Hazan A, Marxer R, Brossier P, Purwins H, Herrera P, Serra X. What/when causal expectation modelling applied to audio signals. Connection Science. 2009;21(2-3):119-43. DOI: 10.1080/09540090902733764

RESEARCH ARTICLE

What/when causal expectation modeling applied to audio signals

Amaury Hazan*, Ricard Marxer, Paul Brossier, Hendrik Purwins, Perfecto Herrera and Xavier Serra[a]

[a]*Universitat Pompeu Fabra, Ocata 1, 08001 Barcelona, Spain*

(*v2.0 released January 2008*)

A causal system to represent a stream of music into musical events, and to generate further expected events, is presented. Starting from an auditory front-end which extracts low-level (i.e. MFCC) and mid-level features such as onsets and beats, an unsupervised clustering process builds and maintains a set of symbols aimed at representing musical stream events using both timbre and time descriptions. The time events are represented using inter-onset intervals relative to the beats. These symbols are then processed by an expectation module using Predictive Partial Match, a multiscale technique based on N-grams. To characterize the ability of the system to generate an expectation that matches both ground truth and system transcription, we introduce several measures that take into account the uncertainty associated with the unsupervised encoding of the musical sequence. The system is evaluated using a subset of the ENST-drums database of annotated drum recordings. We compare three approaches to combine timing (*when*) and timbre (*what*) expectation. In our experiments, we show that the induced representation is useful for generating expectation patterns in a causal way.

1. Introduction

Since the last two decades there have been many attempts to build computational architectures of musical sequence learning (Bharucha and Todd 1989, Todd and Loy 1991, Mozer 1994, Lartillot et al. 2001, Tillmann et al. 2000, Assayag and Dubnov 2004, Pearce and Wiggins 2004, Pachet 2003). Because of the difficulty of finding a suitable computational representation of musical signals, many of these works have assumed beforehand a particular representation of the musical sequences to process, mainly in a symbolic form.

Our main focus lies in integrating a learning system which can constantly learn the structure of audio signals *while it listens to musical events*, in a way that is inspired by cognitive principles. We propose a causal and unsupervised system that learns the structure of an audio stream and predicts its continuation. Our system uses concepts and approaches from a variety of fields: automatic music transcription, unsupervised learning and model selection, and symbolic statistical learning. A mid-level representation of the signal is constructed as a discrete sequence of symbols representing time dependencies between events and timbre properties. A prediction of the subsequent symbols can then be provided, from which the system can then predict the nature and the timing of the next musical event. As such, the system prediction algorithm is symbolic, but the system is responsible of producing a symbolic representation of the musical stream in an unsupervised manner (i.e. the symbols are grounded on acoustic information).

From a general viewpoint, we assume a musical sequence as being a succession of musical events with intrinsic properties (e.g. pitch, loudness, timbre) which are heard at a given time. While the musical sequence is attended, timbre and timing patterns perceived so far (among other musical information that is not considered in this paper) can be used

*Corresponding author. Email: ahazan@iua.upf.edu

to provide a prediction of the next musical event to be heard. The prediction can be made over several dimensions, e.g. the timbre properties of the next event to be heard (*what*) or temporal location of the next event (*when*). Distinct strategies may be used by listeners to combine predictions along these dimensions. Even tough in the case of timbre recent evidence points to the possibility that timbre dimensions (e.g., attack slope, spectral centre of gravity, etc.) could be processed separately (Caclin et al. 2006) we will consider timbre as a nonseparable dimension to be used for predicting events. From the physiological literature, it seems to be a separation between the circuits dealing with the detection of violations of musical expectancies or predictions (in the ventrolateral prefrontal cortex), and the circuits dealing with the processing of perceptual objects and their monitoring in working memory (in the dorsolateral prefrontal cortex and posterior parietal cortex) (Sridharan et al. 2007).

In addition, we may assume a functional and physiological separation between sequential processing (related to musical syntax and grammar) which could take place in Broca's area (Maess et al. 2001) and the processing of timing information, which could be happening at the right temporal auditory cortex and superior temporal gyrus (Peretz and Zatorre 2005)). From these physiological considerations we could address the prediction of what and when dimensions as two independent processes which may be modeled with two independent predictors.

However, from the Auditory Scene Analysis (Bregman 1990) point of view, sound events may be separated into different auditory objects (auditory stream segregation) or assigned to a single auditory object (auditory stream fusion). This depends on the intrinsic properties of the musical events (e.g. timbre, pitch) and the events timing. If auditory stream integration takes place, each event may be described as how it differs from the preceding event in terms of timbre and timing. In this sense, the *what* and *when* dimensions might be merged into a unique dimension, which would be modeled with a unique next-event predictor. Finally, if auditory stream segregation takes place, a different representation of the events has to be considered. In a perceptual experiment (Creel et al. 2004), while the statistical regularities between nonadjacent tones were hardly learned when the musical sequence contained events which were similar in pitch or timbre, these regularities could be acquired when the temporally nonadjacent events differed in pitch range or timbre. Therefore, if a musical sequence can be perceptually segregated into separate timbre streams, the temporal dependencies between events might be computed (to a certain extent) from the separate timbre stream rather than from the temporally adjacent elements. This could be modeled by considering each segregated stream as a separate dimension which would be modeled with a stream-specific timing predictor. The system we present in this article enables us to implement these three strategies and compare them empirically using a database of annotated sequences of percussive events.

The rest of the paper is structured as follows. Section 2 introduces work related to our approach. We present, in Section 3, the details of our system, illustrating the different steps involved in the encoding and expectation of both time and timbre dimensions. In Section 4 we present a set of unsupervised and supervised performance statistics and present an experiment using a database of annotated drum excerpts. We also suggest that the expectation entropy can be used to detect regularities in the processed audio stream. We then discuss the results and present work directions in Section 6, and finally present our conclusions in Section 7.

2. Related work

Because we aim at building an integrated system that is able to learn the mid-level structure of the incoming audio stream, the work we present in this section is related to distinct fields. We first review approaches to model sequence learning applied to musical signals,

with special emphasis in research investigating the relations between timbre and timing in music perception and learning. Then, we review more applied works from the field of music information retrieval aiming at providing a mid-level description of timing and timbre in musical audio. Finally, as we are interested in investigating the learning dynamics of a musical system while it listens to a musical stream, we present works which provide an information-theoretic treatment of expectation in music.

2.1. *Models of sequence learning and musical expectation*

Sequence learning and structure acquisition have been modeled in a range of computational architectures, mostly using a symbolic encoding. One way to investigate sequence learning models lies in using connectionist architectures (Rumelhart and McLelland 1986). Early connectionist approaches to music modeling are presented by Bharucha and Todd (1989) and Todd and Loy (1991). Among these methods, recurrent neural networks such as *simple recurrent networks* (SRN) proposed by Elman (1990) or the *long short-term memory* architecture (LSTM) by Hochreiter and Schmidhuber (1997) can successfully deal with sequences of symbols because of their ability to encode the events' context and the fact that the size of the context is not fixed. Recurrent neural networks have been applied to musical tasks. For instance, Mozer (1994) applies recurrent neural networks to melody and chord expectation tasks and shows how to learn musical structures in an invariant form. More recently, Eck and Schmidhuber (2002) use the LSTM to learn Blues improvisations. Self-Organizing maps can also be used to model music sequence perception (Tillmann et al. 2000). Alternatively, Markov-chain models such as N-grams can be considered, because they provide a simple and efficient way of learning the structure of sequences of events from a probabilistic point of view. Such techniques have been long considered in musical applications from machine improvisation (Lartillot et al. 2001) to cognitive modeling of music perception such as Ferrand et al. (2002). We refer to Pearce and Wiggins (2004) for a detailed review of monophonic musical sequence modeling, in which several N-gram implementations are compared. Other approaches make use of Markovian modeling to learn the structure of musical sequences in an interaction setting, the best known being Pachet's Continuator (Pachet 2003). Assayag and Dubnov (2004) use the oracle factor (Allauzen et al. 1999) to encode hierarchically the presented sequences. This latter work has been applied to audio signals by using an acoustic front end. Jehan (2005) considers learning and prediction applied to a wider range of musical audio signals. The author suggests several computational approaches to the prediction of musical features (e.g. downbeat prediction), but does not consider the prediction of a representation of the musical sequence, which involves several dimensions.

2.2. *Mid-level description of audio signals*

In the last two decades, there has been a substantial amount of work aimed at providing a mid-level description of musical audio. Here, we narrow our presentation to works aimed at producing a mid-level description of musical signals in terms of timing and timbre.

2.2.1. Time description: when

First, the incoming audio stream can be described in terms of onsets, that is, the beginning of a musical note or sound. This can be done, for instance, using psychoacoustical knowledge (Klapuri 1999), deriving a detection function from spectral features (Brossier et al. 2004, Collins 2004) or using supervised learning approaches (Lacoste and Eck 2007). The musical stream can also be described in terms of beats which characterize the stream periodicity even in the absence of clear perceptual attacks (Scheirer 2000, Dixon 2001, Goto 2001, Gouyon and Herrera 2003, Davies et al. 2005, Desain and Honing 1999,

Smith 1996)

2.2.2. *Timbre description: what*

Apart from the time dimension, another mid-level description we are interested in is the nature of the sounds whose attack is detected. Applied to a melody this would correspond to pitch detection, but from a broader point of view this corresponds to timbre categorization. In music information retrieval, classification of unpitched sounds has been addressed by Herrera et al. (2002) and was followed by works focusing on the transcription of percussive excerpts such as drums (Gillet and Richard 2004, Tanghe et al. 2005, Yoshii et al. 2005) or beat-box (Kapur et al. 2004, Hazan 2005). While these approaches are essentially supervised, other works have proposed unsupervised approaches to timbre categorization (Paulus and Klapuri 2003, Schwarz 2004, Hazan 2005), assuming a fixed number of timbre categories. This was followed by attempts to estimate the optimal number of categories (Gao et al. 2004, Hazan et al. 2007) or to incrementally derive hierarchies of categories (Marxer et al. 2007).

2.3. *Information-Theoretic Approaches*

Abdallah (2002) has considered redundancy reduction and unsupervised learning applied to musical and spoken audio (either waveform or spectral distribution) and has defined a measure of *suprisingness* rooted in perception and information theory. In this context, music sequence models evolve as new musical event are presented. At each point in time, the models can produce a probabilistic expectation about the structure of the musical sequence observed so far. Information-theoretic measures can be applied to characterize this random process. Abdallah and Plumbey (2007) state that "the general thesis is that perceptible qualities and subjective states like uncertainty, surprise, complexity, tension, and interestingness are closely related to information-theoretic quantities like entropy, relative entropy, and mutual information." Pearce and Wiggins (2004) use *entropy* and *cross-entropy* to evaluate whether statistical models can learn symbolic monophonic melodies. The *entropy rate*, denoted $H(X|Z)$, reflects the instantaneous certainty of statistical models to characterize the current observations X given past observations Z, while *cross-entropy* applied to test melodies informs about the generalization accuracy of a learned statistical model. Abdallah and Plumbey (2007) extend these ideas and propose to compute the *average predictive information rate*, noted $I(X, Y|Z)$ which may be seen as the average rate at which new information arrives about the present and future observations X and Y, given past observations Z. In an experiment using a very simple Markov-chain model applied to two Philip Glass minimalistic music pieces, the mentioned authors show that these measures reveal the structure of the pieces in agreement with the judgment of a human expert listener. Also, Dubnov et al. (2007) have proposed an algorithm to build causally Prediction Suffix Trees so as to describe the redundancies of attended audio signal.

We take an intermediate approach in which we focus on the symbolic expectation of musical audio signals through the encoding of beat-relative timing and timbre features using a causal approach. A proof of concept for these ideas has been implemented by Hazan et al. (2007), in which the expectation of an attended stream can be sonified using concatenative synthesis. In this paper, we extend and refine this framework by introducing several alternatives to timbre and timing description, cluster estimation and assignments. Furthermore, we propose and compare three ways of combining time and timbre dimensions regarding the prediction of the next event. In addition, we define a set of performance measurements which enable us to characterize quantitatively the behavior of the resulting system in a loop-following experimental setting.

3. Approach

The system has the following modules: *feature extraction*, *dimensionality reduction*, and *next event prediction*. These components, all of which run simultaneously when the system is following a musical stream, are shown in Figure 1. First, the *feature extraction* module is the audio front-end, which extracts timbre descriptors, onsets and beats from the incoming signal. This module is based on the aubio library (Brossier 2006). Each extracted hit is encoded in the *dimensionality reduction* and quantization module based on both time and timbre description, following an unsupervised scheme. Therefore, we obtain a symbolic representation of the incoming events, to be used by the *next event prediction* module.

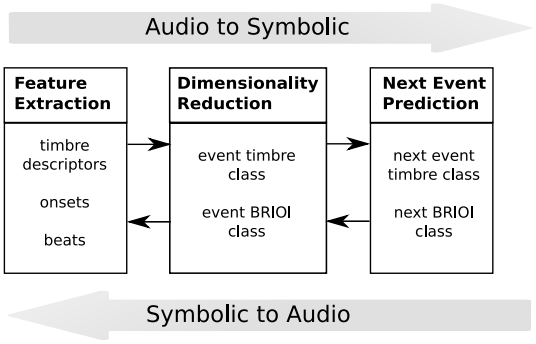

Figure 1. System diagram. Feedforward connections (left to right) create a stream of symbols to be learned. Feedback connections (right to left) enable symbolic predictions to be mapped back into absolute time.

3.1. *Low and Mid-level Feature Extraction*

3.1.1. Analysis Settings

We use a window size of 1024 samples (23 ms using a sampling rate of 44 KHz) with 50 percent overlap. We apply a Hamming window before computing the Fast Fourier Transform to perform spectral analysis.

3.1.2. Temporal detection

Onsets and beat locations are extracted as events are presented. We present the methods used for achieving both tasks and show how onsets and beats locations are combined to produce a tempo-independent timing characterization between successive events.

Onset detection We compare different onset detection techniques that are available in the aubio library. The methods we compare are the following:

- High-Frequency Content (HFC, Masri (1996)), obtained by summing the linearly-weighted values of the spectral magnitudes
- Complex domain (Duxbury et al. 2003), obtained by observing the fluctuation of the spectrum in the complex domain, thus taking advantage of both phase and magnitude.
- Dual: a hybrid function which combines the complex domain function with a function based on the KL-divergence.

Details and evaluation for these algorithms are available in Brossier (2006).

Beat tracking The tempo detection algorithm is based on Davies et al. (2005). This algorithm is based on the autocorrelation of the onset detection function. A comb filter is

applied to the resulting autocorrelation function, leading to an histogram of the period candidates. The histogram peak is then selected as the detected period, denoted $Period(t)$.

3.1.3. Inter Onset Intervals Characterization

We propose here alternatives to characterize timing relations between events. As an outcome of the temporal detection, the duration between successive events can be measured. These intervals can be seen as an absolute difference of onset times, or beat-relative difference.

Inter Onset Interval If an onset occurs at time t_{curr}, and the previous onset has occurred at time t_{prev} we can derive the absolute Inter Onset Interval (IOI) as:

$$IOI(t) = t_{curr} - t_{prev} \tag{1}$$

Beat-Relative Inter Onset Interval Based on this, for each new event, the beat-relative inter-onset interval (BRIOI) is computed as follows:

$$BRIOI(t) = \frac{IOI(t)}{Period(t)} \tag{2}$$

where $IOI(t)$ refers to the inter-onset interval between the current event onset and the previous onset, and $Period(t)$ refers to the current extracted beat period.

Cluster-wise Inter Onset Intervals Each incoming event belongs to a timbre class. In Section 3.2 we propose an unsupervised approach to assign a timbre symbol to incoming events. This makes it possible to compute IOI between events that belong to the same timbre category, instead of characterizing IOI between successive events of distinct timbre classes.

3.1.4. Timbre Description

We use 13 Mel-Frequency Cepstrum Coefficients (Davis and Mermelstein 1980). We have implemented the MFCC in the *aubio* library (Brossier 2006) following Slaney's MATLAB implementation. To provide a one-dimensional description of a detected onset, we compute the median of each coefficient over a 100 ms window starting from the onset frame.

3.2. *Dimensionality reduction and quantization*

3.2.1. Bootstrap step

Before starting to effectively encode and expect musical events, the system accumulates observations and therefore acts as a short-term memory buffer to gather statistics based on the incoming hits. During this accumulation period, which is fixed to 40 detected events, the system does not provide any prediction regarding the future events. The processes involved here are (a) feature normalization, (b) Principal Component Analysis (PCA) for dimensionality reduction and (c) estimation of the number of clusters, for both timbre features and BRIOI. While (a) and (c) are always performed, (b) is only applied to the timbre features, and it is optional.

Bootstrap feature preprocessing

- Feature normalization: we normalize the accumulated timbre descriptors and BRIOI so that they have zero-mean and unit variance. The initial distribution parameters are stored so that any normalized instance can be mapped back into its initial feature space.
- PCA: A PCA can be trained on the bootstrap normalized timbre features. In this case, instead of choosing the target dimensionality, we choose the desired amount of explained variance of the projected set compared to the original set.

This information is stored as it enables to subsequently perform normalization and dimensionality reduction on new data (Figure 1, left to right connection), or the expand and apply inverse normalization the projected data (Figure 1, right to left connection). Additionally, we also store the normalized and projected short-term history, which is used to estimate the number of clusters to work with. This step is presented in next paragraph.

Evaluating the number of symbols The number of clusters to represent both BRIOI and timbre events influences the performance of the system and has to be chosen carefully during the bootstrap step. We perform a first cluster estimation using a grid of Gaussian Mixture Models with diagonal covariance matrix, trained with the Expectation-Maximization (EM) (Dempster et al. 1977) algorithm, following a voting procedure derived from Cournapeau (2006). The grid has size $R * M$, where M is the maximum number of clusters we allow, and R is the number of independent runs. Each column of the grid represents R models with an increasing number of clusters, from one to M. We train each grid model with EM, using 20 iterations. Once the grids are trained we proceed to the model selection step, as explained below.

Information criteria for model selection. Each grid model can be described with the following parameters. First, the maximized likelihood, denoted by L, is a quantitative measure of how the trained model fits the data. The number of free parameters K^f, measures how complex the model is. The number of samples N is the number of instances present in the short-term history (see previous paragraph). From this, different information criteria (IC) have been used in the model selection literature, which are described below. First, the Bayesian Information Criterion (BIC) (Schwarz 1978) is defined as follows:

$$BIC = -2\ln(L) + K^f \ln(N) \tag{3}$$

The BIC strongly penalizes complex models. Models with few parameters and which maximize the data likelihood minimize the BIC. Akaike (1974) proposes another information criterion which penalizes less the model complexity, but does not take into account the amount of available data.

$$AIC = 2K^f - 2\ln(L). \tag{4}$$

In Section 4, we will compare the performance of the system when either BIC or AIC are used to determine the number of clusters in the data. To decide the final number of clusters we compute the median over the cross-runs:

$$K = \text{median}_{1\leq i\leq R}(\text{argmin}_{1\leq j\leq M}(IC)) \tag{5}$$

As an outcome, the estimated number of timbre (respectively IOI) clusters is noted K_{Timbre} (respectively K_{IOI}).

3.2.2. Running state

Once the clusters have been estimated for timbre features and BRIOI, we have to generate cluster assignments for incoming instances and update the clusters to take into account these new instances. To achieve this, we use an on-line k-means algorithm, with each cluster mean vector being initialized by the GMM model selected at the end of the bootstrap step. A cluster is assigned to each instance x following:

$$k_x = \text{argmin}_{1<k<K}||x - \mu_k|| \tag{6}$$

where μ_k is the k-th cluster mean. Then the mean of the assigned cluster is updated following:

$$\Delta\mu_{k_x} = \eta(x - \mu_{k_x}) \tag{7}$$

Here η is the *learning rate*, which controls how much each new instance influences the mean update of its assigned cluster. Values near zero have the effect of quantizing the incoming instance to the cluster mean, while values near one tend to shift the cluster mean towards the instance assigned to it. We have experimented with values comprised between 0.1 and 0.9, and have also implemented an optimal learning rate schedule, following Bottou (2004). In this latter case, the learning rate depends on how many instances a given cluster centroid represents.

Then the optimal learning rate can be computed as:

$$\eta_k^{opt} = \frac{1}{n_k} \tag{8}$$

where n_k represents the current number of data points in cluster k.

We illustrate the overall timbre encoding process in Figure 2 using a commercial *drum'n bass* pattern. The normalized distance involved in the cluster assignment step is obtained at the end of the bootstrap step. A principal component analysis has been used to project the internal timbre description into two dimensions.

Finally, we show in Figure 3, the BRIOI cluster assignments when processing the same excerpt. The figure shows the unclustered BRIOI histogram (bottom), and the histogram of clustered BRIOI (top), i.e. in which each BRIOI event has been substituted by the current mean of its assigned cluster.

3.3. *Next Event Prediction*

The prediction module has to deduce the most likely future events based on the sequence observed so far. We treat the incoming encoded signal as a sequence of symbols and use a symbolic expectation algorithm. In this work, we use the Prediction by Partial Match (PPM) (Cleary and Witten 1984) algorithm. In N-gram modeling, the probability distribution of the next symbol is computed based on the count of the sub-sequences preceding each possible symbol. PPM is a multiscale prediction technique based on N-grams, which has been applied to lossless compression and to the statistical modeling of symbolic pitch sequences (Pearce and Wiggins 2004). The probability distribution of the next symbol e_i,

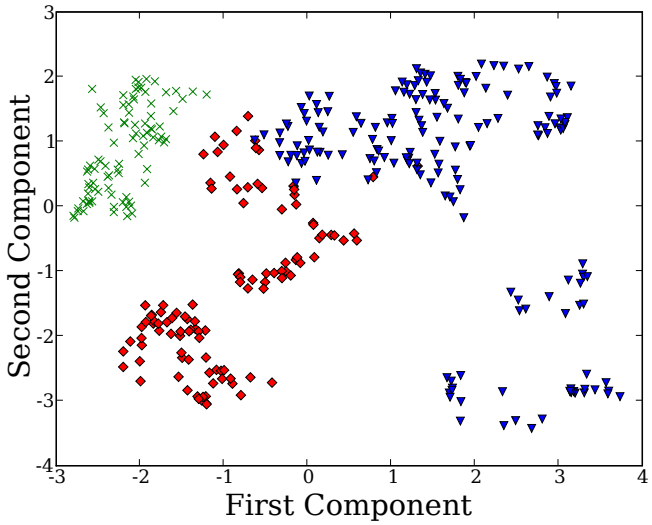

Figure 2. Timbre clusters assigned to each event after exposure to a commercial *drum'n bass* pattern. The timbre descriptors are MFCC. Crosses, squares and triangle represent points assigned to a specific timbre cluster.

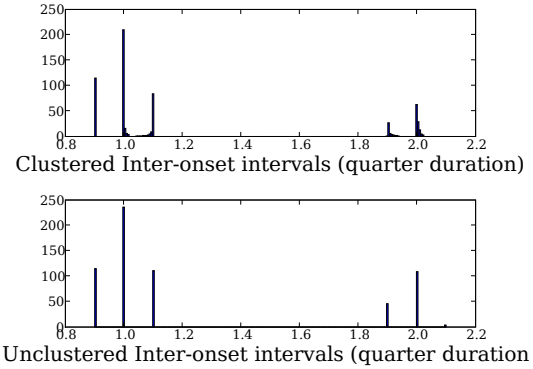

Figure 3. Unclustered and clustered BRIOI histograms after exposure to a commercial *drum'n bass* pattern

where $1 \leq i \leq K$, given the context sequence $e_{(i-n)+1}^{i-1}$ is:

$$p(e_i|e_{(i-n)+1}^{i-1}) = \begin{cases} \alpha(e_i|e_{(i-n)+1}^{i-1}) & \text{if } c(e_i|e_{(i-n)+1}^{i-1}) > 0 \\ \gamma(e_{(i-n)+1}^{i-1})p(e_i|e_{(i-n)+2}^{i-1}) & \text{if } c(e_i|e_{(i-n)+1}^{i-1}) = 0 \end{cases}$$

where $c(e_i|e_{(i-n)+1}^{i-1})$ is the number of counts of each symbol e_i following the subsequence $e_{(i-n)+1}^{i-1}$. The symbol counts, given each possible subsequence of size n, are stored in a transition table containing K^n rows and K columns. When a symbol has not appeared after $e_{(i-n)+1}^{i-1}$, the model performs a recursive *backoff* to a lower-order context. Here we use a PPM model with escape method C (Moffat 1990) and update exclusion, which provides a reasonable tradeoff between accuracy and complexity.

$$\gamma(e_i|e_{(i-n)+1}^{i-1}) = \frac{t(e_{(i-n)+1}^{i-1})}{\sum_K c(e_{(i-n)+1}^{i-1}) + t(e_{(i-n)+1}^{i-1})} \tag{9}$$

$$\alpha(e_i|e_{(i-n)+1}^{i-1}) = \frac{c(e_i|e_{(i-n)+1}^{i-1})}{\sum_K c(e|e_{(i-n)+1}^{i-1}) + t(e_{(i-n)+1}^{i-1})} \tag{10}$$

In Equations 10 and 9, the quantity $t(e_i^j)$ is the number of different symbols which have appeared in the subsequence e_i^j, $j > i$.

3.3.1. Expectation Schemes

The PPM predictor presented above produces an expectation of the next symbol to be observed given the observed context. We are interested in providing two predictions, namely next IOI symbol and next timbre symbol. We propose to compare three expectation schemes whose graphical models are illustrated in Figure 4.

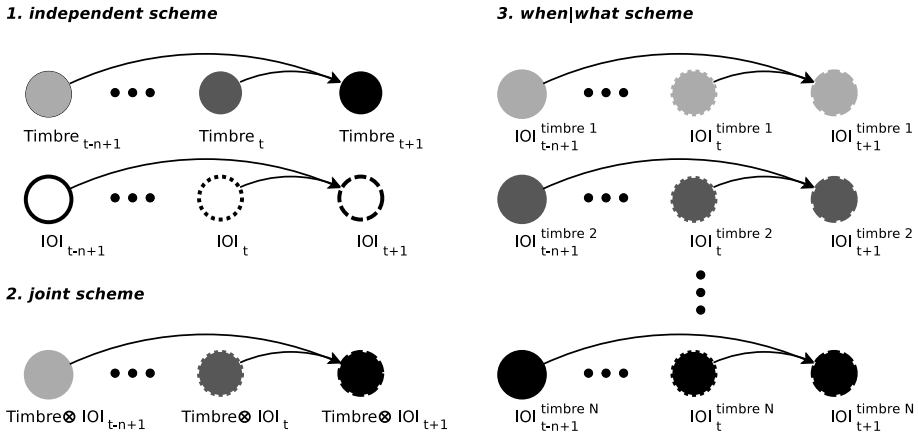

Figure 4. Graphical models of three schemes for combining what and when prediction

Independent what/when prediction scheme Two independent PPM symbolic predictors are used. The random variables Timbre and IOI are thus considered independent.

Joint prediction scheme After the bootstrap step, the number of timbre (respectively IOI) symbols K_{Timbre} (respectively K_{IOI}) has been determined. From this, a new set of symbols denoted Timbre⊗IOI is created. This set contains exhaustive combinations of timbre and IOI symbols, therefore it has $K_{Timbre} * K_{IOI}$ elements. This approach can lead to high memory requirements if $K_{Timbre} * K_{IOI}$ becomes high. In this case, a unique PPM predictor is used to predict the symbol $Timbre \otimes IOI_{t+1}$.

When|what prediction scheme Each timbre cluster is associated with a specific cluster-wise IOI symbol predictor (see Section 3.1.3). This means that for each timbre cluster, there is a predictor which provides a guess of when an event belonging to the same timbre cluster will appear.

3.3.2. Unfolding time expectation

Based on the symbolic expectation generated, we can produce a timbre and BRIOI symbol expectation. From this, we apply an inverse normalization of the mean of the chosen BRIOI cluster and scale it to the current extracted tempo to obtain the absolute time position of the expected onset for the expected timbre cluster. Following we show in Figure 5 both transcription and expectation timelines obtained during exposure to a drum example. Here K_{Timbre} equals 3. While the beginning of the expectation timeline

only contains events of timbre cluster #1 with random inter onset intervals, after a few seconds events from timbre cluster #3 start to be predicted following the transcription pattern. Then, the IOI pattern involving timbre cluster #1 events is learned, progressively followed by the timbre cluster #2 pattern.

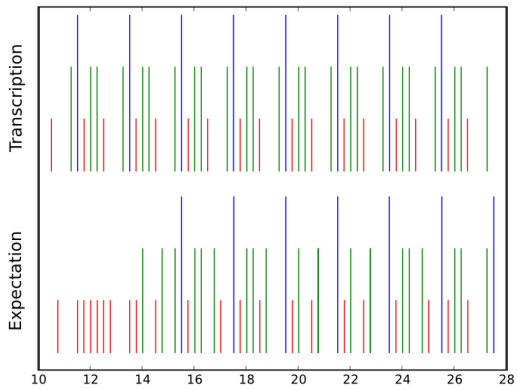

Figure 5. Comparison of transcription (top) and expectation (bottom) during exposure to an artificial drum pattern

4. System evaluation

In this section we present an evaluation of the system using a database of drum patterns. We first introduce a range of performance metrics for this task. We then present an experiment involving predictive learning of drum patterns.

4.1. *Performance metrics*

Our system produces an on-line transcription of the incoming audio stream and estimates, for each run, the optimal number of clusters to be used to produce a transcription. The transcription is used in turn to produce an expectation timeline which contains the same number of clusters than the transcription. Consequently, by using a database of annotated audio excerpts, several comparisons can be made. First, the transcription accuracy can be computed. Then, the transcription and the expectation produced by the system can be compared, without taking into account the annotations. To this respect, it would be desirable to know about violations of expectancies when human listeners learn these patterns. Unfortunately such experiments have not yet been performed. Finally the expectation timeline can be compared to the ground truth annotations. In this section, we present the performance metrics we use to achieve all these steps.

4.1.1. Comparison with the ground truth

Precision and recall applied to unsupervised transcription If the sequence transcribed is labeled we can evaluate the analysis derived from the event detection and unsupervised clustering processes. We use a measure introduced in Marxer et al. (2007) and Marxer et al. (2008) that is designed to evaluate clustering when the mapping between the reference classes and estimated clusters is unknown. The confusion matrix is first constructed by using the onset matching technique presented in Brossier (2006) adapted to multiple classes of onsets. Let us consider C ground truth classes and K clusters. We write $n_{c,k}$

the number of co-occurrences of class c and cluster k, n_c the total number of occurrences of class c. Then we can express the precision and recall as:

$$P(c,k) = \begin{cases} 1 - \frac{\sum_{1 \leq i \leq C, i \neq c} n_{i,k}}{\sum_{1 \leq i \leq C, i \neq c} n_i} & \text{if } C > 1 \\ 1 & \text{otherwise.} \end{cases} \quad (11)$$

$$R(c,k) = \begin{cases} \frac{n_{c,k}-1}{n_c-1} & \text{if } n_c > 1 \\ 1 & \text{otherwise.} \end{cases} \quad (12)$$

The pairs of precision and recall of each cluster are integrated to achieve precision and recall measures per class. The integration is performed by doing a weighted average of the precision and recall values of the co-occurrences, among all occurrences of class c. The total precision and recall measures are the weighted sums of the per-class measures.

We will use P and R to evaluate the transcription and prediction accuracy of our system. When comparing the transcription timeline with the ground truth we will call the precision measure P *Transcription Incremental Precision (TIP)* and R *Transcription Incremental Recall (TIR)*, and derive from these measures the *Transcription Incremental F-Measure ($TIFM$)*. When comparing the expectation timeline with the ground truth we will use the terms *Expectation Incremental Precision (EIP)* for P and *Expectation Incremental Recall (EIR)* for R, and derive from these measures the *Expectation Incremental F-Measure ($EIFM$)*

Other useful metrics In addition to the measures defined above, we also consider a simple metric which compares the complexity of the system representation with the timbre complexity of the attended signal. This can be done straightforwardly by computing the class to cluster ratio as follows:

$$CCR = \frac{K_{Timbre}}{C_{Timbre}} \quad (13)$$

where C_{Timbre} is the number of different annotated timbre labels for this excerpt, and K_{Timbre} has been estimated at the end of the bootstrap step (see Equation 5).

Finally, it may happen, if the sensitivity of the system is low, that the bootstrap step leads to an estimate of timbre clusters equal to one. This is clearly not desirable in the context of evaluating a system which combines time and timbre dimension. For this reason, we introduce another statistic, which we write $P1$, and define as the percentage of runs in which the estimate of clusters led to one timbre cluster.

4.1.2. Comparing expectation and transcription

Comparing expectation and transcription timelines is an easier task than a comparison against the ground truth, mainly because both timelines share the same cluster representation. That is, the list of transcribed event onsets indexed by cluster n can be compared directly to the the list of expected event onsets indexed by cluster n by using onset detection related measures. We could compute the average F-measure by comparing both transcribed and expected timbre cluster onset times for each of the timbre clusters. However, such metrics cannot be considered here because of the inaccuracy of the unsupervised encoding we use. That is, the encoder provides an approximate estimation of the number of clusters (e.g. the encoder returns an estimate of five timbre clusters for an excerpt containing three instruments). Consequently, during the running state few of these clusters are indeed used, because a vast majority of the incoming instances are assigned to a subset of

the estimated clusters (in the example, three of the five estimated clusters). Consequently, the underrepresented clusters are likely to return very low F-measures, which may in turn affect the average computed F-measure.

In Hazan et al. (2007) we have proposed to use the weighted average F-measure, which is defined as follows:

$$WFM = \sum_{i=1}^{K_t} w_i F_i \tag{14}$$

where K_t is the number of timbre clusters, each w_i is obtained by dividing the number of onsets assigned to cluster i by the total number of onsets, and F_i is the standard F-measure (with +/-50ms tolerance windows) between onsets assigned to cluster i. The individual cluster-wise F-measures involved in the resulting average computation are weighted by the proportion of events appearing in that cluster. This enables us to reduce the contribution of unused or scarcely used timbre clusters.

4.1.3. Information-theoretic viewpoint: expectation entropy

For each incoming event, the entropy of each expectation can provide information about the certainty of the returned prediction. For each predictor, the entropy can be computed as follows:

$$H(p) = -\sum_{1 \leq i \leq K} p(e_i) log_2 p(e_i) \tag{15}$$

where $p(e_i)$ is the next event estimated probability distribution over the set of possible symbols K.

In Figure 6, we show the entropy (Equation 15) of both BRIOI and timbre predictors for the commercial *drum'n bass* excerpt we used as a running example. The basic loop boundaries, which are unknown by the system, are shown using vertical lines. We observe an overall decreasing trend in the entropy curve. The basic loop consists itself of four variations of the same rhythmic pattern, and this internal structure appears plotted in the figure. Although we will not use the expectation entropy statistics in the experiment presented below, the expectation entropy signal may then be used to mark temporal cues and perform some segmentation, meter detection and accent detection tasks. This will be investigated in future work.

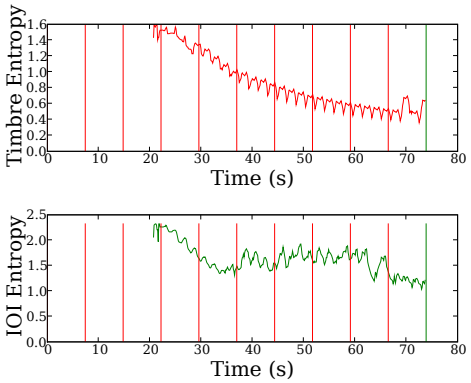

Figure 6. Instantaneous Entropy of timbre (top) and BRIOI (bottom) predictors for a commercial drum'n bass excerpt. Globally, the entropy displays a decreasing trend that corresponds to the learning of the excerpt. Locally, we can see repeating patterns reflecting the loop structure.

Table 1. System default parameters

Parameter	Value
Descriptor set	MFCC
Onset detection threshold	0.6
Onset detection Method	Dual
PCA explained variance	0.6
# Bootstrap events	40
Maximum N-gram order	2
Model Selection Criterion	AIC
On-line K-means η	η^{opt}

4.2. *Experiment: Loop following*

Based on the performance metrics introduced above, we present an empirical evaluation based on a database of drum loops. The task, material and system settings are presented, then we present and discuss the results. The task we propose to simulate is the following: the system is initialized and it does not access any previously trained model; it is exposed to a drum loop repeated several times; once the system has accumulated enough information to perform the bootstrap step and both timbre and IOI symbols are defined, the transcription and expectation timelines are produced. From this, we can compute, at each loop repetition, the statistics presented in previous section.

4.2.1. Material

We have selected a subset of audio recordings from the ENST-Drums database (Gillet and Richard 2006). We use polyphonic drum loops played by one drummer, namely drummer #2 in the database. The database consists of 49 drum excerpts (called phrases in the database) of 9 different styles, for a total of 5627 events. Most of the database patterns have a duration of approximately 10 seconds, that is, slightly more than the time needed to perform the bootstrap computation. In order to observe the learning dynamics of the system, we need to work with longer excerpts. We therefore have edited and looped the original signals 2, 4, 8 and 10 times, depending on the experiment.

Annotations preprocessing As stated by Gillet and Richard (2006), each drum excerpt was annotated following a semi-automatic process. For instance, each cymbal has a distinct label and there is a distinction made between open hi-hat and closed hi-hat. Because each drum excerpt was recorded using 8 microphones, simultaneous strokes were also precisely annotated. The average number of timbre labels present in each excerpt included in our subset is 5.5.

Because our system is processing the input stream with a fixed frame rate (the time precision is 11.6 ms) and performs timbre clustering as if the input was monophonic, we decide to apply the following ground truth preprocessing. If consecutive onsets follow one after the other so that they lie into the same analysis frame, we merge the annotated labels into one joint label and keep as onset time the first onset of the consecutive attacks.

4.2.2. Experimental setting

The system is designed to learn in a causal way, therefore it may learn more accurately when being exposed to the excerpts several times: this is why we have experimented with a number of repetitions of the basic loop of 2, 4, 6, and 8. The other parameters (e.g. onset detection threshold, model selection information criterion, timbre PCA explained variance, next event predictor context size) are varied for comparison purposes as explained below. Unless explicitly notified, the default parameters we use are listed in Table 1.

Table 2. Onset detection F-measure, depending of the method used and the peak-picking threshold.

threshold/method	complex	dual	hfc
0.1	0.587	0.740	0.622
0.2	0.601	0.755	0.636
0.3	0.614	0.752	0.647
0.4	0.625	0.760	0.654
0.5	0.634	0.763	0.661
0.6	0.644	**0.764**	0.667
0.7	0.650	0.763	0.671
0.8	0.659	0.763	0.675
0.9	0.664	0.762	0.678

4.3. *Implementation*

Our current implementation is named *billabio*, as it uses concepts from both *aubio*[1] (Brossier et al. 2004) and *billaboop* (Hazan 2005) projects. The components of billabio are written in the Python programming language, which the exception of the audio analysis processes, which are implemented in C. Both dimensionality reduction and next event prediction modules, constituting the machine learning algorithms, are implemented using numpy[2]. Additionally, we use the *em* package (Cournapeau 2006) for applying the Expectation Maximization Algorithm, and the *mdp* package (Berkes and Zito 2007) for performing the Principal Component Analysis.

4.4. *Results*

This section presents the results of the evaluation whose details are presented in previous section. Because our expectation system is made of several components, we first present the evaluation of these components in isolation. Then, we provide an overall system evaluation based on the system transcription and expectation accuracies.

4.5. *Evaluation of system components*

It is possible to evaluate some of the system components by providing them as input the data which can be extracted from the ground truth. Given the ground truth annotations (i.e. onset times and labels), we can evaluate some processing stages of the system, such as onset detection and timbre clustering, in isolation. Other stages cannot be directly evaluated with the annotations. The ENST-Drum data is not beat-marked, consequently we have not performed an evaluation of the beat tracking component, and refer to Brossier (2006) for an evaluation of this component. Also, some components rely on a symbolic representation of time. This is the case of the IOI clustering component which provides IOI cluster labels as output, and the next-event prediction component, which uses IOI cluster labels as input and output. We refer to Pearce and Wiggins (2004) for an evaluation of the symbolic expectation component. Consequently, we present here the results of the onset detection and timbre clustering processes.

4.5.1. Onset Detection

The evaluation of the onset detection stage is provided in Table. 2. We compare the onsets extracted by the onset detector component with the onset times provided by the ENST ground truth.

Table. 2 shows that the F-measure is maximized using the dual detection method. In this case, the threshold setting which leads to the best F-measure is 0.6. In subsequent

[1] `http://aubio.org`
[2] `http://numpy.scipy.org`

experiments, we will set the parameters of the onset detection component to method dual with threshold 0.6.

4.5.2. *Timbre Clustering*

We report here the results obtained when evaluating the timbre clustering component. We propose two evaluation settings in which the inter-onset regions processed by the timbre clustering component are obtained differently. In the first configuration, the regions correspond to the onsets provided by the ground truth of the ENST-drums database. Thus we evaluate the timbre clustering component in isolation. In the second configuration, the audio regions correspond to the onsets computed by the onset detection component, consequently we evaluate the overall accuracy of the combination of onset detection and timbre clustering components.

The timbre clustering accuracy is largely influenced by the outcome of the bootstrap process, in which a PCA can be trained on the bootstrap data, and the estimation of the optimal number of clusters is performed. For this reason, we vary the PCA desired explained variance between 0.1 and 0.8 (or do not perform PCA at all), and the information criterion to be used (either BIC or AIC).

In Table 3 we present the mean transcription statistics obtained when varying these parameters. For each run, we report the Transcription Incremental F-measure (TIFM), the class to cluster ratio (CCR) and percentage of estimates, and the percentage of estimates leading to one timbre cluster (P1). The second and third columns present the results obtained using the ground truth onsets, while the forth and fifth columns present the results using detected onsets.

On the one hand, the results corresponding to ground truth onsets show that the combination of the amount of data compression (controlled by the PCA desired variance) and the information criterion play an important role in the out-coming timbre representation. In all cases, the TIFM lies between 0.509 and 0.572. The TIFM average for runs based on AIC (respectively BIC) is 0.551 (respectively 0.536). However we see that AIC-based estimation leads to a better estimation (mean CCR 0.753) than the BIC-based estimation (mean CCR: 0.514). If both criteria tend to estimate less clusters than the ones available in the ground truth, the AIC estimation tends to produce a higher number of timbre clusters. This is reflected in the P1 measure: AIC-based runs estimate a single cluster less frequently than BIC-based runs. Overall, the AIC criterion can be seen as more sensitive than the BIC. From this, the choice of the PCA explained variance is a tradeoff between higher TIFM (low PCA explained variance) or higher CCR (high PCA explained variance). Situations resulting in one single cluster render the clustering process useless, since there is no information gain. The P1 statistic shows that the BIC-based estimation leads to a unique timbre cluster in 20.21% of the runs in average, while the P1 of AIC-based runs has a mean of 0.6%. For the subsequent experiments, we choose to perform the bootstrap estimation with the AIC criterion and a PCA explained variance of 0.6, as a tradeoff to maximize both TIFM and CCR.

4.6. *Expectation*

After the system has completed the bootstrap step it generates expectations. From this moment, we can evaluate how generated expectations match both transcription and ground truth. We now present the distribution of the expectation statistics obtained when attending each excerpt of the ENST subset as described in previous section.

4.6.1. *Influence of exposure*

In this experiment, we aim to show the impact of exposure in the system predictive accuracy. Our initial guess is that the system is sensitive to exposure, but we aim at quantifying how each expectation scheme is sensitive to repeated patterns. For each expectation

Table 3. Timbre clustering statistics for different bootstrap settings depending on the PCA desired explained variance ("No" means no PCA is applied during bootstrap). The columns show the information criterion used. AIC_{GT} and BIC_{GT} corresponds to runs that use the ground truth onsets. AIC_{TR} and BIC_{TR} correspond to runs using the detected onsets. For each run, the measures presented are, from left to right, Transcription Incremental F-measure, Class to Cluster Ratio, and Percentage of estimates leading to one timbre cluster.

PCA var./Information Criterion	AIC_{GT}	BIC_{GT}	AIC_{TR}	BIC_{TR}
0.1	0.572 0.521 6.382	0.539 0.369 19.178	0.572 0.521 2.127	0.539 0.369 29.787
0.2	0.565 0.519 8.510	0.525 0.353 19.148	0.565 0.519 2.127	0.525 0.353 34.042
0.3	0.570 0.495 4.255	0.539 0.366 21.276	0.570 0.495 2.127	0.539 0.366 31.914
0.4	0.549 0.654 2.127	0.556 0.468 14.891	0.542 0.659 0.000	0.539 0.441 21.276
0.5	**0.559 0.847 0.000**	0.551 0.550 10.631	0.519 0.819 0.000	0.550 0.579 19.148
0.6	0.522 0.896 0.000	0.548 0.635 8.512	**0.526 0.904 0.000**	0.557 0.590 17.021
0.7	0.536 0.958 0.000	0.545 0.649 8.512	0.516 0.899 0.000	0.553 0.569 23.404
0.8	0.545 0.956 0.000	0.515 0.620 19.143	0.522 0.883 0.000	0.558 0.475 25.531
No	0.545 0.933 0.000	0.509 0.624 19.571	0.502 0.891 0.000	0.556 0.469 26.341

scheme, we report in Figure 7 four independent runs in which we present to the system each drum pattern repeated 2, 4, 6 and 8 times. All schemes are characterized by a high EIP (greater than 0.65 in all cases) and a lower EIR (lower than 0.42 in all cases).

As expected, when the number of repetitions increases we observe an increase of all expectation statistics. In the case of the joint and independent schemes, the WFM has the biggest increase, which means the PPM expectator can take advantage of the repetitions to learn to provide a prediction which fits its transcription. The other expectation statistics, which are related to the ground truth, also increase -to a lesser extent- with an increasing number of repetitions. The when|what scheme expectation statistics are less affected by an increase in the number of repetitions. For this scheme, the EIFM decreases when the number of repetitions goes from 6 to 8.

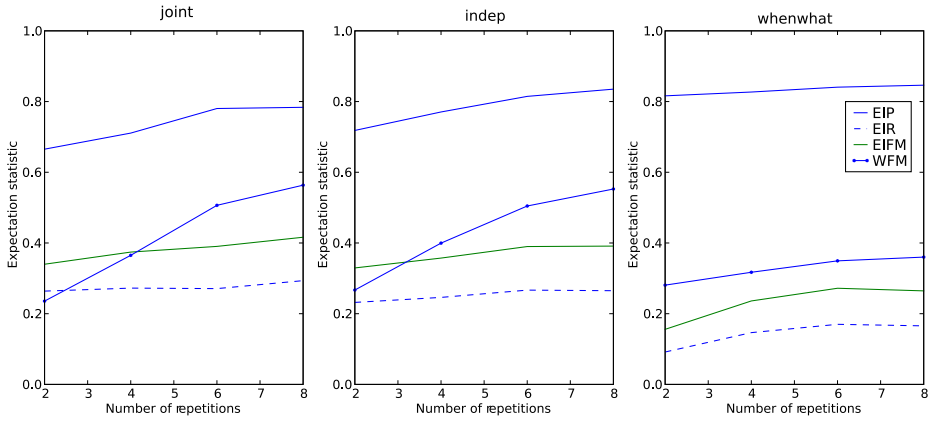

Figure 7. Comparison of expectation statistics (EIP, EIR, EIFM, WFM) as a function of the number of repetitions of a given loop. The three expectation schemes are compared. Left: joint scheme, middle: independent scheme, right: when|what scheme.

4.6.2. Influence of context size

As the PPM predictor we use to provide a prediction of the next event is based on the observation of a context of fixed size N (see Section 3.3), we are interested in measuring the impact of the prediction context size. Indeed, the number of past items involved in the posterior probability computation may affect the behavior of the learner by biasing it with predictions which are too general (low N), or by overfitting the prediction (high N). In Figure 8 we show the average expectation statistics (EIP, EIR, EIFM and WFM) plotted against the size of the context length N, which is varied from 1 to 6.

Overall, the joint scheme leads to the highest EIR and EIFM while the independent scheme leads to the highest EIP. Both independent and joint schemes exhibit a similar

Table 4. Expectation Onset detection statistics (F-measure, Precision, Recall) compared to the ground truth.

joint	indep	when|what
0.613	0.619	0.424

dependency to context size. Both models exhibit an increase of the WFM when the context size varies from 2 to 4, and the independent scheme WFM grows along the context size. This is not the case of the when|what scheme, in which the expectation statistics exhibit almost no variation with an increasing context size.

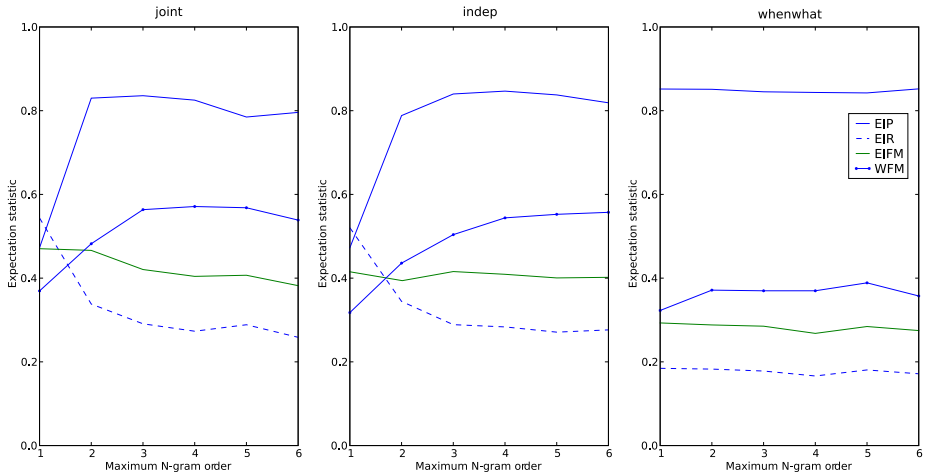

Figure 8. Comparison of expectation statistics (EIP, EIR, EIFM, WFM) as a function of the number of repetitions order used used to provide a prediction of the next event. The three expectation schemes are compared. Left: joint scheme, middle: independent scheme, right: when|what scheme.

4.7. *Expected onset detection*

Finally, we report in Table 4 the results of comparing the train of expected onsets, with the ground truth onsets. The parameters we use are the default values presented in Table 1. These figures can be compared with the results obtained in the onset detection evaluation in Table 2. The joint and independent schemes lead to an expected F-measure which is above 0.6. The when|what scheme leads to poorer onset expectation results.

Overall, the evaluation reported here shows that the expectation schemes have distinct behavior with respect to the variations of the expectation statistics depending of exposure and context size. While the joint and independent schemes have a similar behavior (the joint scheme slightly outperforms the independent scheme in most of the cases) and dependency to exposure and context size, the when|what scheme exhibit worse performance and a smaller dependency on exposure and context size. These findings will be addressed in the discussion.

5. Examples

Let us now explore our system in more detail with two examples from the ENST data base. The two drum patterns have contrasting degrees of complexity: *phrase_ disco_ sim-*

ple_ slow_ sticks (*simple disco*) and *phrase_ funk_ complex_ fast_ sticks* (*complex funk*) (Figure 9). No source separation is performed as a preprocessing step. Therefore, it should be considered how many different percussion sounds appear in the drum samples and how many different combinations appear (Table 5), since each sound combination may lead to a different cluster. We have calculated the matching matrix between the annotated onset events ('score') of a class (e.g. chh_bd=closed hi-hat and bass drum played synchronously) and the detected onsets of a cluster that has emerged in our system. In this matching matrix we can iteratively yield the maximal entry thereby establishing a connection between a row (class) and a column (cluster). After elimination the row and column of the maximal entry we determine the maximal entry again until the matrix vanishes. This procedure endows us with an optimal mapping between the classes and the clusters. In Figure 9, we display sequences of classes and clusters on the same line if they are interconnected through this mapping. For the *simple disco* pattern (Figure 9 c), it can be seen that fragments of the basic pattern bass/open hi-hat/snare/closed hi-hat are captured. The single cymbal instance is not captured. However, one cluster (second highest row) can be interpreted as detecting the sustain phase of the cymbal (three hits). The first half of the looped *complex funk* pattern is shown in Figure 9 (b) and (d). The number of extracted clusters (six) is less than the order of occurring combinations of percussion sounds (twelve). Several sound combinations occur sparsely. The mapping between sounds and clusters is not clear. The expectations cannot capture the complexity of the pattern well.

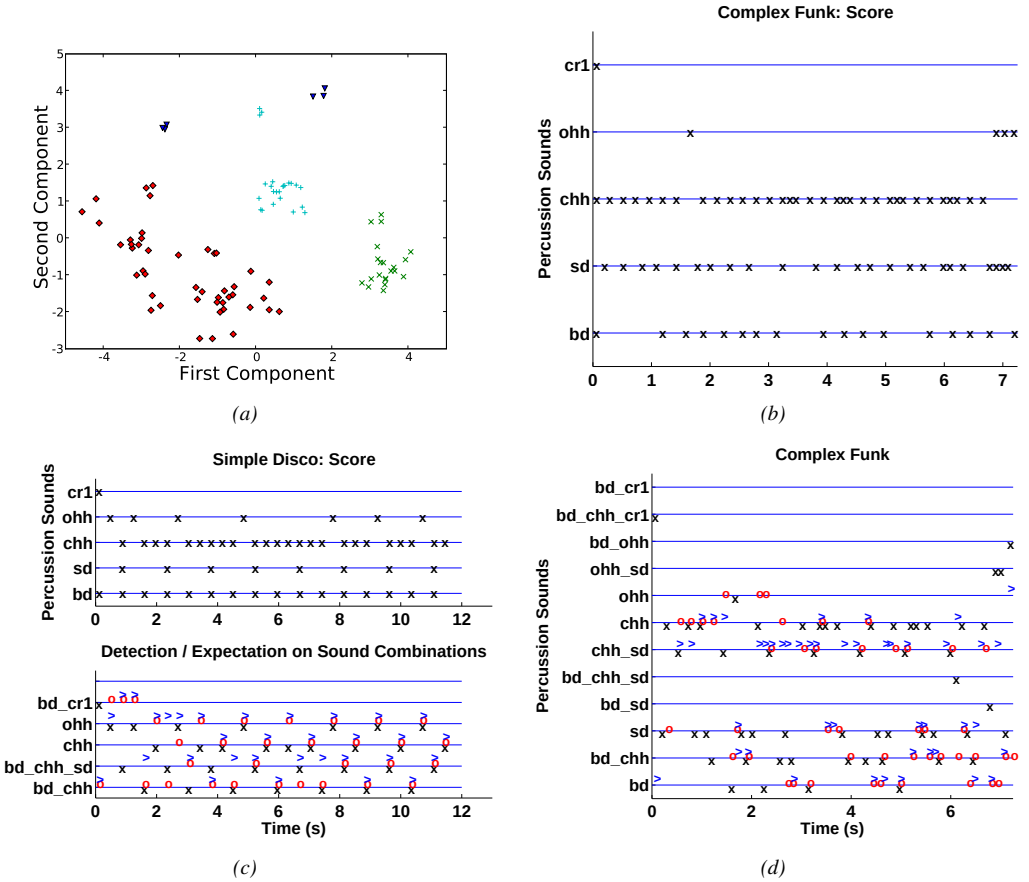

Figure 9. Clustering, score ('x'), extracted cluster sequence ('o') and expected events ('>') according to the independent scheme for a simple and a complex drum pattern. (bd=base drum, sd=snare drum, chh/ohh=closed/open hi hat,cr1=crash cymbal) (a) The first two principal components of the simple disco pattern. (b) The score of a *simple disco* pattern with alternating bass, hi-hat, snare, hi-hat. We show the 3rd run of the repeated drum pattern. The expectation is learned well. (c) Score of the highly irregular Funk percussion pattern. (d) For the highly irregular Funk pattern the system reaches its limits (cf. text).

Table 5. Numbers of percussion sound classes (perc.) and of used combinations of percussion sound classes (comb. perc.) in the two examined drum patterns.

drum loop	# perc.	# comb. perc.
simple disco	5	5
complex funk	5	12

6. Discussion and future work

We have presented a system that addresses simultaneously the detection of temporal and timbre events, their categorization, and the prediction of forthcoming ones. The idea of describing the what/when musical stream through the formation of a set of time and timbre categories is rooted in experimental findings. For instance, Srinivasan et al. (2002) study how humans can create instrument categories when exposed to acoustic cues, while the formation of rhythmic categories is investigated in (Desain and Honing 2003). Furthermore, we have discussed 3 different architectures, some of them having more physiological and cognitive plausibility than others. Independently of the architecture used, we have defined two main processing constraints, namely unsupervised learning and causal processing. In our approach, we do not cope with an incremental learning approach which is capable of constantly modifying itself by adding or removing clusters representing categories of timbre or durations. Instead, during a bootstrap phase of the system, the number of clusters is estimated and they are initialized, before generating expectations. However, the constraints make our approach distinct to engineering approaches to timbre transcription, in which the target timbre categories are known in advance, and this has motivated us to define new measures to evaluate both unsupervised transcription and unsupervised expectation tasks.

We have run an evaluation of two system components (onset detection and timbre clustering) and the combination of those two, which represent the unsupervised transcription. These experiment have shown that the way the information is represented (e.g. by varying the PCA explained variance) and the information criterion we use influence the timbre events representation. The Incremental F-measure proposed by Marxer et al. (2008) is a means of evaluating how close the set of symbols is to the ground truth labels. However, additional measures are needed to assess the sensitivity of the system. The Akaike information criterion does not penalize the complexity as much as the Bayesian information criterion does. We find that the Akaike criterion coupled with a moderate compression of the timbre features (i.e. PCA explained variance of 0.6) leads to the best tradeoff between complexity and similarity with the ground truth.

In a way, our approach of calculating expectations from duration sequences is complementary to generating expectations from the the extracted beat, beat weight, or metrical hierarchy, e.g. based on autocorrelation and wavelet analysis (Smith 1996). Although beat based expectation schemes are not considered here, especially the phase of an onset relative to the beat or some rhythmical cycle/bar are important in the investigated drum loops.

However, in none of the evaluated configurations we have been able to reach an Incremental F-measure higher than 0.572. This figure can be compared to the best performing MIREX 2005 entry Yoshii et al. (2005), with an average F-measure of 0.659. But in this latter work the task was to generate the transcription using three predefined timbre classes. Consequently, the evaluation measures differ.

To increase the transcription accuracy, we aim at investigating how to combine a short-term representation of timbre and time (e.g. bootstrap estimation) with a longer-term representation which may involve a database of predefined timbre categories.

Concerning the performance of the whole expectation system, we can make a distinction between the independent and joint schemes, for which expectation performs in the

same order of magnitude and depends on repetitions and context size, and the when|what scheme, for which expectation performs worse and depends less on training duration and context size. To explain this, we assume that the accuracy of the when|what scheme is more crucially altered by errors in the transcription, because these errors generate in turn errors in the representation of timing events by creating erroneous clusters of inter-onset intervals. From a computational point of view, these architectures may be seen as parallel processing paths. The independent and joint schemes need more information to be stored (because transitions between timbre events are also encoded). The joint scheme space requirements are higher because the transitions between all combinations of timbre and time symbols are stored. From a musical point of view, the independent and joint schemes would be able to code information about the musical surface, such as melodies or drum solos.

Contrastingly, the transitions between timbre events are implicitly coded in the when|what scheme. This makes its space requirements low when representing rhythms (e.g. the time dependencies between onsets with same timbre are simple, even if the sum of onsets over timbre forms a more complex structure). However, to be properly applied to musical audio signal, this scheme requires the transcription component to perform source separation (e.g. via independent component analysis or non-negative matrix/tensor factorization) instead of merging together simultaneous attacks and to provide more reliable results.

Whereas the independent scheme of inter-onsets captures rhythmical aspects encoded as durations between onsets, the independent scheme of timbres encodes the regularities in the pure order of the events abstracting from specific durations. The plausibility of the when/what scheme versus the independent schemes depends on the degree of streaming. If a strong tendency towards streaming yields the perception of separate synchronous rhythms (for each particular percussion sounds, e.g. the hi-hat or bass drum rhythm in isolation), the when/what scheme is preferred over the independent scheme. The more interdependent the sound classes and their durations (more precisely: IOIs) the more appropriate the joint scheme. Overall our system could take advantage of combining these three schemes, which represent different statistical and musical viewpoints for pattern matching and expectation tasks.

The work presented here emphasizes the expectation as being a central process involved in music listening. We can use expectation as a measure for musical complexity. Expectation describes structure by inducing a segmentation through points of high or low expectation. Finally, in the context of causal modeling, we can see the expectation as a dynamic top-down control which may modulate lower-level processes such as onset detection. In our what/when prediction framework, the expectation feedback can be provided to timbre-specific event detectors. In the context of musical audio analysis, we view models of music expectation as general components able to dynamically accumulate the structure of the musical environment, and where the expectation signal may help to solve more specific musical tasks.

7. Conclusion

We have presented an unsupervised and causal approach to transcribe, encode and generate *what* and *when* expectations based on constant-tempo musical audio, using both *timbre* and *timing* dimensions. Alternatives to combining these two dimensions have been presented and evaluated. We have illustrated the steps involved in the feature extraction, dimensionality reduction and expectation processes and we have compared these steps when the system processes percussive material. A set of statistics, either supervised or unsupervised, has been presented to characterize the system's learning abilities. An evaluation of the system components has been performed. This enabled us to find the most suitable pa-

rameters for the transcription component. From this, three expectation schemes, namely independent, joint, and when|what schemes have been compared. Our current results show that the joint scheme leads to the highest expectation statistics, and is slightly followed by the independent scheme. Both schemes exhibits a greater dependency to number of repetitions and context size than the when|what scheme. For this latter scheme, the lower expectation accuracy may be explained by the transcription errors which directly affect the architecture. Then, we have discussed our approach and main assumptions. We have suggested that the expectation signal (e.g. expectation entropy) may describe the structure of musical patterns in a causal manner, Finally, we mentioned future directions to improve the existing system. The contribution presented here can serve as a basis for designing an unsupervised interactive music system driven by cognitive principles, pattern recognition and predictive learning.

8. Acknowledgments

This work is partially funded by the *EmCAP* project (European Commission FP6-IST, contract 013123). We thank Joshua Eichen and Owen Meyers for proofreading this manuscript.

References

Bharucha, J.J., and Todd, P.M. (1989), "Modeling the Perception of Tonal Structure with Neural Nets," *Computer Music Journal*, 13, 44–53.

Todd, P.M., and Loy, D.G., *Music and Connectionism*, MIT Press (1991).

Mozer, M. (1994), "Neural network music composition by prediction: Exploring the benefits of psychophysical constraints and multiscale processing," *Connection Science*, 6, 247–280.

Lartillot, O., Dubnov, S., Assayag, G., and Bejerano, G. (2001), "Automatic modeling of musical style," in *Proceedings of the International Computer Music Conference (ICMC 2001)*, La Havana, Cuba.

Tillmann, B., Bharucha, J.J., and Bigand, E. (2000), "Implicit Learning of Tonality: A Self-Organizing Approach," *Psychological Review*, 107(4), 885–913.

Assayag, G., and Dubnov, S. (2004), "Using Factor Oracles for machine Improvisation," *Soft Computing*, 8(9), 604–610.

Pearce, M.T., and Wiggins, G.A. (2004), "Improved methods for statistical modelling of Monophonic Music," *Journal of New Music Research*, 33(4), 367–385.

Pachet, F. (2003), "The Continuator: Musical Interaction With Style," *Journal of New Music Research*, 32(3), 333–341.

Caclin, A., Brattico, E., Tervaniemi, M., Ntnen, R., Morlet, D., Giard, M., and McAdams, S. (2006), "Separate Neural Processing of Timbre Dimensions in Auditory Sensory Memory," *Journal of Cognitive Neuroscience*, 18(12), 1959–1972.

Sridharan, D., Levitin, D., Chafe, C., Berger, J., and V., M. (2007), "Neural dynamics of event segmentation in music: converging evidence for dissociable ventral and dorsal networks.," *Neuron*, 55(3), 521–532.

Maess, B., Koelsch, S., Gunter, T.C., and Friederici, A.D. (2001), "Musical syntax is processed in Broca's area: an MEG study," *Nature Neuroscience*.

Peretz, I., and Zatorre, R. (2005), "Brain Organization for Music Processing," *Annual Review of Psychology*, 56(1), 89–114.

Bregman, A.S., *Auditory Scene Analysis*, Cambridge, MA: MIT Press (1990).

Creel, S.C., Newport, E.L., and Aslin, R.N. (2004), "Distant Melodies: Statistical Learning of Nonadjacent Dependencies in Tone Sequences," *Learning, Memory*, 30, 1119–1130.

Rumelhart, D., and McLelland, J., *Parallel Distributed Processing*, MIT Press (1986).

Elman, J. (1990), "Finding Structure in Time," *Cognitive Science*, 14(2), 179–211.

Hochreiter, S., and Schmidhuber, J. (1997), "Long Short-Term Memory," *Neural Computation*, 9(8), 1735–1780.

Eck, D., and Schmidhuber, J. (2002), "Learning the Long-Term Structure of the Blues," *Lecture Notes in Computer Science, Proceedings of ICANN Conference*, 2415, 284–289.

Ferrand, M., Nelson, P., and Wiggins, G. (2002), "A probabilistic model for melody segmentation.," in *2nd International Conference on Music and Articial Intelligence (IC-MAI2002)*, University of Edinburgh, UK.

Allauzen, C., Crochemore, M., and Raffinot, M. (1999), "Factor Oracle: A New Structure for Pattern Matching," in *Proceedings of the 26th Conference on Current Trends in Theory and Practice of Informatics on Theory and Practice of Informatics*, pp. 295 – 310.

Jehan, T. (2005), "Creating Music by Listening," Massachusetts Institute of Technology, MA, USA.

Klapuri, A. (1999), "Sound Onset Detection by Applying Psychoacoustic Knowledge," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, USA.

Brossier, P., Bello, J., and Plumbley, M. (2004), "Fast Labelling of Notes in Music Signals," in *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2004)*, Barcelona, Spain, pp. 331–336.

Collins, N. (2004), "On Onsets On-The-Fly: Real-Time Events Segmentation and Categorization as a Compositional Effect," in *Proceedings of the First Sound and Music Computing Conference (SMC'04)*, Paris, France.

Lacoste, A., and Eck, D. (2007), "A supervised classification algorithm for note onset detection," *EURASIP J. Appl. Signal Process.*, 2007(1), 153–153.

Scheirer, E. (2000), "Music-Listening Systems," Massachusetts Institute of Technology, MA, USA.

Dixon, S. (2001), "Automatic Extraction of Tempo and Beat From Expressive Performances," *Journal of New Music Research*, 30, 39.

Goto, M. (June 2001), "An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds," *Journal of New Music Research*, 30, 159–171(13).

Gouyon, F., and Herrera, P. (2003), "Determination of the Meter of musical audio signals: Seeking recurrences in beat segment descriptors," in *Proceedings of Audio Engineering Society, 114th Convention*, Amsterdam, The Netherlands.

Davies, M.E.P., Brossier, P.M., and Plumbley, M.D. (2005), "Beat Tracking Towards Automatic Musical Accompaniment," in *Proceedings of the Audio Engineering Society 118th convention*, May, Barcelona, Spain.

Desain, P., and Honing, H. (1999), "Computational Models of Beat Induction: The Rule-Based Approach," *Journal of New Music Research*, 28, 29.

Smith, L. (1996), "Listening to musical rhythms with progressive wavelets," in *TENCON '96. Proceedings. 1996 IEEE TENCON. Digital Signal Processing Applications*, Vol. 2, pp. 508–513 vol.2.

Herrera, P., Yeterian, A., and Gouyon, F. (2002), "Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques," in *Proceedings of Second International Conference on Music and Artificial Intelligence*, Edinburgh, Scotland.

Gillet, O., and Richard, G. (17-21 May 2004), "Automatic transcription of drum loops," *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, 4, iv–269–iv–272 vol.4.

Tanghe, K., Degroeve, S., and Baets, B.D. (2005), "An Algorithm For Detecting and labeling drum events in polyphonic music," in *Mirex Drum Recognition Contest (part of International Symposium on Music Information Retreival)*.

Yoshii, K., Goto, M., and Okuno, H.G. (2005), "AdaMast: A Drum Sound Recognizer based on Adaptation and Matching of Spectrogram Templates," in *Mirex Drum Recognition Contest (part of International Symposium on Music Information Retreival)*.

Kapur, A., Benning, M., and Tzanetakis, G. (2004), "Query By Beat Boxing: Music Retrieval for the DJ," in *Proceedings of the 5th International Symposium on Music Information Retrieval (ISMIR 2004)*, Barcelona, Spain.

Hazan, A. (2005), "Towards Automatic Transcription of Oral Expressive Performances," in *Proceedings of the Intelligent User Interfaces Conference (IUI 2005)*, San Diego, CA, USA.

Paulus, J., and Klapuri, A. (2003), "Model-based Event Labeling in the Transcription of Percussive Audio Signals," in *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*, ed. M. Davies, Sep., London, UK, pp. 73–77.

Schwarz, D. (2004), "Data-Driven Concatenative Sound Synthesis," Ircam - Centre Pompidou, Paris, France.

Hazan, A. (2005), "BillaBoop: Real-Time Voice-Driven Drum Generator," in *Proceedings of Audio Engineering Society, 118th Convention*, Barcelona, Spain.

Gao, S., Lee, C.H., and wei Zhu, Y. (2004), "An unsupervised learning approach to musical event detection," *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, 2, 1307–1310 Vol.2.

Hazan, A., Brossier, P., Holonowicz, P., Herrera, P., and Purwins, H. (2007), "Expectation Along The Beat: A Use Case For Music Expectation Models," in *Proceedings of International Computer Music Conference 2007*, Copenhagen, Denmark, pp. 228–236.

Marxer, R., Hazan, A., Purwins, H., Grachten, M., Herrera, P., and Salselas, I. (2007), "Mock-up of music analysis system," Technical report, Music Technology Group, Pompeu Fabra University.

Abdallah, S.A. (2002), "Towards Music Perception by Redundancy Reduction and Unsupervised Learning in Probabilistic Models," Kingś College London, London, UK.

Abdallah, S., and Plumbey, M. (2007), "Information Dynamics," Technical report C4DM-TR07-01, Centre for Digital Music, Queen Mary, University of London.

Dubnov, S., Assayag, G., and Cont, A. (2007), "Audio Oracle: A New Algorithm for Fast Learning of Audio Structures," in *Proceedings of International Computer Music Conference (ICMC)*, September, Copenhagen, pp. 224–228.

Brossier, P. (2006), "Automatic Annotation of Musical Audio for Interactive Applications," Centre for Digital Music, Queen Mary University of London, London, UK.

Masri, P. (1996), "Computer modeling of Sound for Transformation and Synthesis of Musical Signal," University of Bristol, Bristol, U.K.

Duxbury, C., Bello, J., Davies, M., and Sandler, M. (2003), "Complex domain onset detection for musical signals," *Proceedings Digital Audio Effects Workshop (DAFx)*.

Davis, S., and Mermelstein, P. (1980), "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Pattern Recognition and Artificial Intelligence, RCH Chen, ed., Academic Press: New York*, 28(4), 357–366.

Dempster, A., Laird, N., and Rubin, D. (1977), "Maximum likelihood from incomplete data via the EM algorithm," *Proceedings of the Royal Statistical Society*, B-39, 1–38.

Cournapeau, D. (2006), "PyEM, a python package for Gaussian Mixture Models," Technical report, University of Kyoto, Graduate School of Informatics.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *The Annals of Statistics*, 6(2), 461–464.

Akaike, H. (1974), "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, 19, 716–723.

Bottou, L. (2004), "Stochastic Learning," *Advanced Lectures On Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003: Revised Lectures*.

Cleary, J.G., and Witten, I.H. (1984), "A comparison of enumerative and adaptive codes," *IEEE Transactions on Information Theory*, 30(2), 306–315.

Moffat, A. (1990), "Implementing the PPM Data Compression Scheme," *IEEETCOMM: IEEE Transactions on Communications*, 38, 1917–1921.

Marxer, R., Holonowicz, P., Purwins, H., and Hazan, A. (2007), "Dynamical Hierarchical Self-Organization of Harmonic, Motivic, and Pitch Categories," in *Music, Brain and Cognition. Part 2: Models of Sound and Cognition, held at NIPS)* Vancouver, Canada.

Marxer, R., Purwins, H., and Hazan, A. (2008), "An F-Measure for Evaluation of Unsupervised Clustering with Non-Determined Number of Clusters," Technical report, Universitat Pompeu Fabra, Music Technology Group.

Gillet, O., and Richard, G. (2006), "ENST-Drums: an extensive audio-visual database for drum signals processing," in *Proceedings of the 7th International Symposium on Music Information Retrieval (ISMIR 2006)*, October, pp. 156–

159.
Berkes, P., and Zito, T., "Modular Toolkit for Data Processing (version 2.1)," (2007).
Srinivasan, A., Sullivan, D., and Fujinaga, I. (2002), "Recognition of isolated instrument tones by conservatory students," *Proceedings of the 7th International Conference on Music Perception and Cognition (ICMPC 2002)*, pp. 720–723.
Desain, P., and Honing, H. (2003), "The formation of rhythmic categories and metric priming," *Perception*, 32, 341–365.