COMPARATIVE EVALUATION AND COMBINATION OF AUDIO TEMPO ESTIMATION APPROACHES

JOSE R. ZAPATA AND EMILIA GÓMEZ

Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
{joser.zapata,emilia.gomez}@upf.edu

The automatic analysis of musical tempo from audio is still an open research task in the Music Information Retrieval (MIR) community. The goal of this paper is to provide an updated comparative evaluation of different methods for audio tempo estimation. We overview, following the same block diagram, 23 documented methods. We then analyze their accuracy, error distribution and statistical differences, and we discuss which strategies can provide better performance for different input material. We then take advantage of their complementarity to improve the results by combining different methods, and we finally analyze the limitations of current approaches and give some ideas for future work on the task.

INTRODUCTION

Tempo is a relevant semantic descriptor of a piece of music which represents the speed of the piece under study. For that reason, much research within the Music Information Retrieval (MIR) community has been devoted to finding ways to automate its extraction and many methods have been proposed. The task of tempo estimation is closely related to the detection of pulse, or beat defined as "one of a series of regularly recurring, precisely equivalent stimuli" [1].

Beat tracking and tempo estimation have been used in different application contexts, such as music retrieval, cover detection, playlist generation, and synchronization for audio mixing, structural analysis and score alignment.

Many approaches for tempo estimation have been proposed in the literature, and some efforts have been devoted to their quantitative comparison. The first public evaluation of tempo extraction methods was carried out in 2004. It considered 11 methods participating at the ISMIR audio description contest [2]. Later, the MIREX (Music Information Retrieval Evaluation eXchange) initiative <http://www.music-ir.org/> continued the evaluation of tempo extraction methods in 2005, 2006 and 2010. This evaluation was carried out with ground-truth consisting of three values, a slower tempo T1, a faster tempo T2 and the relative strength of T1 relative to T2 (0-1).The evaluation metrics were overall tempo p-score, percentage of agreement at least with one tempo and percent of agreement with two tempi are correct. In order to avoid the training of methods to the specific MIREX dataset, the audio files are not available to participants, so it is sometimes difficult to analyze limitations in current systems.

The goal of this paper is to provide an updated comparative evaluation of existing methods for audio tempo estimation. In order to do this, we have accessed 23 different approaches, and analyzed their differences in the different steps of the methods. We study their performance and error distribution, and discuss on the strategy that seems to get better results. We also propose to combine their outputs to improve their estimations and discuss on the limitations of current methods and ideas for future work.

1 OVERALL SCHEME

The general scheme of tempo induction methods is presented in Figure 1 and was proposed in [3]:

1. **Feature list creation block:** it transforms the audio waveform into a temporal series of features representing predominant rhythmic information.
2. **Pulse induction block**: it uses the parsed information to estimate periodicities in the signal.

Some methods also include the following steps:

3. **Beat-tracking block:** it provides the temporal positions of the beats.
4. **Back-end block:** it parses the beat positions to a global tempo estimation or selects the strongest tempo for some methods. In order to compare all the methods in the same conditions, this last block had to be implemented for some of them.

We have considered a total of 23 audio tempo induction methods, 11 of which were already evaluated in [2]. We had access to the others methods results through different infrastructures. A general description of each of the methods are presented. All the methods were tested using default configuration parameters.

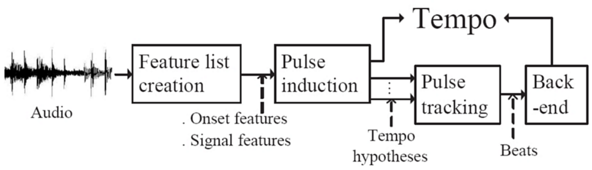

Figure 1: General tempo induction Blocks [2]

1.1 Aubiotempo

Aubio is an open source software released under the GNU/GPL license. The implementation for beat extraction is a Vamp plugin for Sonic Annotator was used for the test[1]. The method is described in detail in [4]. The output of this method are the beat positions, so a back end block is added to transform this information in to a single global tempo value: beat positions are converted into BPMs (Beats per minute), we then compute the IBI (inter beat interval), and the median value of all BPM's is taken as the algorithm output.

1.2 BeatIt

BeatIt from Music Technology Group, Universitat Pompeu Fabra, is a C++ implementation of a tempo and beat tracking method. The input signal is split into several frequency bands. For each band, the energy is computed, compressed, and differentiated. Next, the peak-to-peak distances between the maximum peaks of the autocorrelation function of each band are computed and stored. These are added to a histogram of one BPM octave. The maximum of the histogram (The tatum, fastest metrical level) sets the wrapped BPM estimation. Some statistics on the peak-to-peak distribution are used to select the output BPM octave.

1.3 Beatroot

BeatRoot from the Centre for Digital Music Queen Mary University of London, is a java implementation for beat tracking under the GNU Public License[2]. The output of this method are the beat positions, so a back end block is added to transform this information to a single global tempo value with the back end block. The method is described in detail in [5][6]. This is the same author of three other methods (DixonI, DixonT, DixonACF)[2] which took part in the experiment comparison of audio tempo induction methods realized for ISMIR 2004.

1.4 Ellis

Dan Ellis from Columbia University, proposed an approach for tempo induction implemented in Matlab[3]. The output of this method is given as a MIREX audio tempo estimation task (a slower tempo, T1, a faster tempo, T2, and the strength of T1 relative to T2). We then selected the BPM according to the value of the strength of T1 relative to T2 . The method is described in detail in [7].

1.5 Essentia

Eduard Aylon and Nicolas Wack from the Music Technology Group (UPF), proposed an approach for tempo estimation based on the Predominant Local Pulse curves (PLP) , by Grosche and Müller in [8]. The method is described in detail in [9]. This method is found in the Essentia library <http://mtg.upf.edu/technologies/essentia>.

1.6 Fixedtempo

Chris Cannam from Centre for Digital Music Queen Mary, University of London, wrote this “Simple Fixed Tempo Estimator” as a simplification of the method derived from work by Davies and Plumbley [10]. This method[4] is part of the vamp examples Plugins in Sonic annotator.

1.7 IBT

João Lobato Oliveira, Fabien Gouyon and Luis Gustavo Martins proposed a C++ implementation of a method integrated in the MARSYAS 0.4.0 framework <http://marsyas.info/>, and based on Beatroot [5] [6] under GPL. The method was tested in offline mode and gives a BPM estimation value as one of its outputs. The method is described in detail in [11].

1.8 jAudio

Cory McKay from McGill University, Canada, is the author of the tempo induction method called StrongestBeat of BeatHistogram and implemented in the system Jaudio 1.0.4. using the Java framework <http://jaudio.sourceforge.net/>. The output of the tempo estimation is a single tempo value[5].

1 http://www.omras2.org/SonicAnnotator
2 http://www.eecs.qmul.ac.uk/~simond/beatroot/

3 http://labrosa.ee.columbia.edu/projects/beattrack/tempo2.m
4 http://www.vamp-plugins.org/plugin-doc/vamp-example-plugins.html #fixedtempo
5 http://jaudio.sourceforge.net/jaudio10/javadoc/jAudioFeatureExtractor/AudioFeatures/BeatHistogram.html

1.9 MIRTempo

Olivier Lartillot from University of Jyväskylä, provides the Mirtempo[6] method at the Mirtoolbox platform. The method is described in detail in [12]. The MIR toolbox is available under the GNU General Public License.

1.10 MPEG7-xm

Jan Rohden from the Fraunhofer Institute for Digital Media Technology IDMT, wrote this method in the MPEG output document w5212 (15938-4:2001/FPDAM) in Matlab[7]. The source code and the method description are web available. The output is a BPM value each time the estimated tempo changes. For our evaluation, the estimated BPM of each song is computed as the median of all the provided estimates.

1.11 QM-Tempo

Matthew Davies and Christian Landone from the Centre for Digital Music Queen Mary, University of London, proposed an method for tempo and beat tracking called qm-tempotracker, which works as a Vamp plugin in Sonic Annotator[8]. This method is a hybrid of [13] and the dynamic programming by [7] . Its output is BPM values for each time the estimated tempo changes. For our test, estimated BPMs are combine into a single value using the median.

1.12 Tzanetakis

George Tzanetakis from Victoria University proposed a tempo induction method for MARSYAS, presented in the MIREX 2010 audio tempo estimation task, and is described in detail in [14]. This method provides MIREX output (a slower tempo, T1, a faster tempo, T2, and the strength of T1 relative to T2), so that the tempo used for this test is selected according to the value of the relative strength. Marsyas is distributed under the GNU Public Licence (GPL). George Tzanetakis is the author of three other methods (TzanetakisMS, TzanetakisMM and TzanetakisH)[2] which took part in the experiment comparison of audio tempo induction methods at ISMIR 2004.

1.13 Methods of ISMIR Audio Description Contest 2004

The evaluation presented here has been carried out using the same music collection from the ISMIR Audio Description Contest 2004 [2]. This allows us to compare the results of new methods with the ones that were evaluated then: Alonso (AlonsoACF, AlonsoSP) [15], Dixon (DixonACF [16], DixonI and DixonT [5]), Klapuri [17], Scheirer [18], Tzanetakis (TzanH, TzanMS and TzanMM [19]), Uhle [20]. The results of the all methods of this test are accessible at <http://mtg.upf.edu/ismir2004/contest/tempoContest/data3.tar.gz>.

A brief summary of the different steps of all tested methods is provided in Table 1.

2 EXPERIMENTAL FRAMEWORK

2.1 Infrastructure

All methods were run on a single computer with Windows OS (XP Professional Version 2002, SP 3 with an Intel core duo 2.666 GHz, 2 GB RAM). The evaluation framework was designed as a set of Matlab (version 7.5.0.342, Release 2007b) scripts.

2.2 Music Collection

The database was used for ISMIR 2004 tempo induction contest presented in [2]. It consists on songs with approximately constant tempi, and the format is the same for all: mono, linear PCM, 44100 Hz sampling frequency, 16 bits resolution. The total duration of the test set is approximately 9300 sec.

The database is composed of 465 song excerpts of 20 seconds, A professional musician placed beat marks on all song excerpts, and and the annotations were cross-checked by Fabien Gouyon. We also performed a cross-annotation of the data. The ground-truth tempo was computed as the median of the IBIs (Inter Beat Interval). Data and annotations are available in <http://mtg.upf.edu/ismir2004/contest/tempoContest/data3.tar.gz >

- Total number of instances: 465
- Duration: around 20 s
- Total duration: around 9300 s
- Genres: see distribution in Table 2
- Tempo range: between 24 and 242 BPM, (Figure 2)

3 EVALUATION STRATEGY

This evaluation was carried out using different metrics than the ones proposed by MIREX. First, the ground truth consisted of a single tempo value, and in order to compare with the previous evaluation, the test was done only with one BPM output for each method. Other metric were then added to determine if the tempo methods detects faster or slower level tempo metrics in a relation of 4,1/4, 6 with the ground-truth.

Three evaluation metrics were then used for the test:

6 https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

7 http://mpeg7.doc.gold.ac.uk/mirror/v2/Matlab-XM/AudioBpmD/AudioBpmD.m

8 http://www.vamp-plugins.org/plugin-doc/qm-vamp-plugins.html#qm-tempotracker

Algorithm	Aubio	BeatIt	Beatroot	Ellis	Essentia	Fixedtempo
Author	Brossier [4]	Bonada and Gouyon	Dixon [5] [6]	Ellis [7]	Aylon and Wack [8] [9]	Cannan [10]
Infrastructure	Sonic annotator	Windows Binary	Java	Matlab	Essentia	Sonic annotator
Output	Beats Positions in Time	One Bpm	Beats Positions in Time	(slower tempo, T1, a faster tempo, T2, and the strength of T1 relative to T2)	One Bpm	One Bpm
Feature list	Onset detection (Complex spectral difference)	Energy envelope differences for 8 bands	Onset detection (Spectral Flux)	Onsets obtained from the Mel spectrogram	Energy enveloped differences (Novelty curves) for 5 bands	Overall energy rise function.
Pulse induction	ACF	ACF	IOI Clustering	ACF	ACF	ACF
Algorithm	**IBT**	**jAudio**	**MIRTempo**	**Mpeg7-xm**	**Qmtempo**	**Tzanetakis**
Author	Oliveira et al. [11]	McEnnis and McKay	Lartillot [12]	Rohden	Davies and Plumbley [13]	Tzanetakis [14]
Infraestructure	Marsyas	Java	Matlab	Matlab	Sonic annotator	Marsyas
Output	One Bpm	One Bpm	One Bpm	BPM value each time the estimated tempo changes	BPM value each time the estimated tempo changes	(slower tempo, T1, a faster tempo, T2, and the strength of T1 relative to T2)
Feature list	Onset detection (Spectral Flux)	Energy envelope (256 window)	Onset curve, (10 channel gammatone filterbank and a low pass filter)	Energy envelopes for 6 bands	Onset detection (Spectral Flux)	Onset strength signal (based on Spectral Flux)
Pulse induction	ACF	ACF	ACF	ACF	ACF	ACF
Algorithm	**AlonsoACF**	**AlonsoSP**	**DixonACF**	**DixonI**	**DixonT**	**Klapuri**
Author	Alonso [15]	Alonso [15]	Dixon [16]	Dixon [5]	Dixon [5]	Klapuri [17]
Infrastructure	Matlab	Matlab	Matlab	Java	Java	Linux binary
Output	One Bpm	One Bpm	One Bpm	One Bpm	One Bpm	Beats Positions in Time
Feature list	Onsets of Notes	Onsets of Notes	Downsampled and smoothed Energy of 8 frequency bands	Energy based Onset Detector	Energy based Onset Detector	The differentials of the loudness in 36 frequency subbands
Pulse induction	ACF	Spectral Product	ACF	IOI Clustering	IOI Clustering	Bankcomb Filter
Algorithm	**Scheirer**	**Tzan_H**	**Tzan_MM**	**Tzan_MS**	**Uhle**	
Author	Scheirer [18]	Tzanetakis [19]	Tzanetakis [19]	Tzanetakis [19]	Uhle [20]	
Infrastructure	Linux binary	Linux binary	Linux binary	Linux binary	Windows Binary	
Output	Beats Positions in Time	One Bpm	One Bpm	One Bpm	One Bpm	
Feature list	Energy envelope differences for 6 bands computed by a filterbank	Energy Envelopes of 5 octave – space frequency bands obtained by Wavelets	Energy Envelopes of 5 octave – space frequency bands obtained by Wavelets	Energy Envelopes of 5 octave – space frequency bands obtained by Wavelets	The differential of a smoothed Energy envelopes for logarithmically space frequency bands	

Table 1: Brief Description of the methods, ACF (Autocorrelation), IOI (Inter Onset Interval)

- Metric 1: The percentage of the tempo estimation within 4% (the precision window) of the ground truth tempo. This procedure was used to evaluate the accuracy of the methods to detect the general Bpm of the song.
- Metric 2: The percentage of the tempo estimation within 4% (the precision window) of the 1, 2, 1/2, 3, 1/3 times the ground-truth tempo. This procedure is to take into account problems of double or triple deviation of the tempo estimation.

- Metric 3: The percentage of the tempo estimation within 4% (the precision window) of the of the 1, 2, 1/2, 3, 1/3, 4, 1/4, 6 times the ground-truth tempo. This procedure is to evaluate the errors made by the automatic estimation of the various levels of beats, present in a song.

Genre	# Songs
Rock	68
Classical	70
Electronic	59
Latin	44
Samba	42
Jazz	12
Afrobeat	3
Flamenco	13
Balkan and Greek	144
Fado	10

Table 2: Genre Distribution of the song excerpts

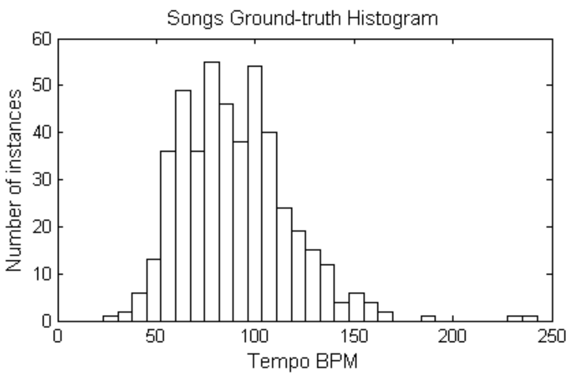

Figure 2: BPM ground- truth Histogram.

4 RESULTS[9]

Table 3 shows the overall ranking of methods according to the average estimation accuracy (metrics 1,2 and 3). We have then outlined some characteristics of the methods that we consider relevant: the strategy for pulse induction; those methods using multi-band processing, and whether if the band integration happens before or after the periodicity detection stage.
The results in percentage of the evaluation metrics 1, 2 and 3 are summarized in Table 4 and Figure 3.
As a general observation, for all the song excerpts, at least one method correctly estimates the tempo with a ratio of 1, 2, 1/2 or 3. With the results of all methods is possible to reach (**oracle result**) an accuracy of: {90,53%, 100%,100%} in the evaluation metrics 1, 2 and 3 respectively. We then assume that rhythmic periodicity can be accurately estimated from the raw audio signal.
We also found that the maximum number of methods estimations agreeing with the ground-truth is 21, and this occurs for 3 song-excerpts. This means that none of the songs are correctly labelled by all the approaches.

Algorithm	Pulse Induction	Combinig Bands
Klapuri	B	After
BeatIt	A	After
Ellis	A	
qmtempo	A	
IBT	A	
Essentia	A	After
Mpeg7-xm	A	After
Uhle	A	After
Scheirer	B	After
AlonsoSP	SP	
DixonACF	A	Before
Aubio	A	
MIRTempo	A	Before
Beatroot	IOI	
DixonT	IOI	
Tzanetakis	A	
DixonI	IOI	
AlonsoACF	A	
Tzan_MS	A	Before
Fixedtempo	A	
Tzan_H	A	Before
Tzan_MM	A	After
jAudio	A	

Table 3: Evaluation performance ranking of methods and information of Onset detection, Pulse induction (A= Autocorrelation, B: Bank-comb filter, SP: Spectral product) and Band Combination (after o before the pulse induction)

The best performance in the evaluation is obtained by Klapuri method, with the following accuracy measures: {58.49%, 91.18%, 91.18%}. The first metric was lower than the BeatIt method {60,43%}, but the results are statistically comparable between them. For evaluation metrics 2 and 3, Klapuri's method provides the best performance and the difference between its results are statistically significant compared to the rest of tempo estimators.

The method with lowest accuracy was Jaudio. It's important to note that the Jaudio method estimates the tempo with default parameters.

9 https://sites.google.com/site/tempoandbeattracking/

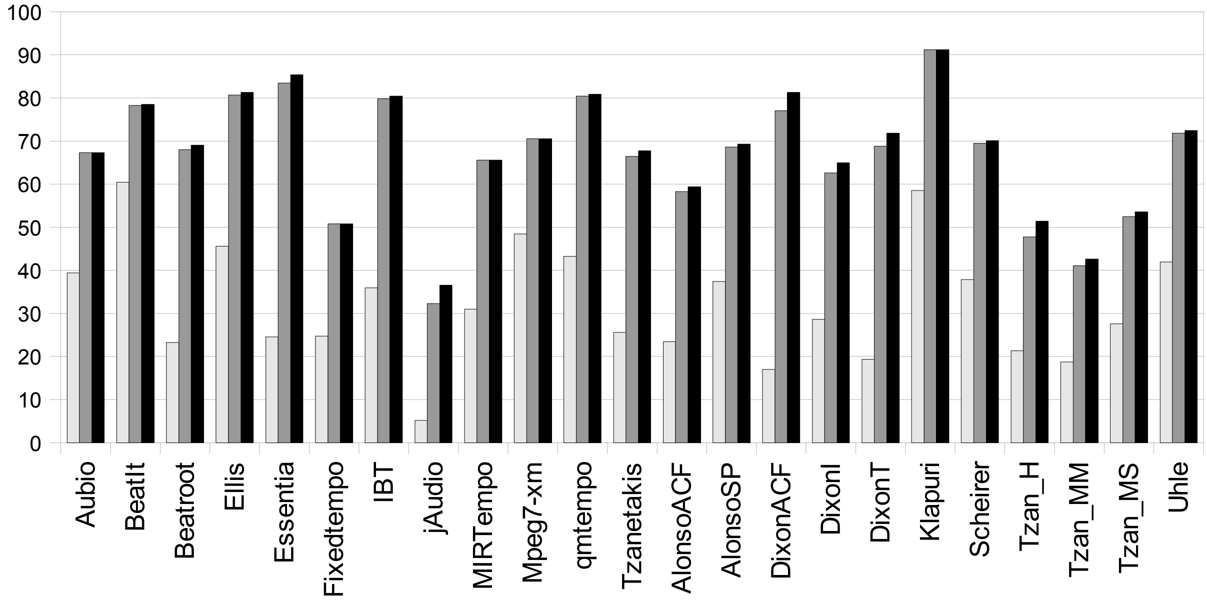

Figure 3 Evaluation Results, Light bar Metric 1, Grey bar Metric 2 and Black bar Metric 3

	Aubio	BeatIt	Beatroot	Ellis	Essentia	Fixedtempo	IBT	jAudio	MIRTempo	Mpeg7-xm	qmtempo	Tzanetakis
Evaluation 1	39,35%	60,43%	23,23%	45,59%	24,51%	24,73%	35,91%	5,16%	30,97%	48,39%	43,23%	25,59%
Evaluation 2	67,31%	78,28%	67,96%	80,65%	83,44%	50,75%	79,78%	32,26%	65,59%	70,54%	80,43%	66,45%
Evaluation 3	67,31%	78,49%	69,03%	81,29%	85,37%	50,75%	80,43%	36,56%	65,59%	70,54%	80,86%	67,74%

	AlonsoACF	AlonsoSP	DixonACF	DixonI	DixonT	Klapuri	Scheirer	Tzan_H	Tzan_MM	Tzan_MS	Uhle
Evaluation 1	23,44%	37,42%	16,99%	28,60%	19,35%	58,49%	37,85%	21,29%	18,71%	27,53%	41,94%
Evaluation 2	58,28%	68,60%	76,99%	62,58%	68,82%	91,18%	69,46%	47,74%	41,08%	52,47%	71,83%
Evaluation 3	59,35%	69,25%	81,29%	64,95%	71,83%	91,18%	70,11%	51,40%	42,58%	53,55%	72,47%

Table 4: Evaluations Results

The main difference between Klapuri's method and other methods lies in, first, the feature list creation block, which computes the subtle energy changes that might occur in narrow and wide frequency sub-bands: second, the induction block is a bank of comb filter resonators used by Scheirer. So we might assume than the accuracy of the method lies in a good feature list extraction and beat tracking, rather than a complex tempo induction block. For ISMIR 2004 Klapuri's method had a back end block that computes the tempo as the median of the IBI's estimated by the method [2], being the same method used to compute the ground truth of the music collection.

Comparing the 11 methods (Klapuri, Beatit, Mpeg7-xm, Uhle, essentia, Scheirer, DixonACF, MIRtempo, Tzan_ms, Tzan_h, Tzan_mm) which divide the signal in sub-bands in the feature list block and use autocorrelation in the pulse induction block, the six most accurate methods compute the autocorrelation function before the frequency integration. The statistical difference of the accuracy between the first three methods and the others methods, which compute autocorrelation after combining signals of each band, are significant.

The methods which use differential values of the energy signal (Klapuri, Beatit, Qmtempo, IBT, Uhle, Essentia and Scheirer) in the feature list creation block performed better than those that use frame values.

In the 13 song-excerpts with tempi below 49 bpm, none of the methods correctly estimates them, but more than 4 methods provided double or triple of the ground-truth annotation. This might reflect the fact that current methods are not adapted to slow tempi, but they can detect one metrical level of the song-excerpt.

For the 41 song-excerpts with only one correct estimation (ratio=1), the methods with most correct estimations were BeatIt = 11 and IBT = 5, and the double tempo was the most common error among the rest of the methods.

We also observe significant differences in the values of metric 1 and metric 2, as most of the methods detect double or triple tempo of the ground-truth. Comparing metric 2 and metric 3, at least 12 methods (Aubio, Beatit, Beatroot, Essentia, IBT, MIRtempo, Mpeg7 – xm, Qmtempo, Tzanetakis, AlonsoSP, DixonACF, Klapuri, Scheirer and Uhle) are above 65% accuracy of the ground-truth.

4.1 Statistical Significance

The statistical significance of the method estimations in the three evaluation metrics was carried out by means of the McNemar Test [21], considering a **p-value of 0.01** as the threshold for statistical significance.

The significance between methods for each evaluation is presented in Table 5, Table 6 and Table 7 for metric 1, metric 2 and metric 3 respectively. The statistical significance is provided by a symmetrical matrix. From the statistical comparison between methods, a filled cell represents equal statistical performance between the methods.

The method ids used in the statistical significance tables are: Essentia= 1, Fixedtempo = 2, qmtempo = 3, Aubio = 4, Beatroot = 5, Ellis = 6, Tzanetakis = 7, MIRTempo = 8, IBT = 9, BeatIt = 10, Mpeg7-xm = 11, jAudio = 12, AlonsoACF = 13, AlonsoSP = 14, DixonACF = 15, DixonI = 16, DixonT = 17, Klapluri= 18, Scheirer= 19, Tzan_H = 20, Tzan_MM= 21, Tzan_MS = 22, Uhle=23.

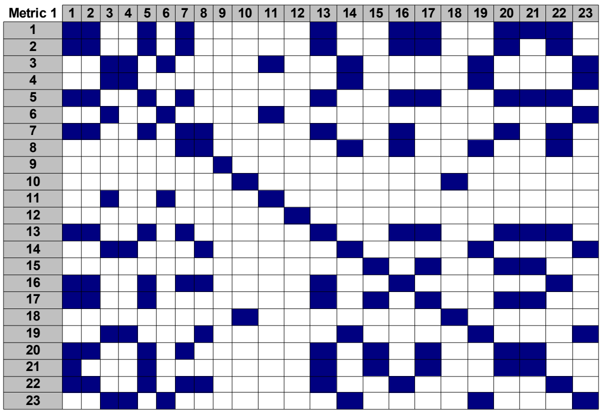

Table 5: Statistical significance between methods In Metric 1

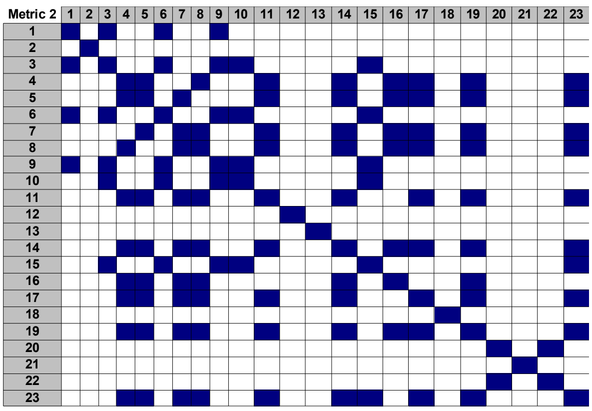

Table 6: Statistical significance between methods In Metric 2

The diagonal in the matrices are present because each method is compared with itself.

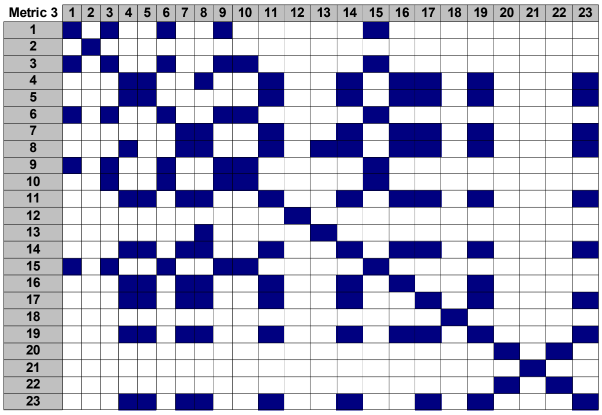

Table 7: Statistical significance between methods In Metric 3

Comparing the statistical significance presented in Table 5, Table 6 and Table 7, and the ranking of the Table 3 most of the best tempo estimations in the evaluation can be obtained by restricting to the seven most accurate methods (Klapuri, Beatit, Ellis, Qmtempo, IBT, Essentia and Mpeg7-xm).

The better performance in metric 1 obtained by BeatIt method is statistically similar with the Klapuri method, but different with metric 2 and 3. the difference between the metric 1, 2 and 3 of the BeatIt method is because its lower tempo estimation tendency made by its octave correction.

In the results of the three metrics some methods were found to be related to each other: Essentia and DixonACF; Tzanetakis and MIRtempo; AlonsoSP with Aubio, Scheirer and Uhle. The statistical similarity between DixonI, DixonT and Beatroot in metrics 2 and 3 shows that the difference in the feature extraction block and variations of IOI clustering for pulse induction block does not represent statistical differences in the obtained results.

4.2 Error Analysis

Compared with the ground truth, most of the methods tend to estimate the double tempo, but other error tendencies such as 1/2, 3, 4, 4/3, 2/3 were present in the results of all methods in the whole database. The test was done without any knowledge of the meter of the songs, but a ternary tendency of some songs can be detected from the relation between the method estimations. The error ratio tendencies of all methods can be seen in Table 8.

We first observe that seven methods (FixedTempo, Tzanetakis, MIRtempo, Aubio, AlonsoACF, Tzan_mm) had an error with a ratio = 4/3 above 8% with a value of {12.9%, 11.40%, 10.75%, 10.54%, 8.6%, 8.39%} respectively. This represents an error of 3/4 in the Inter Beat Intervals. This error is more common than the 1/2.

Then, the Jaudio method tends to estimate faster tempo than the rest, and has most of its estimations above a ratio of 2. As this method was tested with default parameters, we cannot conclude if it tends to estimate faster tempi than the other ones. On the other hand, BeatIt method tends to estimate slower tempo than the rest, but the tempo distribution of the database had more song-excerpts with a BPM in a range between 60 – 110, so a database with equal Bpm distribution would be needed to confirm this tendency.

Ratio	%
1/4	0,03%
1/3	0,09%
1/2	3,62%
2/3	1,89%
3/4	0,62%
Good	32,12%
5/4	1,02%
4/3	5,36%
3/2	1,88%
2	29,00%
9/4	1,65%
5/2	1,14%
8/3	1,37%
3	1,80%
4	1,25%
16/3	0,07%
6	0,15%
Other	16,94%

Table 8: Other tendencies results of the all methods

We also see that the sum in percentage of the fraction errors of all methods is 18,89%. If the meter of each excerpt were known, the tempo estimation could be then improved. The errors of 2/3, 4/3,8/3 and 3 show a problem in the pulse selection. Given that without knowing the meter it can be difficult to estimate a ground- truth, these estimations can be useful for meter detection.

Based on the differences of evaluation metric 1 and 2, the methods with more than a 40% tendency to detect double or triple the ground- truth are: DixonAFC, Essentia, DixonT, Beatroot, IBT and Tzanetakis, with a difference equal to {60%, 58.92%, 49.46%, 44.73%, 43.87%, 40.86%} respectively.

The 78.26% of the methods had a difference less than 5% between evaluation metric 2 and 3, and these is a problem in the pulse selection process, but shows that the methods can detect a periodicity related with the ground-truth in the audio signal.

The song-excerpts with more failures were *Bach - BWV 639 - Ich ruf' zu dir, Herr Jesu Christ.wav* (65,21 Bpm) and *Vivaldi - Stabat Mater 01.wav* (49 Bpm) with 16 other (non-comparable relation) estimations. For the two song-excerpts 1 approaches estimate the ground-truth and 3 methods estimate the double value. These methods don't have a clear relation between their methods to get a conclusion, but the evidence shows, at least, that the double or triple of the ground-truth can be estimated by automatic systems.

4.3 Combination Method

Based on the statistical significance using the top seven methods in the evaluation (Klapuri, Beatit, Ellis, Qmtempo, IBT, Essentia and Mpeg7-xm) the results of most of the methods are obtained. With these results we use a heuristic strategy to improve the results of the best method in the evaluation.
The hypotheses obtained from the experiment is that the ground-truth and the double tempo value are among the estimations for most of the song-excerpts.
The Klapuri estimations are checked with a metrical hierarchy analysis with the results of the other six methods, and using the low tempo tendency of Beatit and the double tendency of Essentia. In order to correct bad estimations and meter level issues. The steps are presented in pseudo code:

```
function [p, Out] = Relation(n)
{   array = [Beatit(n), Ellis(n), Qmtempo(n),
                     IBT(n), Essentia(n), Mpeg7-xm(n)];

    if  three or more values from array are equal and
    if exist the double of these values in the array
           p  = 1;
           Out = the repeat value in the array;
    else
           p = 0; Out = 0;
    end
}

% Comparisons are computed with  4% window
% 465 song-excerpts
Main() {
for n =1:465
   Result(n) = klapuri(n);
   if  (Result(n) ~= BeatIt(n) & Result(n) ~= Essentia(n))
           [p, Out] =Relation(n);
           if (p ==1)
                     Result(n) = Out;
           end
   end
% To avoid the double error
    if (Result(n) == BeatIt(n)*2 & BeatIt(n) = = Ellis(n))
         Result(n) = BeatIt(n);
    end
end
}
```

The accuracy results of this configuration are: {65.37%, 91.39%, 91.61% } for metric 1, 2 and 3 respectively in this database. This values are less than the oracle results but Klapuri estimations are improved in 6.67% for metric 1, 0.21% for metric 2 and 0.43% for metric 3. Calculating the statistical relation between Klapuri and the combination method the p-values are: {0.0029, 1, 0.7539} for the evaluation metrics 1, 2 and 3 respectively, considering a p-value of 0.01 as threshold for statistical significance, the results of the combination method against Klapuri are statistical different for metric 1.

Some of these tested methods had more than one output and the combination results are less than the oracle results, if all outputs of the methods are used, assuming these results as some other experts, the tempo estimation might be improved.

5 CONCLUSIONS

In this paper, we have evaluated 23 published methods for tempo estimation, that we consider representative of current approaches. The best estimation had an overall accuracy of 91% but can be improve with a heuristic combination of the other methods and the results are better than each approach by itself but an open question remains about how their results could be integrated to improve tempo estimation to get better performance.

For the 90,53% of the song-excerpts, the ground-truth and their metrical levels can be estimated with the results of all methods (the double error is the most common error, followed by the 4/3). In future work, research should be devoted to metrical level estimations, binary and ternary detection and slow tempo estimation.

According to the statistical significance analysis of the evaluation results, we conclude that among the tested methods involving band decomposition, those computing the periodicity detection before the multi-band integration achieve better results.

The best performing methods (Klapuri and BeatIt) are based on the following steps: differential energy values, frequency decomposition, periodicity detection prior to the multi-band integration, and tatum detection in the pulse-induction block. Klapuri's method is still the best performing one among all the evaluated methods.

As for future work, an evaluation with commercial approaches and other already tested methods in the same database can be done. The computational global tempo estimation is intrinsically linked to the beat position, so a similar comparison of state-of-the-art of beat-tracking systems would shed light on this problem.

6 ACKNOWLEDGEMENTS

Thanks to the authors of the methods for making their codes or executables available on the web for research topics, Fabien Gouyon and Giorgios Emmanouil for the annotated collection dataset with the results of the previous participants. Perfecto Herrea, Fabien Gouyon, Matthew Davies and the anonymous reviewers for their suggestions. Eduard Aylon , Nicolas Wack for the Essentia support, Thanks to Universidad Pontificia Bolivariana (Colombia), Colciencias, Music Technology Group at Universitat Pompeu Fabra , Classical Planet and DRIMS project for the financial support. and Robin Hofe for the Matlab code of the McNemar test <staffwww.dcs.shef.ac.uk/people/R.Hofe/mcnemar.html>

REFERENCES

[1] G. Cooper and L. Meyer, The rhythmic structure of music, University Of Chicago Press, 1960.

[2] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction methods," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, 2006, p. 1832–1844.

[3] F. Gouyon and S. Dixon, "A Review of Automatic Rhythm Description Systems," *Computer Music Journal*, vol. 29, 2005, pp. 34-54.

[4] M.E.P. Davies, P.M. Brossier, and M.D. Plumbley, "Beat tracking towards automatic musical accompaniment," *Proceedings of the Audio Engineering Society 118th convention, Barcelona, Spain*, 2005.

[5] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, 2001, p. 39–58.

[6] S. Dixon, "Evaluation of the Audio Beat Tracking System BeatRoot," *Journal of New Music Research*, vol. 36, 2007, p. 39–50.

[7] D. Ellis, "Beat Tracking by Dynamic Programming," *Journal of New Music Research*, vol. 36, 2007, pp. 51,60.

[8] P. Grosche and M. Müller, “A Mid-Level Representation For Capturing Dominant Tempo And Pulse Information In Music Recordings,” *10th International Society for Music Information Retrieval Conference (ISMIR)*, 2009, pp. 189-194.

[9] E. Aylon and N. Wack, “Beat detection using plp,” *MIREX 2010*, 2010, pp. 6-8.

[10] M.E.P. Davies and M.D. Plumbley, “Beat tracking with a two state model,” *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, 2005, pp. Iii/241-244.

[11] J.L. Oliveira, F. Gouyon, L.G. Martins, and L.P. Reis, “IBT: A Real-Time Tempo And Beat Tracking System,” *ismir2010.ismir.net*, 2010.

[12] O. Lartillot, “Mirtempo: tempo estimation through advanced frame-by-frame peaks tracking,” 2010.

[13] M.E.P. Davies and M.D. Plumbley, “Context-dependent beat tracking of musical audio,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, Mar. 2007, p. 1009–1020.

[14] G. Tzanetakis, “Marsyas Submissions To Mirex 2010,” *music-ir.org*, 2010.

[15] M. Alonso, B. David, and G. Richard, “Tempo and beat estimation of musical signals,” Proc. International Conference on Music Information Retrieval, Citeseer, 2004, p. 158–163.

[16] S. Dixon and E. Pampalk, “Classification of dance music by periodicity patterns,” Proc. International Conference on Music Information Retrieval, 2003, pp. 159 – 165.

[17] A.P. Klapuri, A.J. Eronen, and J.T. Astola, “Analysis of the meter of acoustic musical signals,” Audio, Speech, and Language Processing, IEEE Transactions on, vol. 14, 2006, pp. 342-355.

[18] E. Scheirer, “Tempo and beat analysis of acoustic musical signals,” *The Journal of the Acoustical Society of America*, vol. 103, 1998, pp. 588,601.

[19] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” IEEE Transactions on Audio, Speech and Language Processing, vol. 10, 2002, pp. 293-302.

[20] C. Uhle, R. Jan, C. Markus, and H. Jourgen, “Low Complexity Musical Meter Estimation from Polyphonic Music,” Audio Engineering Society Conference: 25th International Conference: Metadata for Audio, London, UK: 2004.

[21] L. Gillick and S. Cox, “Some statistical issues in the comparison of speech recognition methods,” Acoustics, Speech, and Signal Processing, 1989, p. 532–535.