataset, as the SMC dataset is common to the MIREX Beat Tracking task and this study. Comparing the results of each beat tracker on the SMC dataset to the MIREX results for that same beat tracker showed that the local implementation's accuracies were very close to their MIREX counterparts. This showed that the beat trackers and evaluation measures had been implemented properly, as incorrect implementations would be expected to cause a large difference between the two accuracies.

3.6. Obtaining Results

To properly evaluate the beat trackers' performance in the non-streaming scenarios they were each run on every audio clip within the three datasets. Their output beat detections were stored in a directory structure matching that of the datasets themselves, making it simple to locate the audio clip from the detections file, or vice versa. This made it very simple to apply the evaluation measures to the beat detections, as the beat detections' filepath could be easily modified to point to the beat annotations file for the same audio clip. Once the evaluations had been calculated using bash script and the mir_eval python library, the averages of each evaluation for each beat tracker were calculated. This was done using a combination of bash script and R-Script to loop through all calculated evaluations and determining the mean, min, max, and standard deviation (among other things) of each evaluation measure. The resulting average evaluations were then used to analyse beat tracker performance (see Section 4).

The accuracy of each beat tracker in the streaming scenario was obtained in a similar fashion. First the beat trackers were run on each of the audio stream files to produce a set of detections for each audio stream. These detections were then split up into a single detections file for each audio clip within the stream. This was done by looking at the names and durations of each audio clip in the audio stream’s accompanying text file. These split detections could then be compared to the annotations of their related audio clip in the same manner as for the non-stream detections. Before averaging these stream evaluations across all audio clips, the average stream evaluations for each audio clip were calculated. This guaranteed that every audio clip would have a single set of stream evaluations that could be included in the overall average. This was done to give every audio clip the same weight in the overall average. If this extra step were not taken then the overall averages would be weighted in favour of audio clips which were present in many audio streams, and against those that were present in only a few. These audio clip averages were then averaged using a similar combination of bash and R-Script to that used to average the non-stream evaluations.

Several other audio stream averages were also calculated to get a better overview of how audio streams affect the beat trackers. These were averages of specific subsets of the audio streams, such as the dataset wide streams, the genre local streams, and several averages were calculated for each audio clip position within the audio streams. These subset averages were calculated in the same way as the overall averages, by first calculating the average for each audio clip, and then using bash and R-Script to calculate the mean, min, max, and standard deviation of the audio clip averages.

4. Results and Analysis

4.1. Introduction

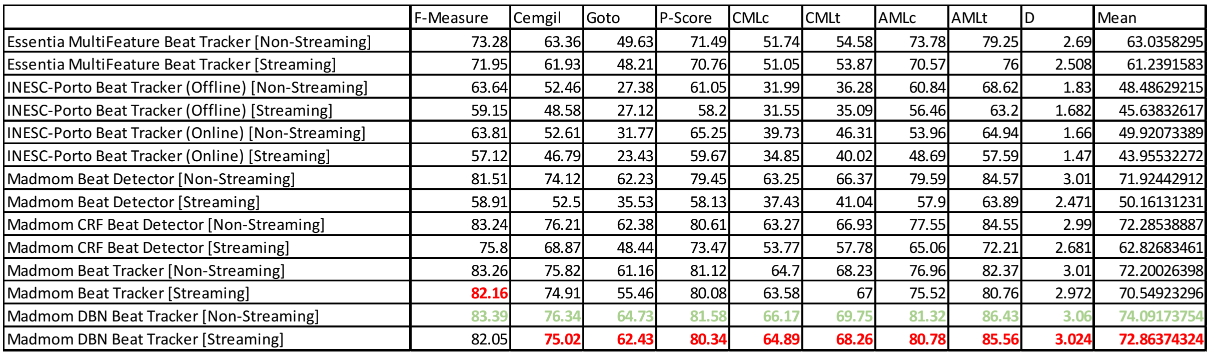
This section looks at the average evaluations of each beat tracker to see how they performed in various streaming and non-streaming scenarios. To glean more information from the results the averages were broken down into overall averages, averages from the dataset wide stream files, averages from the genre local stream files, and averages for each audio clip position within the stream files (first clip, second, third, etc). This allowed the performance of each beat tracker in both streaming and non-streaming scenarios to be compared, as well as showing how dataset streams and genre streams affected beat tracker accuracy. An examination of how each beat tracker's accuracy varied over time was also conducted by looking at the average accuracies for each audio clip position within the stream files.

The results and analysis are broken down into the following two sections. The first section compares each beat tracker's accuracy in streaming scenarios to its accuracy in non-streaming scenarios, including a breakdown of accuracy by streaming file type. The second section examines the accuracy of each beat tracker over time, by comparing the average accuracy of each of each streaming file's five audio clips to one another. The intention of these sections was to observe how streaming scenarios affected each beat tracker's accuracy, and to compare the beat trackers to one another to see which beat tracking techniques were most effective in the streaming/non-streaming scenarios.

4.2. Streaming vs Non-Streaming Accuracy

The various evaluation measures described in the Methodology were applied to each beat tracker's streaming and non-streaming outputs. The mean of each evaluation was then calculated to produce average accuracies for each beat tracker in both streaming and non-streaming scenarios. These averages are presented in Table1 with the top score for each stream evaluation written in green, and the top score for each non-stream evaluation written in red. The mean of these average scores was then calculated to obtain an overall accuracy value for each beat tracker in both scenarios. This average was calculated by first converting the D value to a percentage by multiplying it by 100/LOG2(41), and then using that instead of the D value when calculating the mean of all nine values. These means were then used to produce

Fig1, which shows each beat tracker's overall accuracy in streaming and non-streaming scenarios.



	F-Measure	Cemgil	Goto	P-Score	CMLc	CMLt	AMLc	AMLt	D	Mean
Essentia MultiFeature Beat Tracker [Non-Streaming]	73.28	63.36	49.63	71.49	51.74	54.58	73.78	79.25	2.69	63.0358295
Essentia MultiFeature Beat Tracker [Streaming]	71.95	61.93	48.21	70.76	51.05	53.87	70.57	76	2.508	61.2391583
INESC-Porto Beat Tracker (Offline) [Non-Streaming]	63.64	52.46	27.38	61.05	31.99	36.28	60.84	68.62	1.83	48.48629215
INESC-Porto Beat Tracker (Offline) [Streaming]	59.15	48.58	27.12	58.2	31.55	35.09	56.46	63.2	1.682	45.63832617
INESC-Porto Beat Tracker (Online) [Non-Streaming]	63.81	52.61	31.77	65.25	39.73	46.31	53.96	64.94	1.66	49.92073389
INESC-Porto Beat Tracker (Online) [Streaming]	57.12	46.79	23.43	59.67	34.85	40.02	48.69	57.59	1.47	43.95532272
Madmom Beat Detector [Non-Streaming]	81.51	74.12	62.23	79.45	63.25	66.37	79.59	84.57	3.01	71.92442912
Madmom Beat Detector [Streaming]	58.91	52.5	35.53	58.13	37.43	41.04	57.9	63.89	2.471	50.16131231
Madmom CRF Beat Detector [Non-Streaming]	83.24	76.21	62.38	80.61	63.27	66.93	77.55	84.55	2.99	72.28538887
Madmom CRF Beat Detector [Streaming]	75.8	68.87	48.44	73.47	53.77	57.78	65.06	72.21	2.681	62.82683461
Madmom Beat Tracker [Non-Streaming]	83.26	75.82	61.16	81.12	64.7	68.23	76.96	82.37	3.01	72.20026398
Madmom Beat Tracker [Streaming]	**82.16**	74.91	55.46	80.08	63.58	67	75.52	80.76	2.972	70.54923296
Madmom DBN Beat Tracker [Non-Streaming]	**83.39**	**76.34**	**64.73**	**81.58**	**66.17**	**69.75**	**81.32**	**86.43**	**3.06**	**74.09173754**
Madmom DBN Beat Tracker [Streaming]	82.05	**75.02**	**62.43**	**80.34**	**64.89**	**68.26**	**80.78**	**85.56**	**3.024**	**72.86374324**

Table1. This table shows the mean evaluations for each beat tracker when operating in both streaming and non-streaming scenarios, alongside the average value of each mean evaluation. The highest scores of the non-streaming scenario are highlighted in green, whilst those of the streaming scenario are highlighted in red.

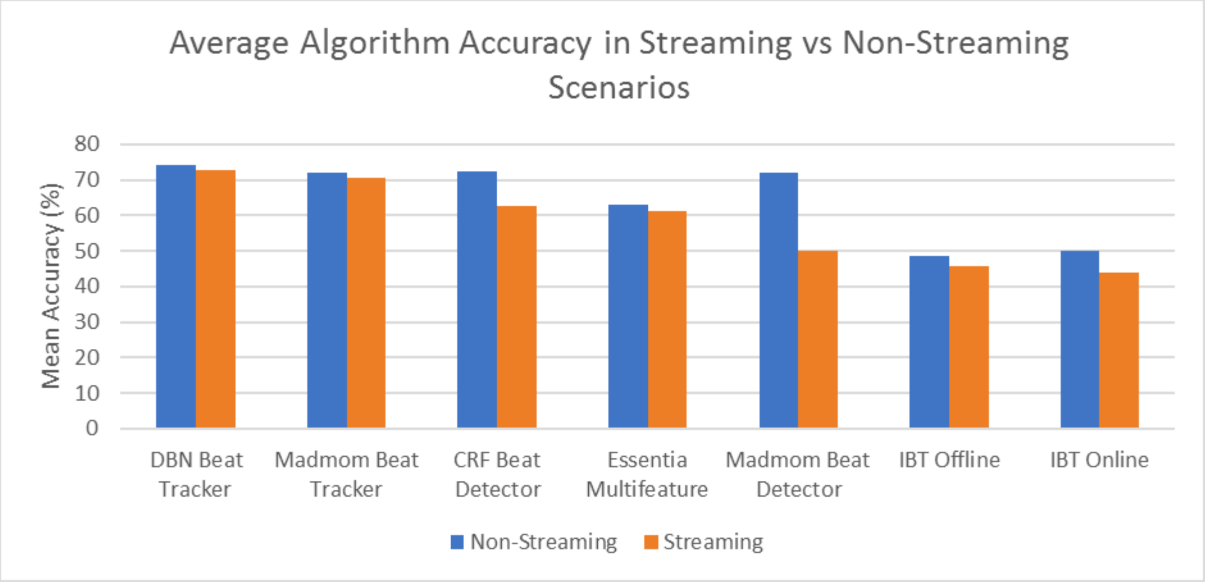


Fig1. This graph shows the average accuracy of each beat tracker in both streaming and non-streaming scenarios, ordered in decreasing order by stream accuracy.

Looking at Fig1 suggests that accuracy on audio streams is consistently lower than accuracy on standalone audio files. To confirm this a series of Z-Tests were performed that compare each beat tracker's non-streaming evaluations to their streaming evaluations. The resulting Z-Scores of these tests are shown in Table2, alongside an average Z-Score for each beat tracker. Using a one-tailed Z-Test with a confidence threshold of 95%, any Z-Score below the critical value of -1.64 is considered significant. The null hypothesis in this case is that a given evaluation measure's stream average does not deviate from its non-stream average. Looking at Table2 shows that most Z-Scores were significantly negative, and none of the Z-Scores were positive. The overall Z-Scores for each beat tracker suggest that their streaming evaluation scores were significantly lower than their non-streaming evaluation scores. This matched expectations as individual audio clips tend to contain less internal variation than groups of audio clips. This is due to potentially varying tempo and musical style between the different clips in the group whilst a single clip tends to stick to a single musical style.

	F-Measure	Cemgil	Goto	P-Score	CMLc	CMLt	AMLc	AMLt	D	Average Z scores
Essentia	**-1.97810524**	**-2.238908**	-1.238286	-1.10177213	-0.653903	-0.6796212	**-3.97040786**	**-4.68950448**	**-6.78204**	**-2.592505755**
IBTOffline	**-7.446249134**	**-7.033893**	-0.254234	**-4.64266527**	-0.46564	-1.2076818	**-4.88395235**	**-6.66143724**	**-6.69059**	**-4.365149427**
IBTOnline	**-10.09530113**	**-9.630783**	**-7.810443**	**-8.90468686**	**-5.160978**	**-6.5253034**	**-6.24922433**	**-9.79076744**	**-8.53706**	**-8.078283508**
MM Beat Detector	**-41.22522885**	**-38.62356**	**-24.01149**	**-37.3946936**	**-25.7451**	**-26.071708**	**-30.0989305**	**-35.5208974**	**-20.6574**	**-31.03877623**
CRF Beat Detector	**-14.59369832**	**-13.74826**	**-12.54664**	**-13.140368**	**-9.430709**	**-9.3759227**	**-16.6189594**	**-21.1047807**	**-11.5771**	**-13.57071591**
MM Beat Tracker	**-2.173808807**	**-1.729773**	**-5.099006**	**-1.97890009**	-1.1657	-1.3400679	**-1.96442584**	**-2.70827856**	-1.35768	**-2.168627073**
DBN Beat Tracker	**-2.574123104**	**-2.459974**	**-2.098811**	**-2.25796742**	-1.313818	**-1.5913913**	-0.79138892	-1.61498168	-1.49983	**-1.800253776**

Table2. This table shows the Z-Scores obtained by comparing each beat tracker's evaluations in a non-streaming scenario to those of a streaming scenario. Significant difference between the two accuracies is highlighted in red. Negative values indicate that the streaming scenario's accuracy was lower than the non-streaming scenario.

4.2.1. CMLc/AMLc Examination

Whilst these averages give a good overview of each beat tracker's overall accuracy, some of the individual evaluation measures can give a deeper insight into the exact nature of the accuracy drop. Looking at how the CMLc and AMLc measures differ between the streaming and non-streaming scenarios can give some insight into the position within the audio clip at which the incorrect beats are occurring. Since the CMLc and AMLc measures score based on the length of the longest segment of continuously correct beats, if their scores drop dramatically this suggests that the longest continuous segment was interrupted close to its centre. As a result, a CMLc or AMLc drop of around a half suggests incorrect beats were introduced close to the centre of the longest segment of beats, whilst only a small drop indicates that the incorrect beats are most likely present at the fringes of the segment.

Fig2 shows the CMLc and AMLc averages for each beat tracker in both streaming and non-streaming scenarios. It shows that only the Madmom Beat Detector lost a sizeable percentage of its CMLc and AMLc scores when switching to the streaming scenario. Looking back at Table1 shows that a similar drop was experienced by the same beat tracker's CMLt and AMLt evaluations, suggesting that the drop could have been due to an overall accuracy drop rather than due to inaccurate beats cropping up close to the centre of the longest segment of continuously correct beats.

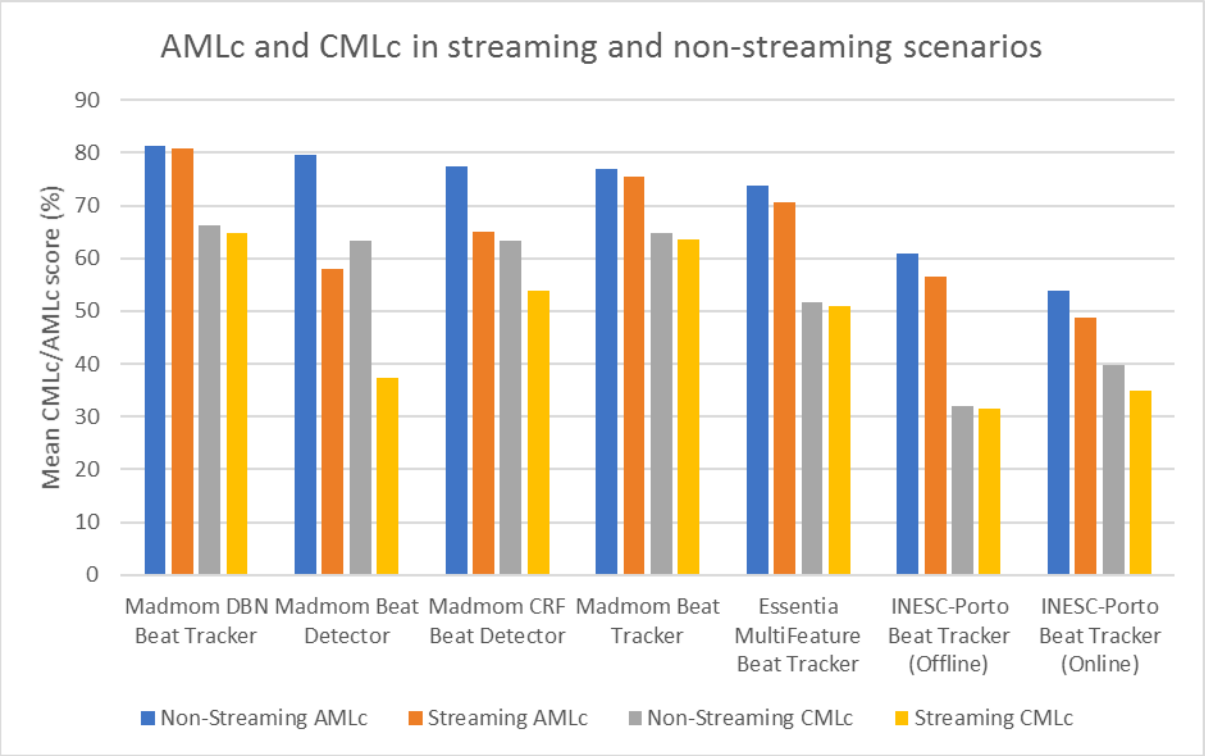
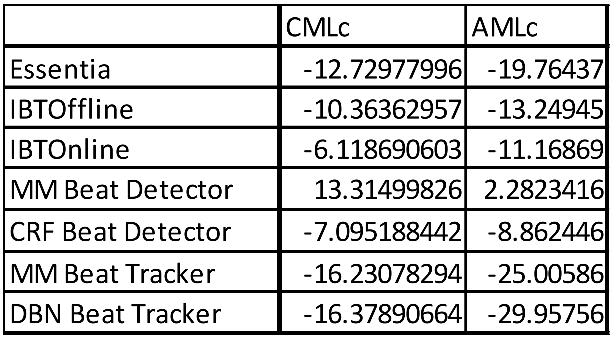


Fig2. This graph shows the average CMLc and AMLc evaluations for each beat tracker in both streaming and non-streaming scenarios, ordered from highest non-streaming AMLc to lowest.

Looking back at the CMLc and AMLc Z-Scores in Table2 shows that four of seven CMLc Z-Scores were insignificant drops, whilst only one AMLc Z-Score was insignificant. Whilst this shows that most CMLc and AMLc drops were significant, it does not show us how significant the drops were. As such a second set of Z-Tests were performed to determine if a fraction of the CMLc and AMLc non-streaming scores (in this case 75% of the original scores) deviated significantly from the streaming scores. The null hypothesis in this case was that 75% of the non-streaming scores did not deviate from the streaming scores. In effect this was checking to see if the streaming scores were 25% lower than the non-streaming scores. Since a 50% decrease would indicate incorrect beats being present at the centre of the longest segment of correct beats, a 25% decrease could indicate incorrect beats near to the middle but not directly in the centre.

The results of these Z-Tests are present in Table3, and clearly show that all Z-Scores were highly significant. The significantly negative scores in Table3 show that the CMLc and

AMLc streaming scores dropped much less than 25%. Meanwhile, the significantly positive Z-Scores of the Madmom Beat Detector showed that this beat tracker's CMLc and AMLc streaming scores dropped by more than 25%.



	CMLc	AMLc
Essentia	-12.72977996	-19.76437
IBTOffline	-10.36362957	-13.24945
IBTOnline	-6.118690603	-11.16869
MM Beat Detector	13.31499826	2.2823416
CRF Beat Detector	-7.095188442	-8.862446
MM Beat Tracker	-16.23078294	-25.00586
DBN Beat Tracker	-16.37890664	-29.95756

Table3. This table shows the Z-Scores of the CMLc and AMLc evaluations when comparing their streaming results to 75% of their non-streaming results. Negative scores indicate the stream evaluations score higher than 75% of the non-stream evaluations.

The Madmom Beat Detector's significant CMLc and AMLc drop could have been the result of incorrect beats being inserted near the middle of the continuous segment of correct beats. The Z-Scores for the other beat trackers show that all of them dropped by much less than 25%. This showed that for most beat trackers the incorrect beats introduced by the streaming scenario fell closer to the edges of the longest segment of continuous correct beats than they did the centre. This could suggest that the incorrect beats were clustered around the edges of the audio clips rather than in their centre. This lines up with the expectation that it was the transitions between songs that caused problems for the beat trackers, as it suggested the incorrect beats were primarily found at the audio clip transitions rather than in the main body of the clips, however, further research would need to be done to confirm this.

4.2.2. Dataset Stream/Genre Stream Results Breakdown

A further breakdown of each beat tracker's average accuracy into genre streams and dataset streams is present in Fig3. Looking at this graph suggests that the Genre streaming scenario had higher average accuracies than the Dataset streaming scenario, whilst both still had lower averages than the non-streaming scenario. To confirm this a series of Z-Tests were performed to compare the average evaluation measures of each beat tracker in genre and dataset streams. The null hypothesis in this case was that the dataset stream evaluations were similar to the genre stream evaluations. The calculated Z-Scores, present in Table4, show that almost all dataset stream evaluations were significantly lower than the genre stream evaluations. This matched expectations as the genre stream files contained less internal variation than the dataset stream files due to the more consistent musical style of the audio clips.

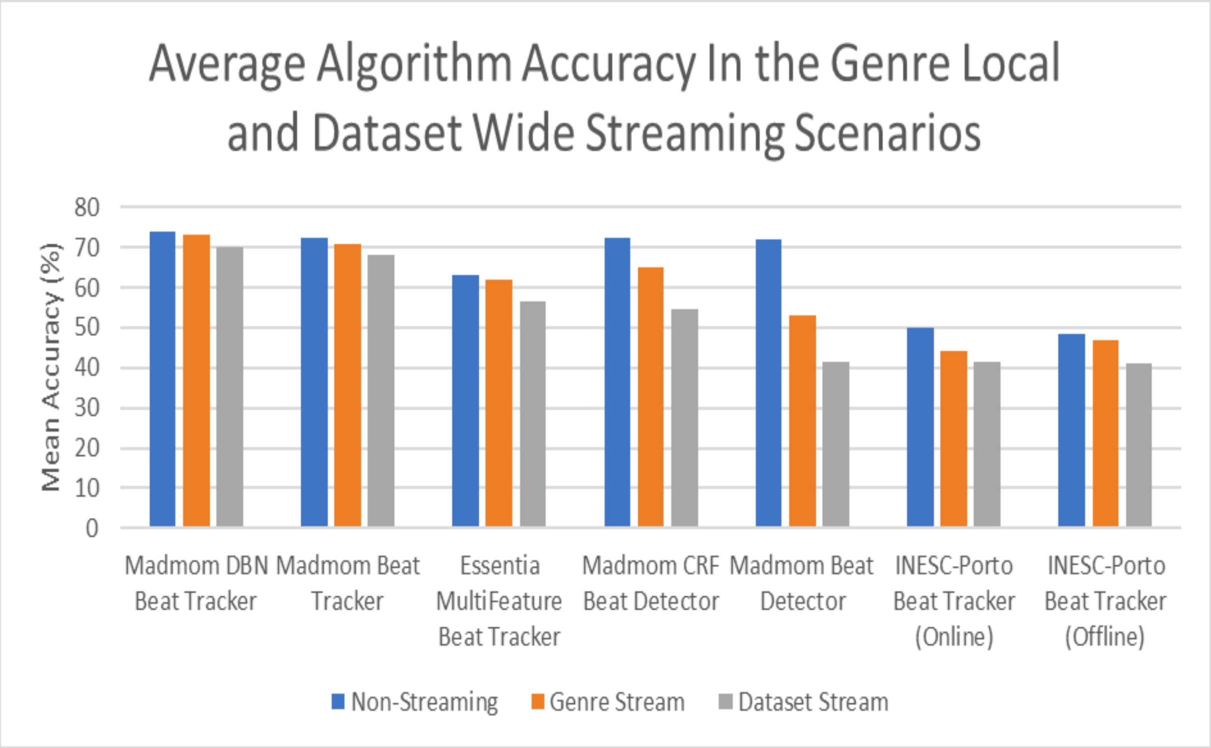


Fig3. This graph shows the average accuracy of each beat tracker in both non-streaming, genre streaming, and dataset streaming scenarios. It is ordered in descending order by dataset streaming accuracy.

	F-Measure	Cemgil	Goto	P-Score	CMLc	CMLt	AMLc	AMLt	D	Average Z scores
Essentia	-6.262662856	-6.480341	-4.266495	-5.00703055	-4.443344	-4.7301905	-4.92045802	-5.62356125	-5.81741	-5.283499548
IBTOffline	-8.78269486	-7.566183	-5.324088	-7.94159304	-6.72029	-7.4753844	-5.34387403	-6.5255943	-3.69166	-6.596818234
IBTOnline	-4.45027353	-3.866701	**-1.252549**	-3.01675961	-2.658245	-3.3820892	-3.58223943	-5.11966381	-3.05761	-3.376236833
MM Beat Detector	-9.982842297	-10.48896	-12.11672	-9.52688107	-12.20813	-11.58281	-14.5403287	-14.3927242	-11.0131	-11.76138525
CRF Beat Detector	-11.2606144	-10.64028	-11.93536	-11.6586412	-11.87174	-12.111057	-8.52656401	-8.90619405	-7.43286	-10.48258928
MM Beat Tracker	-4.125705273	-3.989076	-1.828127	-2.70406883	-2.050891	-2.0569229	-3.13730425	-3.56352797	-4.18085	-3.070719762
DBN Beat Tracker	-4.280215705	-4.52823	-2.9775	-2.81758643	-2.734139	-2.5442823	-3.04867371	-3.03830607	-3.9249	-3.321537146

Table4. This table shows the Z-Scores obtained by comparing each beat tracker’s genre streaming accuracy to their dataset streaming accuracy. A negative value indicates the dataset stream accuracy was lower than the genre stream accuracy. All insignificant values are highlighted red.

The only insignificant Z-Score, highlighted in red, was the INESC-Porto Beat Tracker’s Online Goto evaluation. This suggests that IBT’s Goto score in Online mode didn’t drop significantly between genre and dataset streams. This is counter to the highly significant Z-Score present in table 2 which shows that the Goto evaluation for IBT’s Online mode was significantly lower in streaming scenarios than non-streaming. These Z-Scores, combined with Fig3 would seem to suggest that IBT's Online mode took a large performance hit on all streaming files, but the difference between genre and dataset streams was lower than most. Despite its Goto drop between genre and dataset streams being insignificant, it was still a decrease, and it wasn't too far from the critical value of -1.64. As a result, this data point still fitted with the overall pattern that the dataset streams had significantly lower evaluations than the genre streams.

4.2.3. Beat Tracker Comparison

Fig1 and Fig3 would seem to suggest a beat tracker hierarchy, with the DBN and Madmom Beat Trackers scoring the highest and the INESC-Porto Beat Tracker scoring the lowest. A variety of Z-Tests were performed to confirm this. These Z-Tests compared the evaluation measures of each beat tracker to those of every other beat tracker in both streaming and non-streaming scenarios, with the null hypothesis being that the evaluations of both beat trackers were similar. These evaluation Z-Scores were then averaged to produce a mean Z-Score for each beat tracker comparison in streaming and non-streaming scenarios. These mean Z-Scores are present in Table5. In this table, negative values indicate that the beat tracker named in the row had a lower accuracy than the beat tracker named in the column, whilst values greater than 1.96 or less than -1.96 indicated a significant performance difference. All significant values were highlighted bold, with the significantly positive values coloured green and significantly negative values coloured red.

	IBTOnline	IBTOffline	Essentia	MM Beat Detector	CRF Beat Detector	MM Beat Tracker	DBN Beat Tracker
IBTOnline Non-Stream		1.11287664	**-17.71584**	**-32.12856082**	**-33.14528284**	**-33.51393126**	**-37.04884161**
IBTOnline Stream		**-2.7837689**	**-24.71237**	**-9.870209908**	**-28.45648336**	**-41.14891595**	**-45.32490085**
IBTOffline Non-Stream	-0.99832968		**-18.5366**	**-32.96453191**	**-34.0824112**	**-34.56870104**	**-37.81111781**
IBTOffline Stream	**3.204590407**		**-21.8507**	**-7.257671536**	**-25.70738731**	**-38.07895964**	**-41.96154625**
Essentia Non-Stream	**18.96088808**	**20.1774136**		**-12.55774321**	**-13.52257061**	**-13.60012412**	**-16.34116595**
Essentia Stream	**28.01264532**	**25.4062789**		**17.26152657**	**-2.900022953**	**-14.59905043**	**-17.98979514**
MM Beat Detector Non-Stream	**30.98918863**	**32.481645**	**11.276537**		-0.74459532	-0.55817071	**-3.177879686**
MM Beat Detector Stream	**11.57445232**	**8.39599753**	**-15.12045**		**-19.01273181**	**-31.31407841**	**-34.85512964**
CRF Beat Detector Non-Stream	**31.57999182**	**33.1581584**	**11.832254**	0.671885871		0.199192571	**-2.48373874**
CRF Beat Detector Stream	**31.14410224**	**28.9513092**	**2.7040652**	**20.54838154**		**-11.70783116**	**-15.23983289**
MM Beat Tracker Non-Stream	**31.47384571**	**33.0681473**	**11.658936**	0.443195159	-0.224863452		**-2.735880952**
MM Beat Tracker Stream	**43.12662656**	**41.1656693**	**13.490071**	**32.56210949**	**11.26774279**		**-3.247656546**
DBN Beat Tracker Non-Stream	**33.8408565**	**35.3794797**	**13.983712**	**3.032359127**	**2.337888677**	**2.590895132**	
DBN Beat Tracker Stream	**46.42375639**	**44.3359195**	**16.377712**	**35.67681564**	**14.23169701**	**3.083681334**	

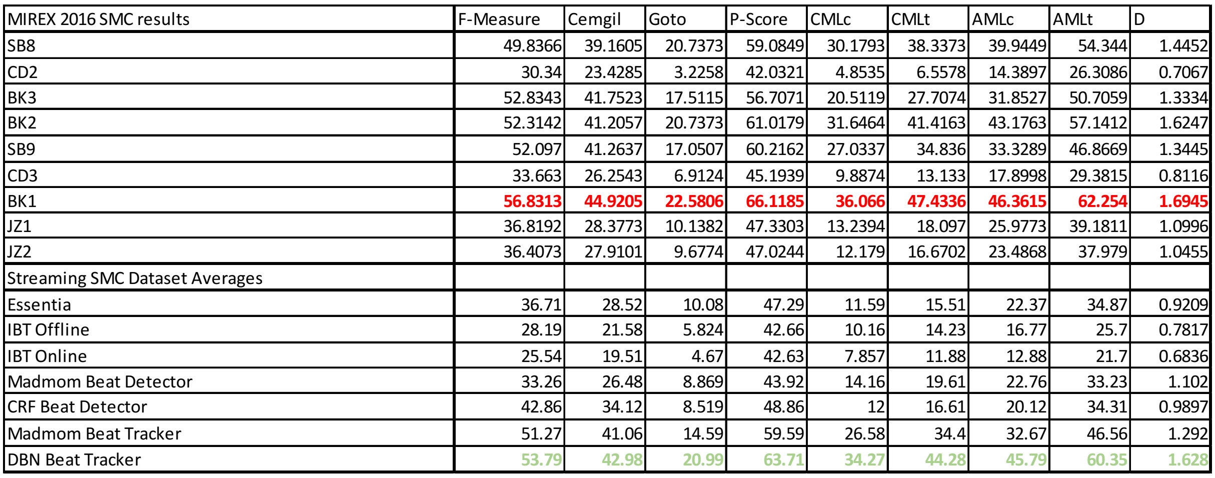
Table5. This table shows the mean Z-Scores obtained by comparing each beat tracker's streaming and non-streaming evaluations to those of the other beat trackers. A negative Z-

Score indicates the beat tracker listed in the row had a lower overall accuracy than the beat tracker listed in the column. Significantly negative values are highlighted in red whilst significantly positive values are highlighted in green.

Looking at Table5 shows the INESC-Porto Beat Tracker performed significantly worse than the other beat trackers in both online and offline modes. This was quite unexpected as it was the only beat tracker that was designed specifically to handle song transitions (Oliveira et al., 2012), so it was expected to perform better in at least the streaming scenarios than the other beat trackers. The CRF and Madmom Beat Detectors scored competitively in the non-streaming scenario, only scoring significantly worse than the DBN Beat Tracker. Unfortunately, their accuracy decreased quite sharply in streaming scenarios, with the Madmom Beat Detector performing significantly worse than the Essentia beat tracker and the CRF Beat Detector scoring significantly worse than the Madmom Beat Tracker. The Essentia MultiFeature Beat Tracker showed middling performance, performing significantly better than the IBT beat tracker in all scenarios, and the Madmom Beat Detector in streaming scenarios.

The DBN Beat Tracker and Madmom Beat Tracker were the most accurate of the tested beat trackers, with DBN Beat Tracker outperforming all others in both streaming and non-streaming scenarios. Looking at Table5 showed that it performed significantly better than all other tested beat trackers in all scenarios. The only time it was outperformed was the non-streaming F-Measure, for which the Madmom Beat Tracker held a higher score than the DBN Beat Tracker. However, performing a Z-Test to compare those two values resulted in a Z-Score of 0.2143, which is insignificant and as such the Madmom Beat Tracker's non-streaming F-Measure is not significantly higher than the DBN Beat Tracker's non-streaming F-Measure. As such the DBN Beat Tracker looks to be the most accurate of all tested beat trackers when it comes to both streaming and non-streaming scenarios.

Following the above internal comparison between the tested beat trackers, a further comparison was done between their streaming accuracies and the results of the MIREX2016 Beat Tracking event. For a direct comparison to be made the MIREX results on the SMC dataset (found at (Nema.lis.illinois.edu, 2016)) were compared to our streaming results on the SMC dataset's audio streams. The relevant data is provided in Table6, which shows that the top score for each evaluation (highlighted red) was held by the MIREX BK1 beat tracker, which was the DBN Beat Tracker’s MIREX submission. The second highest score in each evaluation (highlighted green) was held by our version of the DBN Beat Tracker operating on the SMC dataset stream. This shows that the DBN Beat Tracker’s streaming accuracy was so high that it could even score competitively against other beat trackers in non-streaming scenarios. This shows that it is possible for beat trackers to extract beats from audio streams almost as effectively as from individual audio clips.



MIREX 2016 SMC results	F-Measure	Cemgil	Goto	P-Score	CMLc	CMLt	AMLc	AMLt	D
SB8	49.8366	39.1605	20.7373	59.0849	30.1793	38.3373	39.9449	54.344	1.4452
CD2	30.34	23.4285	3.2258	42.0321	4.8535	6.5578	14.3897	26.3086	0.7067
BK3	52.8343	41.7523	17.5115	56.7071	20.5119	27.7074	31.8527	50.7059	1.3334
BK2	52.3142	41.2057	20.7373	61.0179	31.6464	41.4163	43.1763	57.1412	1.6247
SB9	52.097	41.2637	17.0507	60.2162	27.0337	34.836	33.3289	46.8669	1.3445
CD3	33.663	26.2543	6.9124	45.1939	9.8874	13.133	17.8998	29.3815	0.8116
BK1	**56.8313**	**44.9205**	**22.5806**	**66.1185**	**36.066**	**47.4336**	**46.3615**	**62.254**	**1.6945**
JZ1	36.8192	28.3773	10.1382	47.3303	13.2394	18.097	25.9773	39.1811	1.0996
JZ2	36.4073	27.9101	9.6774	47.0244	12.179	16.6702	23.4868	37.979	1.0455
Streaming SMC Dataset Averages									
Essentia	36.71	28.52	10.08	47.29	11.59	15.51	22.37	34.87	0.9209
IBT Offline	28.19	21.58	5.824	42.66	10.16	14.23	16.77	25.7	0.7817
IBT Online	25.54	19.51	4.67	42.63	7.857	11.88	12.88	21.7	0.6836
Madmom Beat Detector	33.26	26.48	8.869	43.92	14.16	19.61	22.76	33.23	1.102
CRF Beat Detector	42.86	34.12	8.519	48.86	12	16.61	20.12	34.31	0.9897
Madmom Beat Tracker	51.27	41.06	14.59	59.59	26.58	34.4	32.67	46.56	1.292
DBN Beat Tracker	**53.79**	**42.98**	**20.99**	**63.71**	**34.27**	**44.28**	**45.79**	**60.35**	**1.628**

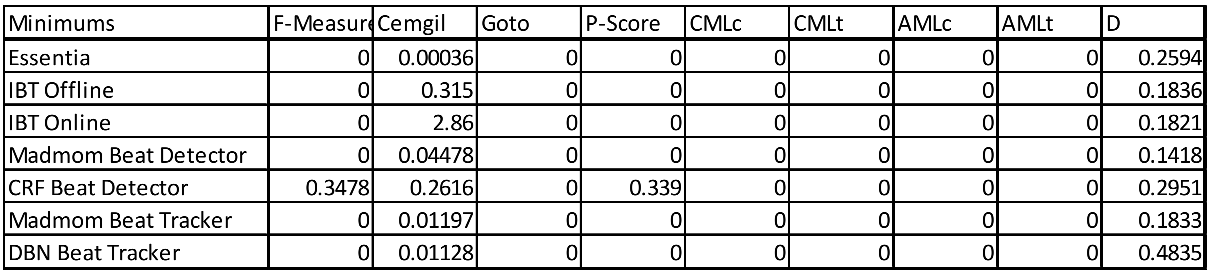
Table6. This table shows the MIREX2016 Beat Tracking event’s results on the SMC dataset, and the test beat trackers’ streaming averages on the SMC dataset stream. Highest scores are highlighted red and second highest scores are highlighted green.

4.2.4. Min/Max analysis

Whilst the previous figures show only the mean accuracies, Table7 and Table8 show the minimum and maximum evaluation scores for each beat tracker. They show that the evaluation values for each beat tracker ranged from roughly 0% accuracy to roughly 100% accuracy in all evaluation measures. In conjunction with Fig1 this shows that while the DBN Beat Tracker had an average accuracy of 72.8, its evaluations could range from 0% to 100%. This shows that even IBT had the potential to have a high accuracy if it was applied to the right audio clips. However, determining which audio clips result in such a high accuracy would require data analysis beyond the scope of this study.

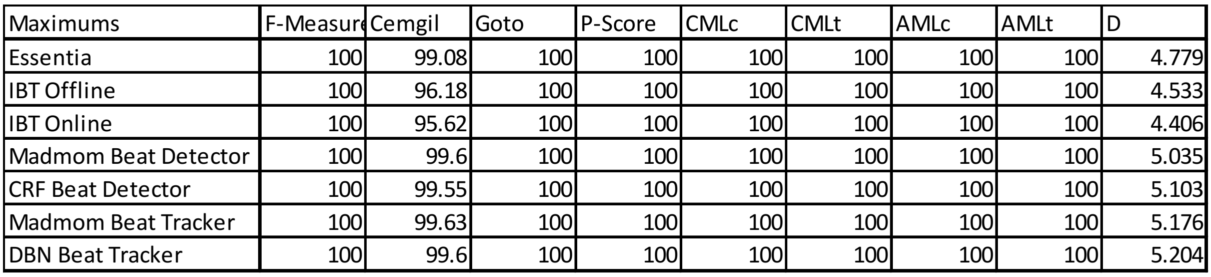
Looking closer at the Cemgil and D measures showed that IBT and CRF Beat Detector had the highest minimum Cemgil scores, whilst Essentia, Madmom Beat Tracker and DBN Beat Tracker had the lowest. This was at odds with the minimum D scores, for which the DBN Beat Tracker, CRF Beat Detector and Essentia had the highest minimum scores. Interestingly, the CRF Beat Detector was the only beat tracker with a non-zero F-Measure and P-Score minimum value. This could indicate that the CRF Beat Detector had a higher lower bound than the other beat trackers, however, more testing would need to be done to confirm this as it could easily be an artefact of the testing process.

Looking at the Cemgil and D evaluations' maximums, they seemed to match the non-streaming results when it came to beat tracker hierarchy. The INESC-Porto Beat Tracker had the lowest maximums whilst the DBN Beat Tracker and Madmom Beat Tracker had the highest. Interestingly the Madmom and CRF Beat Detectors had maximums close to those of the DBN Beat Tracker. This is likely due to their capacity to score highly in non-streaming scenarios also allowing them to score highly in streaming scenarios under very specific circumstances, such as a stream containing relatively constant tempo across all its audio clips.



Minimums	F-Measur	Cemgil	Goto	P-Score	CMLc	CMLt	AMLc	AMLt	D
Essentia	0	0.00036	0	0	0	0	0	0	0.2594
IBT Offline	0	0.315	0	0	0	0	0	0	0.1836
IBT Online	0	2.86	0	0	0	0	0	0	0.1821
Madmom Beat Detector	0	0.04478	0	0	0	0	0	0	0.1418
CRF Beat Detector	0.3478	0.2616	0	0.339	0	0	0	0	0.2951
Madmom Beat Tracker	0	0.01197	0	0	0	0	0	0	0.1833
DBN Beat Tracker	0	0.01128	0	0	0	0	0	0	0.4835

Table7. This table shows the minimum evaluation measures for each beat tracker.



Maximums	F-Measur	Cemgil	Goto	P-Score	CMLc	CMLt	AMLc	AMLt	D
Essentia	100	99.08	100	100	100	100	100	100	4.779
IBT Offline	100	96.18	100	100	100	100	100	100	4.533
IBT Online	100	95.62	100	100	100	100	100	100	4.406
Madmom Beat Detector	100	99.6	100	100	100	100	100	100	5.035
CRF Beat Detector	100	99.55	100	100	100	100	100	100	5.103
Madmom Beat Tracker	100	99.63	100	100	100	100	100	100	5.176
DBN Beat Tracker	100	99.6	100	100	100	100	100	100	5.204

Table8. This table shows the maximum evaluation measures for each beat tracker.

Unfortunately, these minimums and maximums were evaluation specific, so any testing errors resulting in an anomalous 0 or 100 would directly affect these statistics. As a result, it is unknown whether these minimums and maximums accurately reflect each beat tracker's range or if they are merely collections of anomalous readings. A much more in depth study of the individual audio clip evaluations would be required to determine if these statistics are anomalous or not.

4.2.5. Comparison of the Madmom Beat Trackers

Looking closer at the results of the four madmom beat trackers (Beat Detector, CRF Beat Detector, Beat Tracker, DBN Beat Tracker) showed some similarities. This was to be expected as all four beat trackers originally came from the same source paper (Böck and Schedl, 2011) which described a single beat tracker with two variants. The difference between the two variants was that one, named Beat Tracker, kept track of varying tempo whilst the other, Beat Detector, assumed a constant tempo. These two variations are the core of the Madmom Beat Tracker and Madmom Beat Detector that were tested in this study. They were also the base for the DBN Beat Tracker (Böck, Krebs and Widmer, 2014) and the CRF Beat Detector (Korzeniowski, Böck and Widmer, 2014). This explains all four beat trackers' similar performance in non-streaming scenarios, and it allows comparisons between them to show how effective the CRF and DBN modifications were.

Looking back at Fig1 and Fig2 clearly shows that whilst the Beat Trackers and Beat Detectors all had similar accuracy in the non-streaming scenario, the Beat Detectors did a lot worse in the streaming scenario than the Beat Trackers. This is no doubt because the Beat Detectors' assumption that tempo is roughly constant did not hold up in the streaming scenarios, as tempo is likely to vary from audio clip to audio clip. Since audio clips of a single genre tend to share a common tempo, the CRF Beat Detector was likely to have much less trouble extracting beats from a genre stream than from the more varying dataset streams. Looking at the Ballroom dataset's genre stream results(Table9) showed that the CRF Beat Detector scored higher than the DBN Beat Tracker despite the streaming scenario. A Z-Test was performed to determine if the CRF Beat Detector's accuracy on the Ballroom Genre stream deviated significantly from the DBN Beat Tracker's. The mean Z-Score produced was -0.692, which showed that overall the two beat trackers' accuracies do not differ significantly. This is contrary to the results in Table5, which showed a very significant difference between the two beat trackers in streaming scenarios. This conflict suggests that the genre streaming was the cause of the CRF Beat Detector's accuracy increase, which suggests that it is possible for the CRF Beat Detector to detect beats accurately in streaming datasets when the stream contains audio clips of similar tempo. However, it should be noted that this was heavily dependent on the dataset, as the Ballroom genre stream was the only streaming scenario in which the CRF Beat Detector scored similarly to the DBN Beat Tracker. The genre streams of the other two datasets did not show the same results (see Table10 and Table11).

Mean Madmom evaluations for the Ballroom Genre Stream	F-Measure	Cemgil	Goto	P-Score	CMLc	CMLt	AMLc	AMLt	D
Madmom Beat Detector Ballroom Genre Stream Averages	72.89	69.33	62.37	70.03	60.91	62.48	86.59	88.51	3.331
CRF Beat Detector Ballroom Genre Stream Averages	92.09	87.44	85.33	90.16	84.96	88	87.23	90.46	3.42
Madmom Beat Tracker Ballroom Genre Stream Averages	91.11	87.05	77.48	87.92	79.99	81.64	88.43	90.3	3.534
DBN Beat Tracker Ballroom Genre Stream Averages	91.07	86.54	83.63	87.98	80.62	81.94	92.44	93.84	3.534

Table9. This table shows the average evaluations of each Madmom beat tracker operating on the Ballroom dataset's genre stream. The highest score for each evaluation is highlighted in red.

Mean Madmom evaluations for the GTZAN Genre Stream	F-Measure	Cemgil	Goto	P-Score	CMLc	CMLt	AMLc	AMLt	D
Madmom Beat Detector GTZAN Genre Stream Averages	57.25	49.15	28.85	55.66	31.96	36.2	52.44	59.47	2.315
CRF Beat Detector GTZAN Genre Stream Averages	74.47	66.22	38.33	70.67	47.31	51.83	62.39	70.43	2.605
Madmom Beat Tracker GTZAN Genre Stream Averages	82.3	73.35	48.68	78.96	59.83	63.66	75.31	81.23	2.926
DBN Beat Tracker GTZAN Genre Stream Averages	82.1	74.13	57.5	78.88	61.15	64.43	80.22	85.14	2.969

Table10. This table shows the average evaluations of each Madmom beat tracker operating on the GTZAN dataset's genre stream. The highest score for each evaluation is highlighted in red.

Mean Madmom evaluations for the SMC Genre Stream	F-Measure	Cemgil	Goto	P-Score	CMLc	CMLt	AMLc	AMLt	D
Madmom Beat Detector SMC Genre Stream Averages	34.49	27.69	11.03	44.99	16.14	21.34	24.91	34.83	1.158
CRF Beat Detector SMC Genre Stream Averages	42.67	34.01	9.386	48.42	11.81	15.72	20.08	33.98	0.9699
Madmom Beat Tracker SMC Genre Stream Averages	50.93	40.91	14.92	59.88	27.06	35.31	32	45.52	1.297
DBN Beat Tracker SMC Genre Stream Averages	53.54	42.78	19.19	64.02	33.87	44.59	45.15	60.54	1.637

Table11. This table shows the average evaluations of each Madmom beat tracker operating on the SMC dataset's genre stream. The highest score for each evaluation is highlighted in red.

Whilst comparing the Beat Trackers to the Beat Detectors clearly showed the Beat Trackers were better suited to stream testing, comparing the CRF and DBN beat trackers to the originals also showed a performance increase. Looking at Fig1 and Fig2 shows the CRF Beat Detector and DBN Beat Tracker consistently scored higher than the original Beat Detector and Beat Tracker. It also shows that the CRF Beat Detector took much less of a performance hit than the Madmom Beat Detector in streaming scenarios. Looking back at Table2 shows the DBN Beat Tracker took more of a performance hit than the Madmom Beat Tracker for most of the evaluations. The only evaluation measures for which the DBN Beat Tracker had a less significant drop than the Madmom Beat Tracker were the Goto and AMLc/t evaluations. This could suggest that the improved Dynamic Bayesian Network used in the DBN Beat Tracker caused a larger performance decrease in streaming scenarios, however, further research would be required to fully explore this possibility. Since the DBN Beat Tracker outperformed the Madmom Beat Tracker overall in streaming scenarios, regardless of the performance drop, it seems that it would be the preferred beat tracker to use in streaming scenarios.

4.2.6. INESC-Porto Beat Tracker

The INESC-Porto Beat Tracker (IBT for short) was developed in (Oliveira et al., 2012) and was the only beat tracker we examined that had previously been tested in a streaming scenario. As a result, its low performance was quite surprising, producing the worst accuracies in both streaming and non-streaming scenarios (see Fig1). On top of this its accuracy decreased more than the Madmom Beat Trackers and Essentia Multifeature, with only the Madmom Beat Detectors losing more accuracy in streaming scenarios than IBT.

Looking at the IBT's two modes, Online and Offline, Fig1 showed that whilst the Online mode scored higher in non-streaming scenarios it scored lower in streaming scenarios than the Offline version. Looking closer using Fig3 showed that IBT Offline took a much smaller performance hit in the genre streams than IBT Online, however, they both lost approximately the same amount of accuracy on the dataset streams. Fig3 also showed that both Online and Offline modes had roughly the same accuracy on dataset streams. This, combined with the noticeable difference in genre stream accuracies, suggested that the Offline version could take advantage of audio streams with relatively consistent tempo, whilst the Online version couldn't.

4.3. Accuracy Variation Over Time

This section looks at how each beat tracker performed over time during the audio streams. This was done by looking at the average accuracy of all clips in audio stream position one, two, three, and so on. These positional averages were created the same way as the genre and dataset stream averages, resulting in a set of average accuracies for each possible stream position. These averages were then used to examine how each beat tracker's accuracy varied over time throughout an audio stream. These average accuracies are presented in Fig6 which shows the average accuracy for audio clip position 1 through 5 along the X axis to clearly show how each beat tracker's accuracy varied over time.

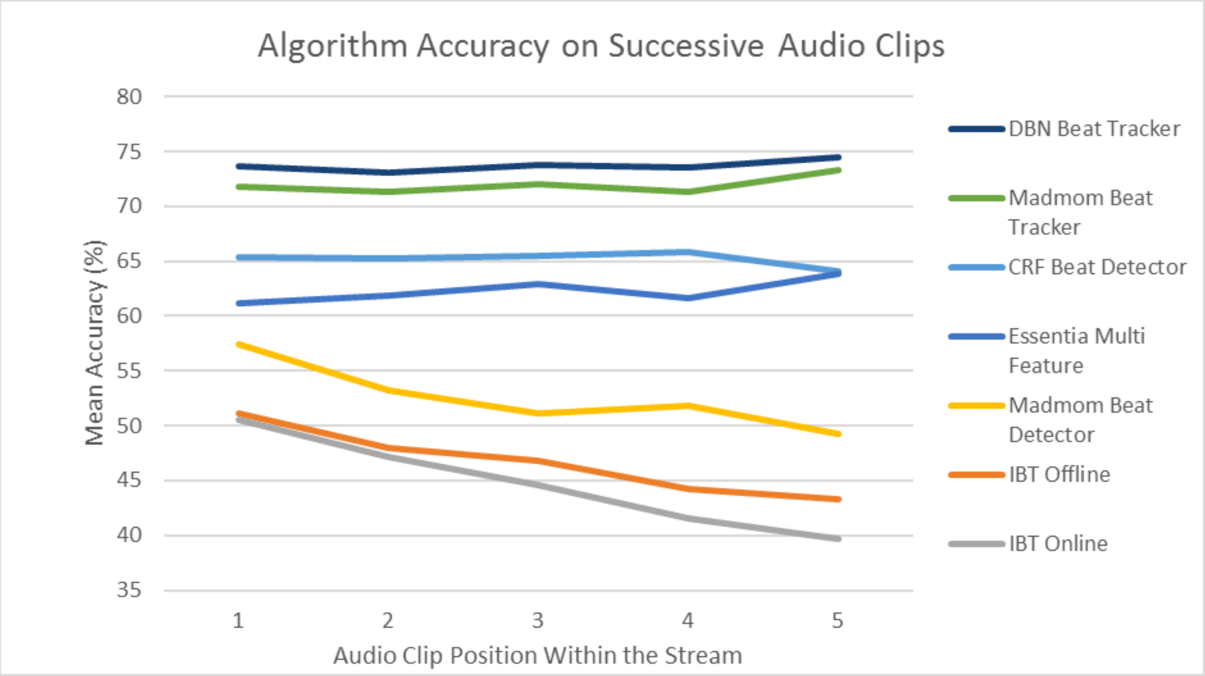


Fig6. This graph shows the average accuracy of each beat tracker on each of the five audio clips that make up every stream file. Each line shows how a given beat tracker's accuracy varies over time, with the start of the audio stream on the left at audio clip 1 and the end of the stream on the right with audio clip 5.

Fig6 suggests that the DBN Beat Tracker, Madmom Beat Tracker, Essentia Multifeature Beat Tracker, and CRF Beat Detector maintained approximately constant accuracy throughout the five audio clips. However, the Madmom Beat Detector and IBT accuracies appear to decrease over time. To confirm that these differences were statistically significant a series of Z-Tests were performed.

These Z-Tests were designed to determine if a beat tracker's accuracy on an audio clip deviated significantly from its accuracy on the previous audio clip in the stream. The Z-Tests used the mean and standard deviation of the individual evaluation scores for the two adjacent audio clips. Then a mean Z-Score was calculated to give an average Z-Score for each beat tracker and audio clip pair. These Z-Scores were evaluated against the critical values of a standard two tailed Z-Test with a confidence threshold of 95%, then again for a confidence threshold of 99%. As a result, any Z-Score above 2.5758 or below -2.5758 was considered highly significant, whilst any Z-Score above -1.96 or below 1.96 was considered insignificant. A significant value indicated that a given audio clip's accuracy deviated from that of the audio clip that preceded it, whilst an insignificant value indicated that there was no significant difference between the two. The mean Z-Scores for each beat tracker are shown in Table12, with partially significant values highlighted in green and highly significant values in red.

Z-Scores	1->2	2->3	3->4	4->5
Essentia	0.792626473	1.171988	-1.47264	2.512397
IBTOffline	**-3.654406473**	-1.45583	**-2.89442**	-1.10155
IBTOnline	**-3.708549091**	**-3.16676**	**-3.85348**	-2.17121
Beat Detector	**-4.347702605**	-2.1642	0.653645	**-2.64911**
CRF	-0.172716611	0.369922	0.25157	-1.65521
Beat Tracker	-0.466263706	0.826149	-0.89293	2.225674
DBN	-0.500994681	0.726615	-0.29559	1.159778

Table12. This table shows the average Z-Scores of each beat tracker when comparing each audio clip to its predecessor from the audio stream. Z-Scores outside the 95% confidence threshold are highlighted in green, and Z-Scores outside the 99% confidence threshold are highlighted in red. A negative Z-Score indicates that the latter of the two audio clips had a lower accuracy than the former.

The CRF Beat Detector and DBN Beat Tracker's Z-Scores shown in Table12 contained no significant values. Hence these two beat trackers' accuracies did not vary over time during the stream datasets, which is backed up by Fig6. The Madmom Beat Tracker and Essentia Beat Tracker also had mostly insignificant differences, however, they each showed a significant accuracy increase on the final audio clip. Due to the relative insignificance of the accuracy differences between earlier audio clips, and the fact that both beat trackers showed both accuracy increases and decreases, it was unlikely that this indicated an accuracy increase over time. Instead it was more likely that the audio streams' final audio clips were on average easier for these beat trackers to extract beats from than the previous clips. Further testing would be required to confirm this, as it is also plausible that this significant increase was inherent to the beat trackers themselves and not an artefact of the audio streams.

Looking at the INESC-Porto Beat Tracker's Z-Scores in Table12 showed that its Online mode's accuracy decreased significantly in every instance, whereas its Offline mode wavered between significant and insignificant decreases. Whilst half of its Z-Scores suggested that the accuracy decrease was insignificant, they did still suggest an accuracy decrease. Combined with the clear accuracy decrease of the Online mode the data suggests that the Offline mode also lost accuracy over time, but that it did so less severely. Further testing would be required to determine the precise nature of the accuracy decrease in both cases.

Finally, looking at the Madmom Beat Detector's Z-Scores showed that the majority suggested a significant accuracy decrease. The presence of an accuracy increase between audio clips 3 and 4 could indicate that the accuracy did not consistently decrease over time. However, the insignificance of the accuracy increase compared to the significance of the decreases suggested that the accuracy increase may simply be an anomaly of the audio streams. Further tests would be required to confirm this, and to determine the precise nature of the decrease.

4.4. Conclusions

From looking at the average accuracies it was clear that the DBN Beat Tracker and Madmom Beat Tracker had the highest accuracy of all the beat trackers in both streaming and non-streaming scenarios. The CRF Beat Detector and Madmom Beat Detector, whilst performing similarly to the DBN and Madmom Beat Trackers in non-streaming scenarios, performed much worse in streaming scenarios. The INESC-Porto Beat Tracker, which had previously been tested in streaming scenarios, showed the lowest accuracies of all tested beat trackers in both streaming and non-streaming scenarios. Its Online and Offline modes showed some slight differences, with the Online mode performing better in non-streaming scenarios whilst the Offline mode performed better in streaming scenarios. Finally, the Essentia Multifeature Beat Tracker showed very middling performance in both streaming and non-streaming scenarios, its most notable trait being that its streaming performance was very similar to its non-streaming performance. This suggested that operating on audio streams impacted its accuracy less than it impacted some of the other beat trackers.

Looking closer at each beat tracker's accuracy over time showed that the INESC-Porto Beat Tracker and Madmom Beat Detector exhibited a steady accuracy decrease in streaming scenarios. The CRF Beat Detector and DBN Beat Tracker did not significantly gain or lose accuracy over time, maintaining roughly constant accuracy across all audio clips in the stream files. Interestingly, the Madmom Beat Tracker and Essentia Multifeature Beat Tracker's accuracies increased significantly at the end of the audio streams. Further study would be required to determine the cause of this accuracy increase, as it could indicate a quirk of the beat trackers or it could be an anomaly of the audio streams themselves.

5. Discussion

Whilst the results obtained in this study have been useful for evaluating the performance of several beat trackers in audio streaming scenarios, the study has several limitations which will be discussed in this section.

The main limitation of this study was the small number of beat trackers that were tested. By only examining the effects of streaming audio on six beat trackers the results could not be used to adequately evaluate beat tracking as a whole. Furthermore, since four of the beat trackers (Madmom Beat Detector, CRF Beat Detector, Madmom Beat Tracker, and DBN Beat Tracker) were all derived from the same base beat tracker, only a small subsection of beat tracking methods were included in this evaluation. As a result, further research that tests a much wider array of beat trackers would be needed to make any judgements about the current state of the art in beat tracking when it comes to audio streams.

As well as a small number of beat trackers this study also used a small number of beat tracking datasets, those being the SMC, Ballroom, and GTZAN datasets. As a result, it is likely that certain sub-genres of music went untested, due to the limited size of each dataset. It is recommended that further study include more datasets to better cover the full range of musical styles.

The audio streams that were created from the three datasets also had several limitations. Since all audio stream files were roughly the same length it was not possible to study the effects that different lengths of audio stream would have on the beat trackers. Furthermore, since the audio streams were all made up of only five audio clips it was quite difficult to determine if an accuracy change between two subsequent audio clips was caused by a trend within the beat tracker or if it was an anomaly of the audio streams themselves. Only the most severe accuracy changes over time could be gleaned from the data, and further study using longer audio clips would be required to locate smaller trends or identify the exact nature of the larger ones. Therefore, it is recommended that further research into how beat tracker accuracy changes over time should use much longer stream files than those used in this study.

A more significant problem with the stream files was caused by the stream file creation process. The stream files were created in batches of 100 by randomly selecting audio files from a given range to make up the audio stream. Since the ranges that were used did not contain the same number of audio clips, some clips were inevitably included in more audio streams than others. This resulted in different audio clips having different weights during the stream testing, potentially skewing the results. This problem was exacerbated during initial averages due to there being more genre streams than dataset streams, which resulted in the genre streams dominating the averages. Later averages accounted for this by only including each audio file once, rather than once for each time it was present in a stream, but the problem of the streaming files themselves favouring certain files over others was discovered too late to rectify. As a result, it is highly recommended that further testing use a more controlled process for creating the audio streams. Making sure that each audio file had the same weight in the audio stream would avoid the stream skewing in favour of a specific file, and it should better represent the overall body of audio clips, making accuracy comparisons between audio streams and audio clips more accurate.

Another potential improvement to the stream testing would have been the inclusion of an audio stream that took audio clips from all three datasets. Such an audio stream could potentially contain even more internal variation than the dataset streams that were used. As a result, such a stream would be the hardest stream to extract beats from, and may be used to show each beat trackers' accuracy under even harsher conditions than those present in this study.

Finally, there were two areas of research that this study did not touch upon. This study did not look at the effects of specific genres of music on the beat trackers' streaming and non-streaming accuracy. It also did not record the running times of each beat tracker, so further research would be required to determine if beat trackers take longer to extract beats in streaming scenarios than in non-streaming scenarios.

6. Conclusions

During this study, we looked at how audio streams affected the performance of several beat trackers. This was done by running the chosen beat trackers on all audio files in the GTZAN, Ballroom, and SMC datasets, and then running them again on audio streams made of audio clips taken from these datasets. The beat detections produced from both streaming and non-streaming tests were then evaluated using the standard beat tracking evaluations used in the yearly MIREX beat tracking task. The evaluations in the non-streaming scenario were then compared to the evaluations from the streaming scenario for each beat tracker to observe how their performance differed between the two.

The results obtained clearly showed that the DBN Beat Tracker and Madmom Beat Tracker had the highest overall accuracy in streaming scenarios, and they were also the least affected by the audio streams of all the tested beat trackers. The CRF Beat Detector and Madmom Beat Detector, which had similar performace to the DBN and Madmom Beat Trackers in the non-streaming scenario, were affected the most by the streaming scenario, with average accuracy drops of roughly 9% and 20% respectively. The INESC-Porto Beat Tracker performed the worst of all tested beat trackers in both streaming and non-streaming scenarios. This was particularly notable as it was originally designed to handle audio transitions, and was the only beat tracker that had previously been tested in a streaming scenario, so being more affected by the audio streams than some of the other beat trackers was highly unexpected.

Each beat tracker's streaming accuracy was then closely examined, this time looking at how their average accuracy varied over time. This examination showed that the INESC-Porto Beat Tracker and Madmom Beat Detector both lost accuracy over time in the streaming scenario, whilst the rest of the beat trackers maintained a fairly constant accuracy. There was some indication that the Essentia Multi Feature Beat Tracker and Madmom Beat Tracker gained accuracy at the end of audio streams, but further research would be required to determine if this was the case.

Following this study there are various areas that have yet to be fully explored. Due to the small number of tested beat trackers, this study is far from a comprehensive overview of beat tracking as a whole. A larger study would be needed to get a better picture of how the current state of the art in beat tracking handles audio streams. Further research with longer audio streams would also be useful to get a better view of how beat tracker accuracy varies over time and to see how audio stream length affects beat tracker accuracy. Finally, this study has not looked at how different musical genres affect beat tracker accuracy in audio streams, so further research in this area may be useful.

7. Bibliography

Music-ir.org. (2016). *2016:Audio Beat Tracking - MIREX Wiki*. [online] Available at: http://www.music-ir.org/mirex/wiki/2016:Audio_Beat_Tracking [Accessed 23 Apr. 2017].

Music-ir.org. (2016). *2016:Main Page - MIREX Wiki*. [online] Available at: http://www.music-ir.org/mirex/wiki/2016:Main_Page [Accessed 27 Apr. 2017].

Allen, P. and Dannenberg, R. (1990). Tracking Musical Beats in Real Time. In: *Proceedings of the 1990 International Computer Music Conference*. Glasgow: ICMA, pp.140-143.

Store.steampowered.com. (2008). *AudioSurf on Steam*. [online] Available at: http://store.steampowered.com/app/12900/ [Accessed 25 Apr. 2017].

Bello-Correa, J. (2003). *Towards the automated analysis of simple polyphonic music: A knowledge-based approach*. Ph.D. Department of Electronic Engineering, Queen Mary, University of London.

Böck, S. (2013). *CPJKU/madmom*. [online] GitHub. Available at: https://github.com/CPJKU/madmom [Accessed 23 Apr. 2017].

Böck, S. (2016). MIREX 2016 submission SB8. In: *Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX'16)*. [online] Available at: http://www.music-ir.org/mirex/abstracts/2016/SB8.pdf [Accessed 23 Apr. 2017].

Böck, S. (2016). MIREX 2016 submission SB9. In: *Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX'16)*. [online] Available at: http://www.music-ir.org/mirex/abstracts/2016/SB9.pdf [Accessed 23 Apr. 2017].

Böck, S. and Korzeniowski, F. (2016). MIREX 2016 submission BK3. In: *Music Information Retrieval Evaluation Exchange (MIREX'16)*. [online] Available at: http://www.music-ir.org/mirex/abstracts/2016/BK3.pdf [Accessed 23 Apr. 2017].

Böck, S. and Krebs, F. (2016). MIREX 2016 submission BK1. In: *Music Information Retrieval Evaluation eXchange (MIREX'16)*. [online] Available at: http://www.music-ir.org/mirex/abstracts/2016/BK1.pdf [Accessed 23 Apr. 2017].

Böck, S. and Schedl, M. (2011). Enhanced beat tracking with context aware neural networks. In: *Proceedings of the International Conference on Digital Audio Effects (DAFx)*. Paris, France.

Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F. and Widmer, G. (2016). madmom: a new Python Audio and Music Signal Processing Library. *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*, pp.1174-1178.

Böck, S., Krebs, F. and Widmer, G. (2014). A multi-model approach to beat tracking considering heterogeneous music styles. In: *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*. Taipei.

Böck, S., Krebs, F. and Widmer, G. (2015). Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In: *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*. Malaga, Spain.

Böck, S., Krebs, F., Korzeniowski, F. and Widmer, G. (2015). MIREX 2015 submissions. In: *Music Information Retrieval Evaluation eXchange (MIREX'15)*. [online] Available at: http://www.music-ir.org/mirex/abstracts/2015/BK1.pdf [Accessed 23 Apr. 2017].

Bramwell, T. (2008). *Audiosurf tops February Steam charts*. [online] Eurogamer.net. Available at: http://www.eurogamer.net/articles/audiosurf-tops-february-steam-charts [Accessed 25 Apr. 2017].

Cemgil, A., Kappen, B., Desain, P. and Honing, H. (2000). On tempo tracking: Tempogram Representation and Kalman filtering. *Journal of New Music Research*, 29(4), pp.259-273.

Clark, R. and Wilson, H. (2015). *Crypt of the NecroDancer*. Brace Yourself Games.

Collins, N. (2006). *Towards autonomous agents for live computer music: Realtime machine listening and interactive music systems*. Ph.D. Department of Music, University of Cambridge.

Daniels, M. (2014). Tempo Estimation and Causal Beat Tracking using Ensemble Learning. In: *Music Information Retrieval Evaluation Exchange (MIREX'14)*. [online] Available at: http://www.music-ir.org/mirex/abstracts/2014/MD2.pdf [Accessed 23 Apr. 2017].

Dansart, L. (2016). *Melody's Escape*. Icetesy SPRL.

Davies, M., Degara, N. and Plumbley, M. (2011). Measuring the Performance of Beat Tracking Algorithms Using a Beat Error Histogram. *IEEE Signal Processing Letters*, 18(3), pp.157-160.

Davies, M., Degara, N. and Plumbley, M. (2009). *Evaluation methods for musical audio beat tracking algorithms*. Queen Mary University of London.

Degara, N., Rua, E., Pena, A., Torres-Guijarro, S., Davies, M. and Plumbley, M. (2012). Reliability-Informed Beat Tracking of Musical Signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1), pp.290-301.

Di Giorgi, B., Zanoni, M. and Sarti, A. (2014). Multipath Beat Tracking. In: *Music Information Retrieval Evaluation Exchange (MIREX'14)*. [online] Available at: http://www.music-ir.org/mirex/abstracts/2014/DZ1.pdf [Accessed 23 Apr. 2017].

Essentia.upf.edu. (2013). *ESSENTIA*. [online] Available at: http://essentia.upf.edu/ [Accessed 23 Apr. 2017].

Eyben, F., Weninger, F., Ferroni, G. and Schuller, B. (2013). Tempo Estimation and Beat Tracking with Long Short-Term Memory Neural Networks and Comb-Filters. In: *Music Information Retrieval Evaluation Exchange (MIREX'13)*. [online] Available at: http://www.music-ir.org/mirex/abstracts/2013/EWFS1.pdf [Accessed 23 Apr. 2017].

Fitterer, D. (2008). *Audiosurf*. Valve Corporation.

Goto, M. and Muraoka, Y. (1997). Issues in evaluating beat tracking systems. *Working Notes of the IJCAI-97 Workshop on Issues in AI and Music - Evaluation and Assessment*, pp.9-16.

Hainsworth, S. (2004). *Techniques for the automated analysis of musical audio*. Ph.D. Department of Engineering, Cambridge University.

Holzapfel, A., Davies, M., Zapata, J., Oliveira, J. and Gouyon, F. (2012). Selective Sampling for Beat Tracking Evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9), pp.2539-2548.

Holzmann, G. (2009). *Music Information Retrieval Datasets*. [online] Grh.mur.at. Available at: http://grh.mur.at/sites/default/files/mir_datasets_0.html [Accessed 23 Apr. 2017].

Smc.inesctec.pt. (2012). *IBT*. [online] Available at: http://smc.inesctec.pt/technologies/ibt/ [Accessed 23 Apr. 2017].

Madmom.readthedocs.io. (2015). *Installation — madmom 0.15.dev0 documentation*. [online] Available at: https://madmom.readthedocs.io/en/latest/installation.html [Accessed 24 Apr. 2017].

Essentia.upf.edu. (2013). *Installing Essentia*. [online] Available at: http://essentia.upf.edu/documentation/installing.html [Accessed 23 Apr. 2017].

Klapuri, A., Eronen, A. and Astola, J. (2006). Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1), pp.342-355.

Korzeniowski, F., Böck, S. and Widmer, G. (2014). Probabilistic extraction of beat positions from a beat activation function. In: *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*. Taipei.

Krebs, F., Böck, S. and Widmer, G. (2013). Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio. In: *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*. Curitiba, Brazil.

Krebs, F., Böck, S. and Widmer, G. (2015). An efficient statespace model for joint tempo and meter tracking. In: *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*. Malaga, Spain.

Lerch, A. (2017). *datasets | Audio Content Analysis*. [online] Audiocontentanalysis.org. Available at: http://www.audiocontentanalysis.org/data-sets/ [Accessed 23 Apr. 2017].

Longuet-Higgins, H. (1976). Perception of melodies. *Nature*, 263(5579), pp.646-653.

Marchand, U. and Peeters, G. (2015). Swing Ratio Estimation. In: *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx-15).*

Marchini, M. and Purwins, H. (2011). Unsupervised Analysis and Generation of Audio Percussion Sequences. *Lecture Notes in Computer Science*, 6684, pp.205– 218.

Mauch, M. and Dixon, S. (2010). Simultaneous Estimation of Chords and Musical Context From Audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6), pp.1280-1289.

McKinney, M., Moelants, D., Davies, M. and Klapuri, A. (2007). Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms. *Journal of New Music Research*, 36(1), pp.1-16.

Nema.lis.illinois.edu. (2012). *MIREX 2012: Audio Beat Tracking - MIREX06 MCK Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2012/results/abt/mck/summary.html [Accessed 24 Apr. 2017].

Nema.lis.illinois.edu. (2012). *MIREX 2012: Audio Beat Tracking - MIREX09 MAZ Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2012/results/abt/maz/summary.html [Accessed 24 Apr. 2017].

Nema.lis.illinois.edu. (2012). *MIREX 2012: Audio Beat Tracking - MIREX12 SMC Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2012/results/abt/smc/summary.html [Accessed 24 Apr. 2017].

Nema.lis.illinois.edu. (2013). *MIREX 2013: Audio Beat Tracking - MIREX06 MCK Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2013/results/abt/mck/summary.html [Accessed 24 Apr. 2017].

Nema.lis.illinois.edu. (2013). *MIREX 2013: Audio Beat Tracking - MIREX09 MAZ Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2013/results/abt/maz/summary.html [Accessed 24 Apr. 2017].

Nema.lis.illinois.edu. (2013). *MIREX 2013: Audio Beat Tracking - MIREX12 SMC Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2013/results/abt/dav/summary.html [Accessed 24 Apr. 2017].

Nema.lis.illinois.edu. (2014). *MIREX 2014: Audio Beat Tracking - MIREX06 MCK Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2014/results/abt/mck/summary.html [Accessed 24 Apr. 2017].

Nema.lis.illinois.edu. (2014). *MIREX 2014: Audio Beat Tracking - MIREX09 MAZ Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2014/results/abt/maz/summary.html [Accessed 24 Apr. 2017].

Nema.lis.illinois.edu. (2014). *MIREX 2014: Audio Beat Tracking - MIREX12 SMC Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2014/results/abt/smc/summary.html [Accessed 24 Apr. 2017].

Nema.lis.illinois.edu. (2016). *MIREX 2016: Audio Beat Tracking - MIREX06 MCK Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2016/results/abt/mck/summary.html [Accessed 27 Apr. 2017].

Nema.lis.illinois.edu. (2016). *MIREX 2016: Audio Beat Tracking - MIREX09 MAZ Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2016/results/abt/maz/summary.html [Accessed 27 Apr. 2017].

Nema.lis.illinois.edu. (2016). *MIREX 2016: Audio Beat Tracking - MIREX12 SMC Dataset - Summary*. [online] Available at: http://nema.lis.illinois.edu/nema_out/mirex2016/results/abt/smc/summary.html [Accessed 23 Apr. 2017].

Nguyen, D. (2014). *15 billion songs have been identified by music recognition service Shazam - Play | siliconrepublic.com - Ireland's Technology News Service*. [online] Silicon Republic. Available at: https://www.siliconrepublic.com/play/15-billion-songs-have-been-identified-by-music-recognition-service-shazam [Accessed 25 Apr. 2017].

Oliveira, J., Davies, M., Gouyon, F. and Reis, L. (2012). Beat Tracking for Multiple Applications: A Multi-Agent System Architecture With State Recovery. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(10), pp.2696-2706.

Oliveira, J., Davies, M., Gouyon, F. and Reis, L. (2012). MIREX 2012 Audio Beat Tracking Submission: IBT. In: *Music Information Retrieval Evaluation Exchange (MIREX'12)*. [online] Available at: http://www.music-ir.org/mirex/abstracts/2012/ODGR1.pdf [Accessed 23 Apr. 2017].

Oliveira, J., Gouyon, F., Martins, L. and Reis, L. (2010). IBT: a real-time tempo and beat tracking system. In: *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*. Utrecht, Netherlands.

Raffel, C. (2014). *mir_eval*. [online] GitHub. Available at: https://github.com/craffel/mir_eval [Accessed 23 Apr. 2017].

Raffel, C. (2015). *MIR Datasets*. [online] Colinraffel.com. Available at: http://colinraffel.com/wiki/mir_datasets [Accessed 23 Apr. 2017].

Rafii, Z. and Pardo, B. (2013). REpeating Pattern Extraction Technique (REPET): A Simple Method for Music/Voice Separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(1), pp.73-84.

Shazam.com. (2016). *Shazam*. [online] Available at: https://www.shazam.com/gb/company [Accessed 25 Apr. 2017].

Wilburn, T. (2008). *Catching Waveforms: Audiosurf Creator Dylan Fitterer speaks*. [online] Ars Technica. Available at: https://arstechnica.com/gaming/2008/03/catching-waveforms-audiosurf-creator-dylan-speaks/ [Accessed 25 Apr. 2017].

Zapata, J. (2016). MIREX 2016: Multi Feature Beat Tracker (INF + REG) + Source Separation. In: *Music Information Retrieval Evaluation Exchange (MIREX'16)*. [online] Available at: http://www.music-ir.org/mirex/abstracts/2016/JZ1.pdf [Accessed 23 Apr. 2017].

Zapata, J., Davies, M. and Gomez, E. (2014). Multi-Feature Beat Tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4), pp.816-825.