

FAST vs SLOW: LEARNING TEMPO OCTAVES FROM USER DATA

Jason A. Hockman and Ichiro Fujinaga

Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT)

Distributed Digital Music Archives and Libraries (DDMAL)

Music Technology Area, Schulich School of Music

McGill University, Canada

jason.hockman@mail.mcgill.ca, ich@music.mcgill.ca

ABSTRACT

The widespread use of beat- and tempo-tracking methods in music information retrieval tasks has been marginalized due to undesirable sporadic results from these algorithms. While sensorimotor and listening studies have demonstrated the subjectivity and variability inherent to human performance of this task, MIR applications such as recommendation require more reliable output than available from present tempo estimation models. In this paper, we present a initial investigation of tempo assessment based on the simple classification of whether the music is *fast* or *slow*. Through three experiments, we provide performance results of our method across two datasets, and demonstrate its usefulness in the pursuit of a reliable global tempo estimation.

1. INTRODUCTION

Within the last ten years, beat tracking and tempo induction methods have been significantly improved. Several state-of-the-art methods [1–3] are now capable of identifying and providing reliable beat calculations through difficult passages marked by features such as expressive timing or competing rhythms. However, the usefulness of such methods for information retrieval tasks has been limited due to the unpredictable behavior of these algorithms. While many studies demonstrate musical beat localization for humans to be variable and highly subjective [4–8], MIR applications such as recommendation and harmonic description require more reliable tempo estimates. The most frequent error in this context is the so-called *octave error*, or the halving or doubling of the perceived tempo caused by attributing the driving beat level to a metrical level other than the most predominant pulse.

Identification of the most appropriate tempo octave has been shown to be a difficult problem, as demonstrated in the discrepancy between beat tracking evaluations in which a single tempo octave and multiple tempo octaves are accepted [2, 3, 9, 16]. As metronomic values are not

absolute, they are not well-suited for defining the perceived relative speed of a piece of music. Unfortunately, if a user of a recommendation system were to request slow music labeled 60 BPM, and received music more commonly associated with 120 BPM, they would not be satisfied! This paper presents a novel approach to this problem, by identifying *fast* or *slow* music without the use of a beat tracker, and demonstrates the usefulness of this categorization in selecting the appropriate tempo octave of a given piece of music.

1.1 Background

The selection of tempo octave is most commonly achieved as an embedded step within the framework of the beat- or tempo-tracking task. The general procedure used in most audio tempo-tracking algorithms is comprised of three steps. First, the audio signal undergoes a process of reduction, which simplifies the signal by accentuating prominent signal information such as transients. Second, periodicity analysis is performed on the simplified signal, to extract possible beat periods (i.e., the duration between beat events). Third, the algorithm identifies which period is most likely, and assigns this value as the *tactus*, or the most influential beat, which typically controls the local timing of a musical piece.

The majority of recent efforts in beat tracking have centered on this third step, mostly through attempts to incorporate musical knowledge. Musical knowledge is, in this sense, information of any complexity that is provided to the model that allows it to focus on a particular subset of candidates within the wide variety of possible solutions. This knowledge may take on several forms, from a simple limiting of values to desired candidates, to conditional dependencies between metrical levels and prior decisions. The need for such knowledge comes from the ambiguity faced in analyzing the output of periodicity functions of real signals, which may include intra-measure timing variations (e.g., the swing factor in jazz music), syncopation, and/or global tempo shifts. Inspection of the output of periodicity functions during most musical signals will demonstrate several peaks including both octave-related (e.g., half- or double-time periods) resonances as well as other peaks due to rhythmic complexity and noise; these peaks often overshadow the otherwise steady period.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

© 2010 International Society for Music Information Retrieval.

Therefore, a selection of the tactus based on output energy of a periodicity function alone at each frame will result in a highly unsteady tempo output for many music sources.

To address the tempo octave problem, Goto and Mu-raoka [10] limit the possible period values to those periods whose tempi are within only one octave.

As an alternative to placing strict boundaries on tempo values, both Ellis [1] and Klapuri et al. [2] weigh the output of their periodicity functions with log-Gaussian distributions originally proposed by Parncutt [6]. The motivation behind this approach is to model tactus preferences exhibited during listening tests [5, 6], and it is intended to provide emphasis to tempi positioned around the mean of the distribution.

Davies and Plumbley [3] use a variable-state method that alternates weighting functions based on the observed variation of the autocorrelation output. The purpose of this method is to model the uncertainty of the listening process upon initial contact with the stimulus, and then to constrain the possible values based on prior observations.

Klapuri et al. [2] use a hidden Markov model to extract the temporal evolution of a hidden metrical sequence exhibited in the output of a comb filterbank. The joint-state estimates of the present tactus, tatum, and meter periods are achieved through a first-order Markov process, in which the present filterbank output and transition probabilities between periods are used to generate a probabilistic determination of the present state. Selection of bar-length periodicities and tatum help to reduce incorrect tactus attribution. The strength of this model lies in its ability to reinforce a metrical framework within sections displaying less prevalent metrical observations.

In a method conceptually similar to our own, Xiao et al. [11] use a Gaussian mixture model to capture the timbral characteristics of a given tempo through the association of Mel-frequency cepstral coefficients (MFCCs) to discrete BPM values. While this method was demonstrated to reduce the occurrence of octave errors for the beat tracker presented in [1], its reliance on a discrete BPM values as class labels requires a large amount of ground truth that is difficult to produce due to human subjectivity during data collection.

1.2 Motivation

With the exception of [11], the above methods rely on some form of limiting or weighting curve applied to the output of the periodicity function (e.g., autocorrelation and comb filterbank) to reduce the effects of alternate tempo octaves, but these curves are based on BPM responses which are highly variable due to the subjectivity of the task.

What can actually be inferred about a piece of music from a BPM value? Given that humans choose different levels at which to tap when synchronizing with music, is it plausible that a BPM measure would provide us with information about the speed of a piece? Certainly within a single tempo octave the BPM scale can be very informative, but the plurality of acceptable BPM values across tempo octaves makes an inter-octave comparison of

musical rates less reliable.

In addition, other than [11], all above methods rely exclusively on periodicity functions and relatively few features for determination of BPM and thus tempo octave. Our method relies instead on the assumption that the difference between fast and slow music manifests itself across multiple features.

1.3 Organization of this paper

Section 2 briefly outlines our technique for the determination of a piece of music as fast or slow. Section 3 presents both experimentation and results for our method, as well as the application of our method to tempo-tracking. Section 4 presents discussion, and Section 5 provides conclusions and future work.

2. METHOD

To address the problem of tempo octave estimation, we present a classification-based approach that does not rely on discrete BPM values. Alternatively, the proposed method performs a binary classification using broad categories of human response to the pace of music: *fast* and *slow*. There are several benefits to the proposed classification scheme. Unlike solving for a discrete BPM value, music classification as fast or slow is a binary classification problem that offers higher accuracy than present multi-class solutions (e.g., discrete BPM values). Evaluation methodology and interpretation is greatly simplified without acceptance of multiple metrical levels. In addition, ground truth—in this case class labels created through listener response to music—is more readily available for this particular problem.

The proposed technique has two immediate applications: first, as a feature within another retrieval task, and second, as a component within a tempo-tracker that guides the algorithm to the more appropriate of two tempo ranges. While the taxonomy of fast or slow is not precisely analogous to a specific BPM range, we propose that the tempo range can roughly be divided in half to accommodate two tempo octaves. With a training set approximately covering several musical styles in both fast and slow categories, a mapping may be achieved between these two taxonomies. Our assumption is that labelling a song as slow is indicative of the existence of prevalent acoustic characteristics that have led to a selection of the lower tempo octave, while a classification of fast is indicative of features that prompted a rate of synchronization within the faster tempo octave.

2.1 Data collection

To generate our datasets, we created a data harvester¹ built on the Last.fm and YouTube APIs. Our initial intention was to extract features and train our classifiers based on audio for songs that were relevant to the fast and slow tags on Last.fm. Because audio content is for the most part not available on Last.fm, we opted instead to generate a list of artist and track names associated with either fast or slow

¹ available at: <http://www.music.mcgill.ca/~hockman/other/mashup>

tags, and use each artist-track combination in this list as search terms for videos on YouTube.

An initial list of artist and track names was created by mining Last.fm for the most popular tracks related to the query tags. Additional tracks were then appended to this list through a search for similar tracks that also displayed these tags. If the video matching the query was available, an audio track was automatically extracted from the video. Each file was then manually verified to be a version of the artist-track combination. The specific size and makeup of the dataset varied with the experiment being performed (as explained in Section 3).

2.2 Feature extraction

The success of our classification relies chiefly on our feature set, which has been generated using jAudio [12], a Java-based feature extraction tool from the jMIR software package [13].²

Each of the tempo estimation methods discussed in Section 1.1 generates an *onset detection function* (also known as a *driving signal*) by analyzing either a single feature or relatively few features, and tracks these over the course of overlapping windows; the aim being to highlight significant local signal characteristics, such as fast attack transients, while attenuating steady-state components.

Alternatively, our approach uses a significantly larger feature set, and characterizes features across entire tracks. We suspect that the perception of acoustic cues differs for songs heard as fast and slow, and that these cues are related to pitch, loudness, and timbre. We therefore extract a large number of features in hopes of exploiting regularities within these three musical attributes. Each audio track is first converted into a normalized 8 kHz single-channel .wav file. For each audio file, we assess over 80 overall features, including spectral centroid, rolloff, flux, variability, peak-based spectral smoothness, zero crossings, MFCCs, LPC, and Method of Moments, along with the aggregates [14] of several of these features, e.g., derivative, running mean, and standard deviation.

2.3 Classification

Classification is performed using jMIR’s Autonomous Classification Engine (ACE) software [15]. Provided feature vectors as created in Section 2.2 and a classifications file containing a list of labels directly from user data corresponding to each audio track as in Section 2.1, ACE performs classification with a variety of machine learning classification algorithms. Our experiments focused on the following six classifiers available in ACE:

- Unweighted k -Nearest Neighbor, with $k=1$ (k -NN)
- Support Vector Machines (SVM)
- Naive Bayes
- C4.5 Decision Trees (C4.5)
- AdaBoost seeded with C4.5 (AdaBoost)
- Bagging seeded with C4.5 (Bagging)

² available at: <http://jmir.sourceforge.net>

3. EXPERIMENTS

The goal for our experiments was to measure how well the above machine learning algorithms can identify fast and slow songs. To evaluate our method, we compared the output of several classifiers tested on two separate datasets. In all, we conducted three experiments: the first two deal specifically with identifying the best classification algorithm for determining fast or slow tempo, and the third compares our method against an existing tempo-tracking algorithm modified to output fast or slow values.

3.1 Experiment 1: Fast vs. slow

For the first of these experiments, we tested the feasibility of our approach using a dataset comprised of audio that users of Last.fm have tagged as fast or slow. The dataset was constructed as explained in Section 2.1, using search terms restricted to *fast* and *slow*. The total size of this dataset was 397 full-length audio tracks, comprised of 109 fast songs and 288 slow songs. Features were extracted as described in Section 2.2. Success rates are based on averages of five runs of three-fold cross-validation performed on the dataset with each classifier. Overall averages are displayed in Table 1.

Classifier	Avg. Success
k -NN ($k=1$)	97.48
SVM	99.37
Naive Bayes	98.24
C4.5	99.18
AdaBoost w/ C4.5	99.44
Bagging w/ C4.5	99.12

Table 1. 3-fold cross-validation results for Experiment 1. Values are presented in percentages for k -NN, SVM, Naive Bayes, C4.5, AdaBoost, and Bagging classifiers.

The best performing classifier was AdaBoost, closely followed by SVMs, C4.5, and Bagging. From the high success rates of these learners, we may infer the effectiveness of training exclusively with global features, as well as the lack of need for a periodicity function.

We can identify two weaknesses in our approach for this experiment, both related to genre. First, we did not attempt to control the influence of genre across tempo classes; it is plausible that relatively few genres comprise a large portion of the dataset, ultimately simplifying the classification task to one of basic genre classification (e.g., ambient vs. punk). Without genre labels we cannot reliably isolate the effect of genre from the determination of fast or slow music within our dataset.

Second, the fast and slow tags may have been made with respect to genre, and we cannot assume the motivation behind the use of these tags. While one listener might use these tags to describe the pace of a piece in relation to other music of many genres, others might use the same tags to describe its pace in relation to a specific genre. This could potentially be an issue if the two tag meanings were not

consistent. For example, a slower Drum and Bass track could conceivably be tagged as slow within the genre, or fast in comparison with other genres.

3.2 Experiment 2: Intra-genre fast vs. slow

Following the results of our previous experiment in Section 3.1, we designed an experiment to ensure that the classifiers were not simply classifying genres. For this experiment, a new dataset was created. An ideal dataset would have comprised of fast and slow versions of each song, eliminating any differences caused by genre that were not related to tempo. As we neither have such music, nor tags to describe it as fast or slow, we instead used our data harvester to find fast and slow music within each genre. For search tags we first looked for tempo-genre pairs in the form of fast x and slow x , where x is a genre taken from a list of over 1500 genres.³ For a tempo-genre tag pair to be considered as search terms, each tag was required to return a tracklist result with no less than five audio tracks for each genre. Once the list of tracks was established, they were downloaded as in the first experiment.

For this particular search, we found the distribution of tracklist results between fast and slow genres highly unbalanced. Many of the returned tempo-genre pairs (fast x and slow x) had a large number of files in one category and close to the minimum in the other. We therefore selected the five most evenly distributed genres (Country, Jazz, Rap, R&B, and Rock). Our desired dataset was comprised of at least thirty tracks in each tempo-genre class. As the number of tracks retrieved in each category did not meet our expectations, we decided to increase the size of the dataset by mining YouTube directly using the tempo-genre terms as queries for playlists. Our final dataset for this experiment was comprised of 831 verified full-length audio tracks, as shown in Table 2, and the complete list of the songs is available online.⁴

	Country	Jazz	Rap	RnB	Rock	Totals
Fast	33	112	63	76	111	395
Slow	66	103	78	120	69	436
Totals	99	215	141	196	180	831

Table 2. Dataset 2 breakdown by genre and tempo class.

We then tested our classification method within each of the five genres using three-fold cross-validation, as in the previous experiment. Results in Table 3 demonstrate the capability of each of the five classifiers in this task. Even the worst performer, the naive Bayesian classifier, scored above 93%. The top performers for each of the genres were either C4.5 or AdaBoost seeded with C4.5. The best classifier across all genres was again AdaBoost seeded with C4.5, and the most difficult genre tested across each classifier was Rap.

Next, as in Section 3.1 we evaluated each classifier’s ability to determine fast or slow across the entire dataset,

³ http://en.wikipedia.org/wiki/List_of_music_genres

⁴ <http://www.music.mcgill.ca/~hockman/projects/fastSlow/dataset.zip>

Genre	k -NN	SVM	Naive	C4.5	Ada	Bag
Cntry	94.83	97.26	92.51	98.48	97.95	97.46
Jazz	95.81	98.49	92.78	98.01	99.30	99.07
Rap	90.28	96.98	93.10	98.24	99.29	99.11
R&B	89.04	95.16	93.98	98.47	98.21	98.08
Rock	92.92	95.71	93.32	99.17	99.28	97.93
Avg.	92.58	96.72	93.14	98.47	98.80	98.33

Table 3. 3-fold cross-validation results for intra-genre tests in Experiment 2. Values are presented in percentages for k -NN, SVM, Naive Bayes (Naive), C4.5, AdaBoost (Ada), and Bagging (Bag) for each genre: Country (Cntry), Jazz, Rap, R&B, and Rock.

without genre separation. Results for this test are presented in Table 4. The top performing classifier was AdaBoost, and success rates were only minimally affected by the absence of genre specification. We can therefore conclude that the classifiers were able to learn fast and slow characteristics of music without prior knowledge of musical genre.

Classifier	Avg. Success
k -NN ($k=1$)	95.97
SVM	96.42
Naive Bayes	90.94
C4.5	95.10
AdaBoost w/ C4.5	96.81
Bagging w/ C4.5	96.45

Table 4. 3-fold cross-validation results (in percentages) for six classifiers tested across entire dataset (i.e., without genre separation) in Experiment 2.

3.3 Experiment 3: Applications in tempo-tracking

A third experiment was undertaken to compare the presented method to another method capable of fast and slow determination. This comparison was achieved using the results of the top performing classifier from Section 3.2 and the binarized output of a beat tracker [16] modified to provide a single tempo for each track in the second dataset. For each song n , the beat tracker calculates the derivative Δ of beats θ_n and outputs a single BPM value Γ_n as:

$$\Gamma_n = 60 / \text{median}(\Delta\theta_n). \quad (1)$$

An obstacle in the comparison between the two approaches is the selection of a boundary λ between fast and slow BPM values output by the tempo tracker. A plausible approach to scoring the output would be to identify a mean tempo for the dataset. However, as we lack ground truth BPM values for this dataset, we were unable to generate an average tempo at which to divide the tempo range. We therefore instead tested a set of integer tempo values $\{50, \dots, 150\}$ for λ , defining the optimal divisor as the tempo that provided the best results for the tempo tracker.

Table 5 shows the results of this experiment, with the best performing divisor between fast and slow, $\lambda = 93$ BPM.

Method	Success Rate
Classification (AdaBoost)	96.81
Tempo tracking, $\lambda = 93$ BPM	61.85

Table 5. Results for Experiment 3 (in percentages). Results for the classifier (AdaBoost) were generated using 3-fold cross-validation. Tempo tracker output was binarized using $\lambda = 93$ BPM as a tempo range divisor.

The discrepancy between results of the two approaches led us to attempt to improve the tempo tracker output using a genre-specific average tempo for each song in the dataset, as we felt that using fixed BPM value λ was unfairly scoring the tempo tracker. For these values, we used average genre tempi calculated from the BPM List⁵, a hand-annotated database of 20,000 BPM labels for popular Western music listed by genre. Unfortunately, decomposition by genre did not improve results.

The success rates for the tempo tracker in this experiment should *not* be taken to be indicative of the algorithm’s overall performance, as the intention of the tracker is not to define musical pace as either fast or slow, but rather to replicate the perceptual phenomenon of synchronization with a heard piece of music.

4. DISCUSSION

Through the three experiments performed in Section 3, classification of songs as either fast or slow has been shown to be a robust method of determining the overall pace of music. We have achieved above 96% accuracy for two separate datasets and demonstrated its effectiveness in this task over another existing methodology. The high success rate of the presented method suggests its reliability as an independent feature within several MIR tasks. In addition to using classification labels as features themselves, the method could also be used to improve lower-level metrical analysis such as tempo-tracking algorithms by selectively correcting misclassified tempo-tracking octave errors by simply using the classification results.

Our method differs considerably from existing approaches to the problem of tempo octave selection. First, we are currently using only two classes of possible output, as opposed to discretized BPM values. To achieve these class labels, we use machine learning algorithms trained on global features, calculated by aggregating windowed features for each training instance. In addition, we are using a large number of such features to describe each audio track in our dataset. A key difference that sets our method apart from all existing methods is that no periodicity calculation is attempted; we instead rely only on global features and statistics.

The two datasets used in the course of this study were created through the use of Last.fm and YouTube APIs, and were specifically created based on listener responses to audio. The composition of generated datasets is essential to the training of our classifiers, as the contents will define the ability of our classifiers to differentiate between the two classes. In review of our first experiment, we were concerned that our classification results were artificially high because our first dataset was constructed by downloading tracks associated with fast and slow tags, and that tracks associated with these tags were possibly leading to a division based on musical genre. We therefore constructed a second dataset for the following experiment, which contained examples of fast and slow music within each genre, reducing the effect of musical genre separation. Results of this experiment demonstrated that the classification approach could not only separate fast and slow music within each genre, but within the entire dataset as well.

A weakness of this approach lies in the ambiguity of responses to particular pieces of music. For example, songs in certain genres, such as Hip Hop, intentionally juxtapose a fast lyrical layer with slower percussion and bass loops (e.g., Bone Thugs’n’Harmony, Twista). In these scenarios, a number of listeners tagged some of these songs as fast, possibly referring to the unusually fast rate of lyrics, while other listeners tagged tracks in the same style as slow, possibly focusing on those characteristics that define the genre standards—namely the percussion and bass lines.

A second issue is the variable number of annotations per training file. On Last.fm, more popular songs are likely to have more instances of listeners using fast or slow tags, and thus improving tag reliability. In the present study, we have combined user data from Last.fm with playlist results from YouTube without regard to the number of listeners agreeing with each tag. While this did not cause difficulty for our experiments, perhaps an optimal method might be to directly label more music with Last.fm tags or even to perform structured listening tests.

5. CONCLUSIONS

We believe estimation of tempo octaves within music to be a perceptual phenomenon that can be learned through use of the presented classification model. In this paper we have outlined the training of such a model using a large number of global features related to the overall pitch, timbre, and loudness of an audio track. Through the use of the proposed fast or slow classification, we believe that it is possible to improve the usefulness of tempo-tracking models within applications requiring a reliable single tempo value.

In our future work, we would like to perform further evaluation of our method with several datasets of varied content. Specifically, we would like to test our method using an artificial dataset containing fast and slow versions of songs with the exact same spectral content. Such a dataset could be created through the use of any commercial

⁵ <http://www.bpmlist.com/>

sequencer using MIDI files to control synthesizer and sampler output. Evaluation on significantly larger datasets would also be of interest. A difficulty here might lie in the collection of ground truth for training. Towards this end, listening tests may be useful as an alternative source.

We also plan to investigate the applicability of the proposed method in the task of beat tracking. An obstacle in this area is that the proposed method defines *entire* songs. As we cannot assume that segments of the audio contain acoustic features that motivated the class labels (i.e., fast or slow) of the entire file, each segment would need to be classified independently, which would require manually labeled segments for training. Informal tests, however, suggest only a slight decrease in performance with audio segments of shorter durations, e.g., 10 seconds.

Finally, we intend to explore alternative strategies for incorporating our approach into tempo- and beat-tracking methods towards improved performance of these algorithms.

6. ACKNOWLEDGEMENT

The authors would like to thank the Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT) for their generous support, M. Davies for providing source code for his algorithm, and C. McKay, A. Hankinson, and J. Thompson for their technical assistance.

7. REFERENCES

- [1] D. Ellis: “Beat tracking with dynamic programming,” http://www.music-ir.org/mirex/2006/mirex/abstracts/2006/TE_BT_ellis.pdf (accessed March 1, 2009), 2006.
- [2] A. Klapuri, A. Eronen, and J. Astola: “Analysis of the meter of acoustic musical signals,” *IEEE Transactions on Speech and Audio Processing*, Vol. 14, No. 1, pp. 342–355, 2006.
- [3] M. Davies and M. Plumbley: “Context-dependent beat tracking of musical audio,” *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 15, No. 3, pp. 1009–1020, 2007.
- [4] S. Dixon and W. Goebel: “Pinpointing the beat: Tapping to expressive performances,” *Proceedings of the 7th International Conference on Music Perception and Cognition*, pp. 617–620, 2002.
- [5] P. Fraisse: “Rhythm and tempo,” in *The Psychology of Music*, ed. D. Deutsch, Academic Press, Orlando, Florida, pp. 649–680, 1982.
- [6] R. Parncutt: “A perceptual model of pulse salience and metrical accent in musical rhythms,” *Music Perception*, Vol. 11, No. 4, pp. 409–464, 1994.
- [7] J. Snyder and C. Krumhansl: “Tapping to ragtime: Cues to pulse finding,” *Music Perception*, Vol. 18, No. 4, pp. 455–489, 2001.
- [8] P. Toiviainen and J. Snyder: “Tapping to Bach: Resonance-based modeling of pulse,” *Music Perception*, Vol. 21, No. 1, pp. 43–80, 2003.
- [9] M. McKinney, D. Moelants, M. Davies, and A. Klapuri: “Evaluation of Audio Beat Tracking and Music Tempo Algorithms,” *Journal of New Music Research*, Vol. 36, No. 1, pp. 1–16, 2007.
- [10] M. Goto and Y. Muraoka: “A real-time beat tracking system for audio signals,” *Proceedings of the 1995 International Computer Music Conference*, pp. 171–174, 1995.
- [11] L. Xiao, A. Tian, W. Li, and J. Zhou: “Using a statistical model to capture the association between timbre and perceived tempo,” *Proceedings of the 9th International Conference on Music Information Retrieval*, pp. 659–669, 2006.
- [12] D. McEnnis, C. McKay, I. Fujinaga, and P. Depalle: “jAudio: A feature extraction library,” *Proceedings of the 6th International Conference on Music Information Retrieval*, pp. 600–603, 2005.
- [13] C. McKay and I. Fujinaga: “jMIR: Tools for automatic music classification,” *Proceedings of the 6th International Society for Music Information Retrieval Conference*, pp. 65–68, 2009.
- [14] D. McEnnis, C. McKay, and I. Fujinaga: “jAudio: Additions and improvements,” *Proceedings of the 7th International Conference on Music Information Retrieval*, pp. 385–386, 2009.
- [15] C. McKay, R. Fiebrink, D. McEnnis, B. Li, and I. Fujinaga: “ACE: A framework for optimizing music classification,” *Proceedings of the 6th International Conference on Music Information Retrieval*, pp. 42–49, 2005.
- [16] M. Davies, N. Degara, and M. Plumbley: “Evaluation methods for musical audio beat tracking algorithms,” Technical Report C4DM-TR-09-06, Queen Mary University of London, Centre for Digital Music, 2009.