JOINT BEAT AND DOWNBEAT TRACKING WITH RECURRENT NEURAL NETWORKS

Sebastian Böck, Florian Krebs, and Gerhard Widmer
Department of Computational Perception
Johannes Kepler University Linz, Austria
sebastian.boeck@jku.at

ABSTRACT

In this paper we present a novel method for jointly extracting beats and downbeats from audio signals. A recurrent neural network operating directly on magnitude spectrograms is used to model the metrical structure of the audio signals at multiple levels and provides an output feature that clearly distinguishes between beats and downbeats. A dynamic Bayesian network is then used to model bars of variable length and align the predicted beat and downbeat positions to the global best solution. We find that the proposed model achieves state-of-the-art performance on a wide range of different musical genres and styles.

1. INTRODUCTION

Music is generally organised in a hierarchical way. The lower levels of this hierarchy are defined by the *beats* and *downbeats* which define the *metrical structure* of a musical piece. While considerable amount of research focused on finding the *beats* in music, far less effort has been made to track the *downbeats*, although this information is crucial for a lot of higher level tasks such as structural segmentation and music analysis and applications like automated DJ mixing. In western music, the *downbeats* often coincide with chord changes or harmonic cues, whereas in non-western music the start of a *measure* is often defined by the boundaries of rhythmic patterns. Therefore, many algorithms exploit one or both of these features to track the *downbeats*.

Klapuri et al. [18] proposed a system which jointly analyses a musical piece at three time scales: the tatum, tactus, and measure level. The signal is split into multiple bands and then combined into four accent bands before being fed into a bank of resonating comb filters. Their temporal evolution and the relation of the different time scales are modelled with a probabilistic framework to report the final position of the downbeats.

The system of Davies and Plumbley [5] first tracks the beats and then calculates the Kullback-Leibler divergence between two consecutive band-limited beat synchronous spectral difference frames to detect the downbeats, exploiting the fact that lower frequency bands are perceptually more important.

Papadopoulos and Peeters [24] jointly track chords and downbeats by decoding a sequence of (pre-computed) beat synchronous chroma vectors with a hidden Markov model (HMM). Two time signatures are modelled. In a later paper, the same authors [25] jointly model beat phase and downbeats while the tempo is assumed to be given. Beat and downbeat times are decoded using a HMM from three input features: the correlation of the local energy with a beat-template, chroma vector variation, and the spectral balance between high and low frequency content.

The system proposed by Khadkevich et al. [17] uses impulsive and harmonic components of a reassigned spectrogram together with chroma variations as observation features for a HMM. The system is based on the assumption that downbeats mostly occur at location with harmonic changes.

Hockman et al. [14] present a method designed specifically for hardcore, jungle, and drum and bass music, that often employ breakbeats. The system exploits onset features and periodicity information from a beat tracking stage, as well as information from a regression model trained on the breakbeats specific to the musical genre.

Durand et al. [10] first estimates the time signature by examining the similarity of the frames at the beat level – with the beat positions given as input. The downbeats are then selected by a linear support vector machine (SVM) model using a bag of complementary features, comprising chord changes, harmonic balance, melodic accents and pattern changes. In consecutive works [8, 9] they lifted the requirement of the beat positions to be given and enhanced their system considerably by replacing the SVM feature selection stage by several deep neural networks which learn higher level representations from which the final downbeat positions are selected by means of Viterbi decoding.

Krebs et al. [20] jointly model bar position, tempo, and rhythmic patterns with a dynamic Bayesian network (DBN) and apply their system to a dataset of ballroom dance music. Based on their work, [16] developed a unified model for metrical analysis of Turkish, Carnatic, and Cretan music. Both models were later refined by using a more sophisticated state space [21].

© Sebastian Böck, Florian Krebs, and Gerhard Widmer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Sebastian Böck, Florian Krebs, and Gerhard Widmer. "JOINT BEAT AND DOWNBEAT TRACKING WITH RECURRENT NEURAL NETWORKS", 17th International Society for Music Information Retrieval Conference, 2016.

The same state space has also been successfully applied to the beat tracking system proposed by Böck et al. [2]. The system uses a recurrent neural network (RNN) similar to the one proposed in [3] to discriminate between beats an non-beats at a frame level. A DBN then models the tempo and the phase of the beat sequence.

In this paper, we extend the RNN-based beat tracking system in order to jointly track the whole metrical cycle, including beats and downbeats. The proposed model avoids hand-crafted features such as harmonic change detection [8–10, 17, 24], or rhythmic patterns [14, 16, 20], but rather learns the relevant features directly from the spectrogram. We believe that this is an important step towards systems without cultural bias, as postulated by the "Roadmap for Music Information Research" [26].

2. ALGORITHM DESCRIPTION

The proposed method consists of a recurrent neural network (RNN) similar to the ones proposed in [2, 3], and is trained to jointly detect the beats and downbeats of an audio signal in a supervised classification task. A dynamic Bayesian network is used as a post-processing step to determine the globally best sequence through the state-space by jointly inferring the meter, tempo, and phase of the (down-)beat sequence.

2.1 Signal Pre-Processing

The audio signal is split into overlapping frames and weighted with a Hann window of same length before being transferred to a time-frequency representation with the Short-time Fourier Transform (STFT). Two adjacent frames are located 10 ms apart, which corresponds to a rate of 100 fps (frames per second). We omit the phase portion of the complex spectrogram and use only the magnitudes for further processing. To enable the network to capture features which are precise both in time and frequency, we use three different magnitude spectrograms with STFT lengths of 1024, 2048, and 4096 samples (at a signal sample rate of 44.1 kHz). To reduce the dimensionality of the features, we limit the frequencies range to [30, 17000] Hz and process the spectrograms with logarithmically spaced filters. A filter with 12 bands per octave corresponds to semitone resolution, which is desirable if the harmonic content of the spectrogram should be captured. However, using the same number of bands per octave for all spectrograms would result in an input feature of undesirable size. We therefor use filters with 3, 6, and 12 bands per octave for the three spectrograms obtained with 1024, 2028, and 4096 samples, respectively, accounting for a total of 157 bands. To better match human perception of loudness, we scale the resulting frequency bands logarithmically. To aid the network during training, we add the first order differences of the spectrograms to our input features. Hence, the final input dimension of the neural network is 314. Figure 1a shows the part of the input features obtained with 12 bands per octave.

2.2 Neural Network Processing

As a network we chose a system similar to the one presented in [3], which is also the basis for the current state-of-the-art in beat tracking [2, 19].

2.2.1 *Network topology*

The network consists of three fully connected bidirectional recurrent layers with 25 Long Short-Term Memory (LSTM) units each. Figures 1b to 1d show the output activations of the forward (i.e. half of the bidirectional) hidden layers. A softmax classification layer with three units is used to model the *beat*, *downbeat*, and *non-beat* classes. A frame can only be classified as downbeat *or* beat but not both at the same time, enabling the following dynamic Bayesian network to infer the meter and downbeat positions more easily. The output of the neural network are three activation functions b_k, d_k, and no_k, which represents the probability of a frame k being a beat but no downbeat, downbeat or non-beat position. Figure 1e shows b_k and d_k for an audio example.

2.2.2 *Network training*

We train the network on the datasets described in Section 3.1 — except the ones marked with an asterisk (*) which are used for testing only — with 8-fold cross validation based on a random splits. We initialise the network weights and biases with a uniform random distribution with range [-0.1, 0.1] and train it with stochastic gradient decent minimising the cross entropy error with a learning rate of 10^{-5} and 0.9 momentum. We stop training if no improvement on the validation set can be observed for 20 epochs. We then reduce the learning rate by a factor of ten and retrain the previously best model with the same early stopping criterion.

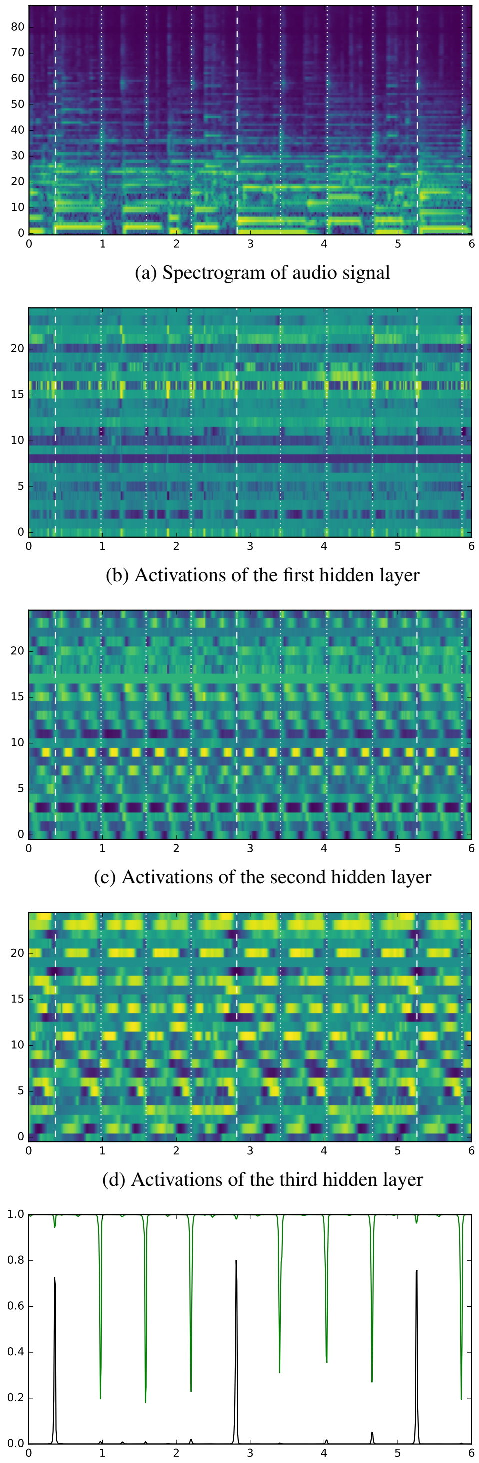
2.2.3 *Network output thresholding*

We experienced that the very low activations at the beginning and end of a musical excerpt can hurt the tracking performance of the system. This is often the case if a song starts with a (musically irrelevant) intro or has a long fade out at the end. We thus threshold the activations and use only the activations between the first and last time they exceed the threshold. We empirically found a threshold value $\theta = 0.05$ to perform well without harming pieces with overall low activations (e.g. choral works).

2.3 Dynamic Bayesian Network

We use the output of the neural network as observations of a *dynamic Bayesian network (DBN)* which jointly infers the meter, tempo, and phase of a (down-)beat sequence. The DBN is very good at dealing with ambiguous RNN observations and finds the global best state sequence given these observations. [1] We use the state-space proposed in [21] to model a whole bar with an arbitrary number of

[1] The average performance gain of the DBN compared to simple thresholding and peak-picking of the RNN activations is about 15% F-measure on the validation set.



(a) Spectrogram of audio signal

(b) Activations of the first hidden layer

(c) Activations of the second hidden layer

(d) Activations of the third hidden layer

(e) Activations of the softmax output layer

Figure 1: Signal propagation of a 6 second song excerpt in 4/4 time signature through the network: *(a)* part of the input features, *(b)* the first hidden layer shows activations at onset positions, *(c)* the second models mostly faster metrical levels (e.g. 1/8th notes at neuron 3), *(d)* the third layer models multiple metrical levels (e.g. neuron 8 firing at beat positions and neuron 16 around downbeat positions), *(e)* the softmax output layer finally models the relation of the different metrical levels resulting in clear downbeat (black) and beat (green, flipped for better visualisation) activations. Downbeat positions are marked with vertical dashed lines, beats as dotted lines.

beats per bar. We do not allow meter changes throughout a musical piece, thus we can model different meters with individual, independent state spaces. All parameters of the DBN are tuned to maximise the downbeat tracking performance on the validation set.

2.3.1 *State Space*

We divide the state space into discrete states s to make inference feasible. These states $s(\phi, \dot{\phi}, r)$ lie in a three-dimensional space indexed by the bar position state $\phi \in \{1..\Phi\}$, the tempo state $\dot{\phi} \in \{1..\dot{\Phi}\}$, and the time signature state r (e.g. $r \in \{3/4, 4/4\}$). States that fall on a *downbeat* position ($\phi = 1$) constitute the set of *downbeat* states $\mathcal{D}$, all states that fall on a *beat* position define the set of *beat* states $\mathcal{B}$. The number of bar-position states of a tempo $\dot{\phi}$ is proportional to its corresponding beat period $1/\dot{\phi}$, and the number of tempo states depends on the tempo ranges that the model accounts for. For generality, we assume equal tempo ranges for all time signatures in this paper but this could easily changed to adapt the model towards specific styles. In line with [21] we find that by distributing the tempo states logarithmically across the beat intervals, the size of the state space can be reduced efficiently without affecting the performance too much. Empirically we found that using $N = 60$ tempo states is a good compromise between computation time and performance.

2.3.2 *Transition Model*

Tempo transitions are only allowed at the beats and follow the same exponential distribution proposed in [21]. We investigated "peephole" transitions from the end of every beat back to the beginning of the bar, but found them to harm performance. Thus, we assume that there are no transitions between time signatures in this paper.

2.3.3 *Observation Model*

We adapted the observation model of the DBN from [2] to not only predict beats, but also downbeats. Since the activation functions (d, b) produced by the neural network are limited to the range $[0, 1]$ and show high values at beat/downbeat positions and low values at non-beat positions (cf. Figure 1e), the activations can be converted into state-conditional observation distributions $P(o_k|s_k)$ by

$$P(o_k|s_k) = \begin{cases} d_k & s_k \in \mathcal{D} \\ b_k & s_k \in \mathcal{B} \\ \frac{n_k}{\lambda_o - 1}, & otherwise \end{cases} \qquad (1)$$

where $\mathcal{D}$ and $\mathcal{B}$ are the sets of downbeat and beat states respectively, and the observation lambda $\lambda_o \in [\frac{\Phi}{\Phi-1}, \Phi]$ is a parameter that controls the proportion of the beat/downbeat interval which is considered as beat/downbeat and non-beat locations inside one beat/downbeat period. On our validation set we achieved the best results with the value $\lambda_o = 16$. We found it to be advantageous to use both b_k and d_k as provided by the neural network instead of splitting the probability of b_k among the N beat positions of the transition model.

2.3.4 *Initial State Distribution*

The initial state distribution can be used to incorporate any prior knowledge about the hidden states, such as meter and tempo distributions. In this paper, we use a uniform distribution over all states.

2.3.5 *Inference*

We are interested in the sequence of hidden states $\mathbf{s}_{1:K}$, that maximise the posterior probability of the hidden states given the observations (activations of the network). We obtain the maximum a-posteriori state sequence $\mathbf{s}^*_{1:K}$ by

$$\mathbf{s}^*_{1:K} = \arg\max_{\mathbf{s}_{1:K}} p(\mathbf{s}_{1:K}|o_{1:K}) \tag{2}$$

which can be computed efficiently using the well-known Viterbi algorithm.

2.3.6 *Beat and Downbeat Selection*

The sequence of beat $\mathbf{B}$ and downbeat times $\mathbf{D}$ are determined by the set of time frames k which were assigned to a beat or downbeat state:

$$\mathbf{B} = \{k : s^*_k \in \mathcal{B}\} \tag{3}$$

$$\mathbf{D} = \{k : s^*_k \in \mathcal{D}\} \tag{4}$$

After having decided on the sequences of beat and downbeat times we further refine them by looking for the highest beat/downbeat activation value inside a window of size Φ/λ_o, i.e. the beat/downbeat range of the whole beat/downbeat period of the observation model (Section 2.3.3).

3. EVALUATION

In line with almost all other publications on the topic of downbeat tracking, we report the F-measure (F_1) with a tolerance window of ±70 ms.

3.1 Datasets

For training and evaluation we use diverse datasets as shown in Table 1. Musical styles range from pop and rock music, over ballroom dances, modern electronic dance music, to classical and non-western music.

We do not report scores for all sets used for training, since comparisons with other works are often not possible due to different evaluation metrics and/or datasets. Results for all datasets, including additional metrics can be found online at the supplementary website `http://www.cp.jku.at/people/Boeck/ISMIR2016.html` which also includes an open source implementation of the algorithm.

Downbeat tracking dataset	# files	length
Ballroom [12,20] [2]	685	5 h 57 m
Beatles [4]	180	8 h 09 m
Hainsworth [13]	222	3 h 19 m
HJDB [14]	235	3 h 19 m
RWC Popular [11]	100	6 h 47 m
Robbie Williams [7]	65	4 h 31 m
Rock [6]	200	12 h 53 m
Carnatic [28]	176	16 h 38 m
Cretan [16]	42	2 h 20 m
Turkish [27]	93	1 h 33 m
GTZAN [23,29] *	999	8 h 20 m
Klapuri [18] [3] *	320	4 h 54 m
Beat tracking datasets		
SMC [15] *	217	2 h 25 m
Klapuri [18] [3] *	474	7 h 22 m

Table 1: Overview of the datasets used for training and evaluation of the algorithm. Sets marked with asterisks (*) are held-out datasets for testing only.

3.2 Results & Discussion

Table 2 to 4 list the results obtained by the proposed method compared to current and previous state-of-the-art algorithms on various datasets. We group the datasets into different tables for clarity, based on whether they are used for testing only, cover western, or non-western music. Since our system jointly tracks beats and downbeats, we compare with both downbeat and beat tracking algorithms.

First of all, we evaluate on completely unseen data. We use the recently published beat and downbeat annotations for the *GTZAN* dataset, the *Klapuri*, and the *SMC* set (built specifically to comprise hard-to-track musical pieces) for evaluation. Results are given in Table 2. Since these results are directly comparable (the only exception being the results of Durand et al. on the *Klapuri* set[4] and of Böck et al. on the *SMC* set[5]), we perform statistical significance tests on them. We use Wilcoxon's signed-rank test with a p-value of 0.01.

Additionally, we report the performance on other sets commonly used in the literature, comprising both western and non-western music. For western music, we give results on the *Ballroom*, *Beatles*, *Hainsworth*, and *RWC Popular* sets in Table 3. For non-western music we use the *Carnatic*, *Cretan*, and *Turkish* datasets and group the results in Table 4. Since these sets were also used during development and training of our system, we report results obtained with 8-fold cross validation. Please note that the results given in Table 3 and 4 are not directly comparable because they were either obtained via cross validation, leave-one-dataset-out evaluation, with overlapping train and test sets, or tested on unseen data. However, we still consider them

[2] We removed the 13 duplicates identified by Bob Sturm: http://media.aau.dk/null_space_pursuits/2014/01/ballroom-dataset.html

[3] The beat and downbeat annotations of this set were made independently, thus the positions do not necessarily match each other.

[4] 40 out of the 320 tracks were used for training.

[5] The complete set was used for training.

Test datasets	F_1 beat	F_1 downbeat
GTZAN		
new (bar lengths: 3, 4) *	0.856	0.640
Durand et al. [9] *	-	0.624
Böck et al. [2] *	0.864	-
Davies et al. [5] *	0.806	0.462
Klapuri et al. [18] *	0.706	0.309
Klapuri		
new (bar lengths: 3, 4) *	0.811	0.745
Durand et al. [9] ‡	-	0.689
Böck et al. [2] *	0.798	-
Davies et al. [5] *	0.698	0.528
Klapuri et al. [18] ‡	0.704	0.483
SMC		
new (bar lengths: 3, 4) *	0.516	
Böck et al. [2] §	0.529	
Davies et al. [5] *	0.337	
Klapuri et al. [18] *	0.352	

Table 2: Beat and downbeat tracking F-measure comparison with state-of-the-art algorithms on the test datasets. ‡ denotes overlapping train and test sets, § cross validation, and * testing only.

to be a good indicator for the overall performance and capabilities of the systems. For the music with non-western rhythms and meters (e.g. Carnatic art music contains 5/4 and 7/4 meters) we compare only with algorithms specialised on this type of music, since other systems typically fail completely on them.

Western music	F_1 beat	F_1 downbeat
Ballroom		
new (bar lengths: 3, 4) §	0.938	0.863
Durand et al. [9] †/‡	-	0.778 / 0.797
Krebs et al. [21] §	0.919	-
Böck et al. [2] §	0.910	-
Beatles		
new (bar lengths: 3, 4) §	0.918	0.832
Durand et al. [9] †/‡	-	0.815 / 0.842
Böck et al. [2] *	0.880	-
Hainsworth		
new (bar lengths: 3, 4) §	0.867	0.684
Durand et al. [9] †/‡	-	0.657 / 0.664
Böck et al. [2] §	0.843	-
Peeters et al. [25]	0.630	
RWC Popular		
new (bar lengths: 3, 4) §	0.943	0.861
Durand et al. [9] †/‡	-	0.860 / 0.879
Böck et al. [2] *	0.877	-
Peeters et al. [25]	0.840	0.800

Table 3: Beat and downbeat tracking F-measure comparison with state-of-the-art algorithms on western music datasets. † denotes leave-one-set-out evaluation, ‡ overlapping train and test sets, § cross validation, and * testing only.

Non-western music	F_1 beat	F_1 downbeat
Carnatic		
new (bar lengths: 3, 4) §	0.804	0.365
—— (bar lengths: 3, 5, 7, 8) §	0.792	0.593
Krebs et al. [21] §	0.805	0.472
Cretan		
new (bar lengths: 3, 4) §	0.982	0.605
—— (bar lengths: 2, 3, 4) §	0.981	0.818
—— (bar lengths: 2) §	0.980	0.909
Krebs et al. [21] §	0.912	0.774
Turkish		
new (bar lengths: 3, 4) §	0.740	0.495
—— (bar lengths: 4, 8, 9, 10) §	0.777	0.631
—— (tempo: 55..300 bpm) §	0.818	0.683
Krebs et al. [21] §	0.826	0.639

Table 4: Beat and downbeat tracking F-measure comparison with state-of-the-art algorithms on non-western music datasets. —— denotes the same system as the line above with altered parameters in parentheses, § cross validation.

3.2.1 Beat tracking

Compared to the current state-of-the-art [2], the new system performs on par or outperforms this dedicated beat tracking algorithm. It only falls a bit behind on the *GTZAN* and *SMC* sets. However, the results on the latter might be a bit biased, since [2] obtained their results with 8-fold cross validation. Although the new system performs better on the *Klapuri set*, the difference is not statistically significant. All results compared to those of other beat tracking algorithms on the test datasets in Table 2 are statistically significant.

Although the new algorithm and [2] have a very similar architecture and were trained on almost the same development sets (the new one plus those sets given in Table 1, except the *SMC* dataset), it is hard to conclude whether the new algorithm performs better sometimes because of the additional – more diverse – training material or due to the joint modelling of beats and downbeats. Future investigations with the same training sets should shed some light on this question, but it is safe to conclude that the joint training on beats and downbeats does not harm the beat tracking performance at all.

On non-western music the results are in the same range as the ones obtained by the method of Krebs et al. [21], an enhanced version of the algorithm proposed by Holzapfel et al. [16]. Our system shows almost perfect beat tracking results on the *Cretan* lap dances while performing a bit worse on the *Turkish* music.

3.2.2 Downbeat tracking

From Table 2 to 4, it can be seen that the proposed system not only does well for beat tracking, but also shows state-of-the-art performance in downbeat tracking. We outperform all other methods on all datasets – except *Beatles* and *RWC Popular* when comparing to the overfitted results obtained by the system of Durand et al. [9] – even the systems designed specifically for non-western music. We find this striking, since our new system is not designed specifically for a certain music style or genre. The results of our method w.r.t. the other systems on the test datasets in Table 2 are all statistically significant.

It should be noted however, that the dynamic Bayesian network must model the needed bar lengths for the respective music in order to achieve this performance. Especially when dealing with non-western music, this is crucial. However, we do not consider this a drawback, since the system is able to chose the correct bar length reliably by itself.

3.2.3 Meter selection

As mentioned above, for best performance the DBN must model measures with the correct number of beats per bar. Per default, our system works for 3/4 and 4/4 time signatures, but since the parameters of the DBN are not learnt, this can be changed during runtime in order to model any time signature and tempo range.

To investigate the system's ability to automatically decide on which bar length to select, we performed an experiment and limited the DBN to model only bars with lengths of three or four beats, both time signatures simultaneously (the default setting), or bar lengths of up to eight beats.

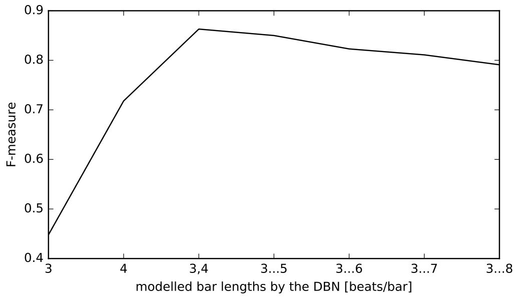


Figure 2: Downbeat tracking performance of the new system with different bar lengths on the *Ballroom* set.

Figure 2 shows this exemplarily for the *Ballroom* set, which comprises four times as many pieces in 4/4 as in 3/4 time signature. The performance is relatively low if the system is limited to model bars with only three or four beats per bar. When being able to model both time signatures present in the music, the system achieves it's maximum performance. The performance then slightly decreases if the DBN models bars with a length up to eight beats per bar, but remains on a relatively high performance level. This shows the system's ability to select the correct bar length automatically.

4. CONCLUSION

In this paper we presented a novel method for jointly tracking beats and downbeats with a recurrent neural network (RNN) in conjunction with a dynamic Bayesian network (DBN). The RNN is responsible for modelling the metrical structure of the musical piece at multiple interrelated levels and classifies each audio frame as being either a beat, downbeat, or no beat. The DBN then post-processes the probability functions of the RNN to align the beats and downbeats to the global best solution by jointly inferring the meter, tempo, and phase of the sequence. The system shows state-of-the-art beat and downbeat tracking performance on a wide range of different musical genres and styles. It does so by avoiding hand-crafted features such as harmonic changes, or rhythmic patterns, but rather learns the relevant features directly from audio. We believe that this is an important step towards systems without any cultural bias. We provide a reference implementation of the algorithm as part of the open-source *madmom* [1] framework.

Future work should address the limitation of the system of not being able to perform time signature changes within a musical piece. Due to the large state space needed this is intractable right now, but particle filters as used in [22] should be able to resolve this issue.

5. ACKNOWLEDGMENTS

This work is supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the GiantSteps project (grant agreement no. 610591) and the Austrian Science Fund (FWF) project Z159. We would like to thank all authors who shared their source code, datasets, annotations and detections or made them publicly available.

6. REFERENCES

[1] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. madmom: a new Python Audio and Music Signal Processing Library. `arXiv:1605.07008`, 2016.

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

[3] S. Böck and M. Schedl. Enhanced Beat Tracking with Context-Aware Neural Networks. In *Proc. of the 14th Int. Conference on Digital Audio Effects (DAFx)*, 2011.

[4] M. E. P. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical Report C4DM-TR-09-06, Centre for Digital Music, Queen Mary University of London, 2009.

[5] M. E. P. Davies and M. D. Plumbley. A spectral difference approach to downbeat extraction in musical audio. In *Proc. of the 14th European Signal Processing Conference (EUSIPCO)*, 2006.

[6] T. de Clercq and D. Temperley. A corpus analysis of rock harmony. *Popular Music*, 30(1):47–70, 2011.

[7] B. Di Giorgi, M. Zanoni, A. Sarti, and S. Tubaro. Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony. In *8th Int. Workshop on Multidimensional Systems (nDS)*, pages 145–150, 2013.

[8] S. Durand, J. P. Bello, B. David, and G. Richard. Downbeat tracking with multiple features and deep neural networks. In *Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[9] S. Durand, J. P. Bello, B. David, and G. Richard. Feature Adapted Convolutional Neural Networks for Downbeat Tracking. In *Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[10] S. Durand, B. David, and G. Richard. Enhancing downbeat detection when facing different music styles. In *Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[11] M. Goto, M. Hashiguchi, T. Nishimura, and R. Oka. RWC Music Database: Popular, Classical, and Jazz Music Databases. In *Proc. of the 3rd Int. Conference on Music Information Retrieval (ISMIR)*, 2002.

[12] F. Gouyon, A. P. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5), 2006.

[13] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Applied Signal Processing*, 15, 2004.

[14] J. Hockman, M. E. P. Davies, and I. Fujinaga. One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass. In *Proc. of the 13th Int. Society for Music Information Retrieval Conference (ISMIR)*, 2012.

[15] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9), 2012.

[16] A. Holzapfel, F. Krebs, and A. Srinivasamurthy. Tracking the "odd": meter inference in a culturally diverse music corpus. In *Proc. of the 15th Int. Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[17] M. Khadkevich, T. Fillon, G. Richard, and M. Omologo. A probabilistic approach to simultaneous extraction of beats and downbeats. In *Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[18] A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1), 2006.

[19] F. Korzeniowski, S. Böck, and G. Widmer. Probabilistic extraction of beat positions from a beat activation function. In *Proc. of the 15th Int. Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[20] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proc. of the 14th Int. Society for Music Information Retrieval Conference (ISMIR)*, 2013.

[21] F. Krebs, S. Böck, and G. Widmer. An Efficient State Space Model for Joint Tempo and Meter Tracking. In *Proc. of the 16th Int. Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[22] F. Krebs, A. Holzapfel, A. T. Cemgil, and G. Widmer. Inferring metrical structure in music using particle filters. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(5):817–827, 2015.

[23] U. Marchand and G. Peeters. Swing ratio estimation. In *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx)*, 2015.

[24] H. Papadopoulos and G. Peeters. Joint estimation of chords and downbeats from an audio signal. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1), 2011.

[25] G. Peeters and H. Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6), 2011.

[26] X. Serra et al. *Roadmap for Music Information ReSearch*. Creative Commons BY-NC-ND 3.0 license, ISBN: 978-2-9540351-1-6, 2013.

[27] A. Srinivasamurthy, A. Holzapfel, and X. Serra. In search of automatic rhythm analysis methods for turkish and indian art music. *Journal of New Music Research*, 43(1), 2014.

[28] A. Srinivasamurthy and X. Serra. A supervised approach to hierarchical metrical cycle tracking from audio music recordings. In *Proc. of the IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

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