Queen Mary University of London

Towards the Automatic Analysis of Metric Modulations

by

Elio Quinton

A thesis submitted in partial fulfilment for the
degree of Doctor of Philosophy

in the

Center for Digital Music

July 2017

Declaration of Authorship

I, Elio Quinton, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature:

Date: 12-02-2017

Details of collaboration and publications: All publications and collaborations related to this thesis are listed in section 1.6.

A Dedette,

Queen Mary University of London

Abstract

Center for Digital Music

Doctor of Philosophy

by Elio Quinton

The metrical structure is a fundamental aspect of music, yet its automatic analysis from audio recordings remains one of the great challenges of Music Information Retrieval (MIR) research. This thesis is concerned with addressing the automatic analysis of changes of metrical structure over time, i.e. metric modulations. The evaluation of automatic musical analysis methods is a critical element of the MIR research and is typically performed by comparing the machine-generated estimates with human expert annotations, which are used as a proxy for ground truth. We present here two new datasets of annotations for the evaluation of metrical structure and metric modulation estimation systems. Multiple annotations allowed for the assessment of inter-annotator (dis)agreement, thereby allowing for an evaluation of the reference annotations used to evaluate the automatic systems. The rhythmogram has been identified in previous research as a feature capable of capturing characteristics of rhythmic content of a music recording. We present here a direct evaluation of its ability to characterise the metrical structure and as a result we propose a method to explicitly extract metrical structure descriptors from it. Despite generally good and increasing performance, such rhythm features extraction systems occasionally fail. When unpredictable, the failures are a barrier to usability and development of trust in MIR systems. In a bid to address this issue, we then propose a method to estimate the reliability of rhythm features extraction. Finally, we propose a two-fold method to automatically analyse metric modulations from audio recordings. On the one hand, we propose a method to detect metrical structure changes from the rhythmogram feature in an unsupervised fashion. On the other hand, we propose a metric modulations taxonomy rooted in music theory that relies on metrical structure descriptors that can be automatically estimated. Bringing these elements together lays the ground for the automatic production of a musicological interpretation of metric modulations.

Acknowledgements

I hate school. I always have and I probably always will. But here I am finalising my PhD thesis after having spent 24 years being a student, which represents 96% of my existence. A psychiatrist would probably diagnose me with some form of masochistic learning insatiability disorder - or perhaps an even more serious condition since I am also scared of heigh but love climbing? Luckily, these questions can be left unanswered. Most importantly, a number of people have made this journey bearable and I intend to give them due credit in the next few lines.

Naturally, my first acknowledgment goes to my supervisors Mark Sandler, Chris Harte and Simon Dixon without whom there would have been no PhD research. A special thank to Chris who kept on helping out and reviewing some of my work even after he had left the Centre for Digital Music half way through my PhD. I must also give a first order acknowledgment to Omnifone Ltd. for partly funding my PhD and for the fantastic people I have had the opportunity to meet and work or interact with. First and foremost, Matt White and Phil Sant who acted as my industry-side supervisors. I must also thank all the other Omnifonians who made it a unique experience: Daniel Escobar, Nako Martinez, Aiko Hara, Becky Brooks, Enrico D'Amelio, Gary Jones, Chris Evans, Oliver Hsu, Masumi Kawamura, Mark Crosbie-Smith, Tom Boswell, Elodie Pereira, Neal Hart, Neil Bates, Damien O'Jeanson, Tiago Esteves, Dom Blatchford, Mark Knight, Ibrahim Saad. My time at Omnifone was a precious learning experience and a prodigious amount of fun. I only wish Omnifone had not gone out of business so soon so that it could have funded my PhD until the end.

Similarly, a vast amount of people have contributed to making my PhD experience worth my while on the Queen Mary side. Cited in no particular order, thank you all for having been fabulous collaborators, colleagues, friends, critiques, travel buddies etc: Giulio Moro, Ken O'Hanlon, Dave Ronan, Dave Moffat, Sebastian Ewert, Mathieu Barthet, Keunwoo Choi, Alo Allik, Bob Sturm, Steve Hargreaves, Holger Kirchhoff, Jordan Smith, Chunyang Song, Yading Song, Geraint Wiggins, Melissa Yeo, Siying Wang, Thomas Vassalo, Florian Thalmann, Adib Mehrabi, Dan Stowell, Luis Figueira, Daniele Barchiesi, Janis Sokolovskis, Michael Terell, Peter Foster, Robert Tubb, Katja Knecht, Katarina Kosta, Thomas Wilmering, Julie Freeman, Johan Pauwels, Yvonne Blokland, Delia Fano Yela, Maria Panteli, Andrew Robertson, Siddarth Siggita, Adan Benito, Saumitra Mishra, Will Wilkinson, Emmanouil Benetos.

If, by a very unfortunate course of events, you happen to be reading this and realise that I have forgotten your name in the above lists, I sincerely apologise and cowardly blame

it on the state of mild lunacy one is in when writing up a PhD thesis. My regrettable omission does not dent my desire to acknowledge your contribution, however.

Last but not least, this section could not possibly be complete without acknowledging all the people outside of my work environment without whom life would not have the same taste. A special award must go to my family who put me on track for this endeavour and have been, for all these years, benevolently putting up with me doing all these things that hardly anyone really understands nor cares about.

This work was supported by the EPSRC award 1325200 and Omnifone Ltd.

Contents

List of Figures

Abbreviations

ACF	**A**uto **C**orrelation **F**unction
BPM	**B**eats **P**er **M**inute
FFT	**F**ast **F**ourier **T**ransform
Hz	**H**ertz (s^{-1})
IOI	**I**nter **O**nset **I**nterval
MIR	**M**usic **I**nformation **R**etrieval
MFCCs	**M**el **F**requency **C**epstral **C**oefficients
NMF	**N**on-negative **M**atrix **F**actorisation
ODF	**O**nset **D**etection **F**unction
s	**S**econds
SSM	**S**elf **S**similarity **M**atrix

Symbols

$s(n)$	Audio signal
f_s	Sampling frequency
$\mathbf{X}$	Audio spectrogram
$\mathbf{x}$	Frame of audio spectrogram
ω	Frequency of a spectrogram bin
m	Spectrogram frequency bin index
n	Spectrogram time frame index
t	Spectrogram frame timestamp
Y	Half-wave rectifier function
L	FFT window length
l	Index of a sample within the window
$\mathcal{W}$	Window function
d	Hop size
φ	Phase
Φ	Onset Detection Function
ν	Audio spectrum weighting function
B	Superflux filterbank
f	Frequency bin in log scale in Superflux
ρ	Width of the Superflux maximum filter
T	Number of triangular filters in Superflux
μ	Spectral flux frame span
ζ	Phase deviation-based onset detection function
Γ	Complex difference
R	Log-likelihood ratio
$\mathcal{M}$	Median filter

k	Median filter centre
ℓ	Median filter length
$\mathbf{p}$	Percussion-enhanced spectrogram frame
$\mathbf{P}$	Percussive spectrogram
$\mathbf{h}$	Harmonic-enhanced spectrogram frame
$\mathbf{H}$	Harmonic spectrogram
$\mathbf{M}$	Median filter mask
$\mathbf{B}$	Self-similarity matrix
b	Self-similarity matrix element
κ	Checkerboard kernel
ξ	Taper parameter
ζ	Foote Novelty curve
$\mathbf{R}$	Rhythmogram
$\hat{\mathbf{r}}$	Periodicity spectrum
I	ACF normalisation factor
κ	Onset envelope shape
u	Metrical level weight
ω	Metrical level pulse rate
λ	Metrical Ratio
Λ	Metrical Ratio sequence
L	Depth of metrical structure
Ш	Dirac comb
F	Metrical hierarchy pulse rates vector
U	Metrical hierarchy Weights vector
$\mathcal{D}$	Distance matrix
$\mathbf{M}$	Matchmatrix
ξ	Tolerance
l	Lag of an autocorrelation function
$\mathcal{K}$	Peak-picking Kernel
$\mathcal{M}$	Metrical hierarchy candidate
E	Effective resonance as defined by McKinney
η	Damping constant in the resonance equation
W	weight given to a tempo candidate

T	Tempo
ST	Relative strength of the two tempi estimates
ρ	Number of frequency bins in log-scaled metergram
TP	True Positives
FP	False Positives
FN	False Negatives
ppr	Pairwise Precision Rate
prr	Pairwise Recall Rate
pfm	Pairwise F-measure
S_o	Over-segmentation score
S_u	Under-segmentation score
S	Entropy
τ	Period of a signal
$\mathbf{W}$	Templates matrix
$\mathbf{H}$	Activations matrix
w	Element of the templates matrix
h	Element of activations matrix
K	Number of templates
D	Reconstruction error
$\mathbf{J}$	Matrix of ones
α	NMF penalty weight
$\boldsymbol{\lambda}$	Relevance vector in L1-ARD Beta-NMF
g	NMF scale factor
$\mathbf{Y}$	NMF Penalty
Υ	Penalty term in NMF cost function
C	K-means Clusters assignment vector
$\mathcal{C}$	K-means Set of observations
$\mathbf{z}$	K-means Cluster centroid
Θ	HMM States transition matrix
$\mathbf{O}$	HMM Vector of observations
o	HMM Observation (=element of the observation vector)
Ψ	HMM States sequence
ψ	HMM Hidden State

$\mathcal{S}$	HMM Hidden State space
π	HMM Emission probability

Chapter 1

Introduction

1.1 Music content analysis

Over the past twenty years, the combination of the democratisation of the internet and the proliferation of music in digital format has brought about a deep and somewhat brutal disruption to the way music may be stored, distributed and consumed. According to the IFPI[1], digital revenues (i.e. streaming and download) overtook physical sales (i.e. CD and Vinyl) in 2015 and represent the growing source of revenue for the recorded music industry. At the end of year 2016, the size of standard commercial music catalogues accessible via streaming and download services exceeded 50 million tracks. Similarly, the quantity of user-generated content shows a rapid growth and largely exceeds the commercial collections. For instance, as of the 2016 statistics, approximately 600 hours of video are uploaded on YouTube every minute[2]. It is clear that not all of this content contains music, but we provide this figure to give the reader a sense of the amount of content that must be looked after on modern platforms. Given the scale, it is clear that the manual organisation, navigation and discovery of such collections has become impractical. The automation of such tasks gives rise to both considerable challenges and opportunities.

[1]The IFPI is a not-for-profit international organisation that represents the interests of the recorded music industry globally and monitors its revenues. http://www.ifpi.org/

[2]Source: personal communication with YouTube employee

In the field of Music Information Retrieval (MIR), research efforts are directed towards the investigation and development of methods for automatically analysing musical content, with the aim of providing new and scalable ways of interacting with musical content [4, 5]. Besides the clear industrial interest they generate, MIR paradigms are also deeply related to more fundamental research questions regarding musicology, cognitive psychology, signal processing, social sciences or natural language processing to name but a few. Music is a largely multi-faceted phenomenon that proves to be an incredibly challenging application for computational methods. As a consequence, MIR is a very active and very diverse field of research, with many more challenges to be tackled [6]. This thesis aims at contributing to the field of MIR by providing a research effort towards the improvement of the computational methods available for the automatic analysis of musical content. Among all the facets of MIR, we put our focus on the rhythmic properties of music and more precisely on metric modulations.

1.2 Rhythm studies in MIR and towards the automatic detection of metric modulations

A sizeable portion of MIR research efforts consist in developing models and methods for the automatic estimation of musical attributes, such as harmony, timbre, melody, rhythm etc. Since this thesis is primarily concerned with the estimation of rhythm-related attributes we will focus on these aspects from now on. We refer the interested reader to relevant literature for studies of other musical attributes, see for instance [5, 7]. Among the various musical attributes considered in MIR research, rhythm has received a substantial amount of attention. A number of rhythm-related tasks are typically investigated. Examples of such tasks are onset detection [1, 8], tempo estimation [9, 10], beat tracking [11, 12], downbeat tracking [13, 14] or metrical structure estimation [15, 16]. We reserve a more detailed review of the body of work relevant to this thesis for Chapter 2.

The term *metric modulation* has previously been used in different contexts. The usage we make of it here might therefore not correspond to that of other authors. For the

purpose of this thesis, let us define a *metric modulation* simply as a *change of metrical structure*. We understand the term *metric modulation* in its most general sense: it applies to any type of metrical change. As such, our definition of this term encompasses definitions such as those given by Fétis or Carter (cf. Section 6.1), which refer to more specific types or families of metrical changes. The estimation of the metrical structure has received some attention from the MIR research community, and we propose an algorithm for estimating metrical attributes in Chapter 4. There exist, however, very few works addressing the automatic detection of metric modulations in the literature. A few latent state space models for estimating the metrical structure that can track changes over time (cf. Chapter 2), i.e. that have the potential to track metric modulations, were published but they are subject to a trade-off between stability of the estimate and sensitivity to metrical changes. Since they are typically used for estimating the metrical structure, they are mostly set for stability, therefore inhibiting their ability to track abrupt changes. Moreover, the architecture of the models, which implements a strong bias regarding the expected musical content and the model parameters are typically set manually or learnt in a supervised fashion. In order to tune the models for the estimation of metric modulations, setting the parameters manually would enforce a very strong bias and learning in a supervised fashion requires a large amount of adequate data. In this thesis, we seek to investigate the automatic detection of metric modulations. Among the vast diversity of possible modulations, we restrict the scope of this work to modulations that consist in an abrupt change from one stable metrical structure to another. Slow and gradual alterations of the metrical structure are therefore not considered. In particular, we consider a blind detection scenario in which no prior knowledge regarding the metrical structure nor metric modulations is assumed. In these circumstances, and given that suitable training data is not available, latent state space models cannot be used for addressing this task. We therefore propose an unsupervised approach to the automatic detection of metric modulations.

Though metric modulations tend not to be used very often by contemporary western popular music composers, they may be encountered a lot more frequently for instance in progressive rock or in classical music compositions of Stravinsky or Monteverdi to name but a few. Irrespectively of their popularity, metric modulations are distinctive musical

features that deeply structure a musical composition. They are therefore susceptible to have a great effect on the listener. As a consequence, the study of metric modulations is of particular interest from a musicological perspective. They may further be exploited as a feature to support the exploration of large music collections. Creative applications involving metric modulations may also be considered. For instance, automatic sequencing or mashup creation systems such as the 'Automashupper' [17] typically assess the rhythmic compatibility of two pieces by performing some form of similarity measure. The transitions generated this way are therefore likely to always have similar musical character. Introducing a method for handling metric modulations would then provide an extension that enables interesting rhythmic effects (i.e. metric modulations) to be produced when sequencing or mashing up pieces.

Both the intrinsic musical relevance, the applications that can be envisioned in today's context and the relatively small amount of existing work motivate our interest for the study of metric modulations, and in particular for going towards their automatic analysis.

1.3 A signal-based approach

Questioning what music *is* is beyond the scope of this thesis. However, it is to be noted that music, and therefore metre, is a construct of the human mind, and it can therefore be argued that is does not exist outside of it [18]. A consequence of this posture is that music can then be related to cognitive science, psychology and neuroscience [19]. Following this rationale, it appears that the perception of music is affected by mental representations [20], training [21], age [22, 23], expectation mechanisms [24] or enculturation [25] to name but a few.

Although there are a variety of processes at play when a human subject is listening to music, the perception of music as the formation of a mental construct is nevertheless strongly conditioned by the acoustic stimulus [26]. In other words, the perception of music is strongly (but not fully) determined by the acoustic phenomenon produced by the musicians and/or captured on a record. As a consequence, we argue that a method

based exclusively on the analysis of the audio stimuli, which may be treated as a signal, can be used to form a first approximation estimate of the human musical perception.

In this thesis we will not consider the psychological and cognitive dimensions of music, but we will rather focus on developing a fully unsupervised approach that only relies on the acoustic stimulus and attempt to push it to its limits. In short, we will take a pure signal-based approach. As such, its relative lack of ability to account for a number of psychological and cognitive variables is counter-balanced by a lighter model that does not require prior knowledge and a resulting wider applicability.

In the remainder of this chapter we summarise the research questions we are concerned with and outline the structure of this thesis.

1.4 Research Questions

RQ1: How can we automatically estimate the metrical structure of music?

This thesis is primarily concerned with the automatic detection of metric modulations, which are characterised by a change of metrical structure over time. Tracking metric modulations thus requires the ability to produce an estimation of the metrical structure, and its evolution over time. As a consequence, it comes that the main objective of this thesis is linked to the question of the automatic estimation of metrical structure of music, which is by no means a solved problem in the current state of the art.

As we detail in chapter 2, we distinguish two categories of approaches to metrical structure estimation: the cycle tracking methods that aim at tracking the length of metrical cycles as well as the location of their start and end locations, and the periodicities estimation methods that characterise the metrical structure via the measure of metrical level pulse rates while disregarding the phase information. The rhythmogram has been employed in the MIR literature for tasks involving the discrimination of different metrical structures. Relating to RQ2, in Chapter 4 we propose and evaluate an algorithm for automatically extracting attributes of the metrical structure from the rhythmogram.

RQ2: To which extent does the rhythmogram capture the metrical structure of music?

A number of prior studies mentioned the ability of the rhythmogram to capture metrical structure. In particular, it has been suggested by a number of authors that metrical level pulse rates correlate with salient peaks of energy in the rhythmogram. But this hypothesis had never been directly evaluated. Moreover, there exist a number of methods to calculate a rhythmogram that have been reported to have different properties for capturing the metrical structure. After providing some background on the rhythmogram feature and its variants in chapter 2, we investigate how the energy distribution in a selection of variations of the rhythmogram relates to the metrical level pulse rates in Chapter 4.

RQ3: What is the impact of human judgment discrepancies on the evaluation of automatic metrical structure extraction algorithms?

The typical approach to evaluating automatic feature extraction algorithms, whether they are applied to music or not, consists in comparing the estimates they produce with a previously established "ground truth". The quality of an algorithm is then measured by how well it reproduces the ground truth. By the very nature of music, there is not always a clear "ground truth" when it comes to estimating musical features (e.g. the metrical structure). In order to evaluate musical feature extraction algorithms, human expert annotations are typically used as a proxy for ground truth.

For many years MIR systems have been evaluated on datasets containing a single annotation per excerpt. In this context the evaluation is merely a quantification of the ability of an algorithm to reproduce the annotator's bias. Since a number of studies have shown that both expert and non-expert humans may disagree when annotating a piece, it became clear that the evaluation produced against a single annotation may not generalise. In a bid to tackle this shortcoming, datasets containing multiple annotations for each excerpt have been created in the recent years for a variety of tasks. The availability of this data has enabled the emergence of studies investigating the (dis)agreement between

expert annotators. A handful of authors have investigated the impact of inter-annotator disagreement on the evaluation of algorithms, notably on the task of music similarity.

In this thesis we seek, in chapter 3, to evaluate whether or not human expert disagree when annotating the metrical structure of music, and if so we seek to understand how they disagree. For instance, is the disagreement related to particular musical attributes? to annotator singularities? Secondly, we investigate in chapter 4 how this disagreement impacts the evaluation of feature extraction algorithms, based on the case study of automatic metrical structure estimation.

RQ4: How can automatic feature extraction failures be predicted?

From the MIR literature, it is clear that the performance of state of the art algorithms reaches different levels depending on the tasks considered. However, despite good and increasing performance, none of these algorithms delivers a perfect performance, which means that cases of failure exist. Numerous metrics are available to quantify a posteriori the respective proportion of successes and failures observed on available datasets. Some algorithms attach a confidence value to each estimate though this practice is not generalised. Therefore, the failures are typically unpredictable, which leaves only two options to the MIR algorithm user: blindly trusting the algorithm and accepting that it will occasionally fail or not using the system at all.

A number of MIR tasks (e.g. cover song detection) or complex systems (e.g. recommendation system) involve processing pipelines in which the later stages rely on features extracted in the earlier stages. In such a context, we believe that having the means to evaluate how reliable the features on which a given processing pipeline relies are would be a valuable asset for MIR research. We view the unpredictable character of this unreliability as one of the main obstacle to MIR algorithms usability. If failures were predictable, the potential MIR user would be given more leeway to handle failures and therefore potentially be able to make use of MIR systems despite their imperfection. To this end, we seek to investigate in this thesis how the reliability of feature extraction systems could be estimated. In particular, we focus on the case of rhythm features extraction in Chapter 5.

RQ5: How can we automatically detect metric modulations?

The automatic detection of metric modulations has seldom been addressed in the field of MIR research. There exist a few latent state space models for estimating the metrical structure that can track changes over time (see for instance [27]). As such, they theoretically have the potential to track metric modulations, but since they are subject to a trade-off between stability of the estimate and sensitivity to metrical changes, they are typically not used to that end (see for instance [28]). In fact they are commonly used for estimating the metrical structure and are consequently set for stability, thereby inhibiting their ability to track abrupt changes. The model parameters are typically set manually or learnt in a supervised fashion. Considering a metric modulation detection scenario, setting the parameters manually would enforce a very strong bias [27] and learning in a supervised fashion would require a large amount of adequate training data, which is currently not available.

In this thesis we consider a blind scenario for the automatic detection of metric modulations, i.e. no prior knowledge regarding the metrical structure nor metric modulations is assumed. Because of this posture and the fact that data suitable for supervised learning is not available, latent state space models appear not to be most appropriate solution for addressing this task. In contrast, we propose an unsupervised approach to the automatic detection of metric modulations.

Relating to RQ1 and RQ2, we show that the rhythmogram enables the capture of attributes of the metrical structure that are altered by metric modulations, namely the metrical level pulse rates. As a result, it shows potential as a feature from which to detect metric modulations. Combining these ideas, we describe an unsupervised approach to detect modulations from the rhythmogram in Chapter 7.

RQ6: Can computational methods be used to automate musicological analyses of metric modulations?

RQ5 is concerned with the detection of metric modulations, which is a challenging problem in itself. However, it is necessary to push the analysis further in order to gain a

significant musicological insight. In particular, once a modulation has been detected it remains to be characterised, for instance by answering questions like: how has the metrical structure been affected by the modulation? What type of modulation is it? So far, in the few examples of studies mentioning automatic detection of metric modulations, the metrical structure classes accessible for the system were manually predefined (cf. [27]), which makes this characterisation relatively straightforward.

In this thesis we aim at a more generalisable approach in which no prior information of the nature of the metrical structures involved is assumed. In this context, how could the metric modulations still be characterised? In order to formalise our approach, the musicological literature provides a body of existing work to get inspiration from. In particular, a taxonomy of metric modulations can characterise each modulation type based on distinctive attributes. However, traditional musicological approaches are typically based on the analysis of a score. The fact that a score is not always available for any musical recording, leads us to ask: how can we transfer the benefits of work carried out in traditional scored-based musicology to the realm of automated computational methods with the aim of producing musicologically meaningful automatic analyses? This question is investigated in more details in Chapter 6.

1.5 Thesis Contributions

The contributions made by this thesis to the field of automatic analysis of rhythmic properties of musical recordings may be summarised as follows:

Datasets

Algorithms for automatic musical analysis are typically evaluated by measuring their ability to reproduce a reference "ground truth". By the very nature of music, it is not always possible to create an objective "ground truth" or "golden standard" for any given musical feature (e.g. chords, tempo etc.). As a consequence, expert annotations are commonly used as a proxy to ground truth for algorithms evaluation. One of the contributions of this thesis is the creation of two new datasets for the evaluation of automatic

metrical structure and metric modulation estimation systems. Our analysis of the inter-annotator agreement on the GTZAN-Met dataset as well as the comparative analysis with the GTZAN-Rhy dataset are reported in Chapter 3 and provide an insight into the properties, qualities and singularities of these annotations, thereby enabling a more robust evaluation of algorithms performance.

Rhythmogram and Metrical Structure

It has been reported in existing literature that the rhythmogram feature (with its multiple variants) captures information related to the metrical structure of music. A number of examples of application of the rhythmogram to classification tasks have indirectly suggested that this assumption may hold, but it had never been evaluated directly. This thesis contributes, in Chapter 4, the first direct evaluation of the ability of the rhythmogram to capture metrical structure related information. We demonstrated that this ability significantly varies depending on which rhythmogram variant is being used and isolate the most efficient one. In addition, we propose and algorithm for estimating metrical level pulse rates from the rhythmogram. Finally, we have shown that taking inter-annotator (dis)agreement into account in the evaluation procedure provides support for more robust conclusions to be drawn.

Feature Extraction Reliability Prediction

The performance of systems for automatically analysing musical recordings has increased significantly during the past decade but occasional estimation failures are still present. However, these systems seldom provide an indication of reliability of the estimates they produce. Unpredictable failures are a major barrier to building trust in an automated system and therefore present a significant obstacle to the adoption and/or usefulness of such a system for scientific research as well as industrial applications. In Chapter 5 we propose a method for predicting the reliability of the automatic production of rhythm related estimates. We demonstrated that it is effective at evaluating the reliability of beat positions, tempo and metrical structure estimation. As a result, this method aims at making automatic musical feature estimation systems trustable and reliably usable on

their own as well as in complex systems despite their imperfection. In a bid to facilitate reliability estimation, we have implemented the method as a Vamp Plug-in.

Metric Modulations Detection and Analysis

The automatic detection of metric modulations is seldom addressed in MIR literature. In fact, there exists models that have the capability of capturing metric modulations, but they are typically not used to that end. Prior to our contribution there was no dedicated dataset for evaluating metric modulation extraction systems. We hypothesise that this may be one of the reasons why this task has not been thoroughly tackled. Moreover, the few examples of existing models capable of capturing metric modulations must be trained in a supervised fashion. In Chapter 7 we propose an unsupervised method for automatically detecting metric modulations and demonstrate state of the art performance on the task of retrieving segments of consistent metrical structure as well as metrical structure change points (i.e. locating the metric modulations). The next step in the analysis of metric modulations consists in characterising their nature. To this end, we propose, in Chapter 6, to use a metric modulation taxonomy, inspired by musicological theory. First, we produced an English translation of an existing metric modulations taxonomy from musicological literature, originally written in French. Secondly, given that a score is not always available, we adapt this taxonomy based on score notation to a formalism that is compatible with features automatically extracted from audio recordings. In particular, we formulate an adapted taxonomy that relies on the metrical level pulse rates to characterise metric modulations.

1.6 Publications

- Peer-reviewed:
  - [29] Elio Quinton, Christopher Harte, and Mark Sandler. “Extraction of Metrical Structure from Music Recordings”. In *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx). Trondheim, Norway*, 2015.

  - [30] Elio Quinton, Mark Sandler, and Simon Dixon, "Estimation of the Reliability of Multiple Rhythm Features Extraction from a Single Descriptor," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 256-260.

  - [31] Elio Quinton, Ken O'Hanlon, Simon Dixon and Mark Sandler, "Tracking Metrical structure Changes with Sparse-NMF" in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017

- Other Contributions:

  - [32] Elio Quinton, Christopher Harte, and Mark Sandler, "Audio Tempo Estimation Using Fusion of Time-Frequency Analyses and Metrical Structure," In *Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2014.

  - [33] Elio Quinton, Ken O'Hanlon, Simon Dixon and Mark Sandler, "Automatic Detection of Metrical Structure Changes" in *Digital Music Research Network 1-Day Workshop (DMRN+11)*, 2016

1.7 Thesis Outline

The remainder of this thesis is organised as follows:

Chapter 2

This chapter reviews existing work in areas of Music Information Retrieval (MIR) related to the tasks considered in this thesis. In addition, computational techniques as well as evaluation metrics used in subsequent chapters are briefly described.

Chapter 3

This chapter briefly presents the datasets used in the work described in this thesis, two of which are new contributions. Their creation and contents is described. In addition, we detail an analysis of the inter-annotator agreement, as permitted by multiple annotations.

Chapter 4

Rhythmograms are one of the common feature used for automatic analysis of rhythm. In this chapter, we propose a quantitative evaluation of the ability of the rhythmogram to capture all metrical level pulse rates present in music recordings and propose an algorithm to extract this information. In addition, we present a simple algorithm to estimate tempo given the metrical structure that was submitted to the audio tempo estimation task at MIREX 2014. The contents of this chapter are made of work reported in [29] and [32] with the addition of further analysis.

Chapter 5

In this chapter, we propose a method to estimate the reliability of several rhythm features extraction. The work described in this chapter was reported in [30].

Chapter 6

This chapter is concerned with the design of a taxonomy of metric modulations. We first produce a brief English translation of an existing metric modulation taxonomy design for score-based analysis taken from musicological literature and originally published in French. We then proposed to take inspiration from it and produce a new taxonomy suitable for analyses based on features automatically extracted from audio recordings. Finally, we used this taxonomy to perform automatic classification of metric modulations.

Chapter 7

This chapter introduces a new method to automatically detect metric modulations from a rhythmogram-like feature. This method is compared to a range of existing and standard methods that we adapted to the task of metric modulation detection. The description and evaluation of our method was partly reported in [31].

Chapter 2

Background

In this chapter we introduce theoretical concepts and mathematical or computational methods that will be used in the rest of this thesis, relate them to existing work, and define notations. Because music is the subject of study in this thesis, it is important to first define some music theory concepts in section 2.1. Given the focus on rhythm, and more specifically on metrical structure, we review the concepts and methods used for automatic analysis of metrical structure of music in sections 2.2 to 2.5. As will be shown later in this thesis, a parallel can be drawn between the automatic detection of metric modulations and the popular task of structural segmentation. As a consequence, we present the relevant underlying concepts and metrics in section 2.10. A number of techniques that were not specifically designed for the study of rhythmic properties of music (e.g. Non-negative Matrix Factorisation) are employed in subsequent chapters of this thesis. We therefore briefly describe them here, motivate their use and introduce relevant notation in sections 2.6 to 2.9.

2.1 Music Theory Concepts

In this section we briefly present some music theory concepts relating to rhythm and more specifically metrical structure. This is motivated by the fact that automatic metrical structure estimation systems necessarily rely on a model of metrical structure of music, whether or not it is explicitly stated. We therefore present the main concepts relevant

in the context of automatic metrical structure analysis and their mutual relationships. Common concepts associations and ambiguities are also highlighted.

2.1.1 Metrical Structure

In many musical cultures, including but not limited to western tradition as suggested by studies on Turkish [34], Indian [35], African [36] or south-American [37, 38] music to only name a few, an underlying sequence of beats and cycles is strongly perceived. Time is typically subdivided in temporal units, the smallest of which is referred to as the *tatum*, which stems from 'temporal quantum' [15, 39, 40]. By grouping these temporal units, or beats, (for instance in patterns of strong and weak beats) a multi layered structure emerges and is known as *meter*.

The Generative Theory of Tonal Music

Lerdahl and Jackendoff proposed a formalised description and definition of the metrical structure in the Generative Theory of Tonal Music (GTTM) [41]. Since the GTTM is useful to define some of the theoretical musical concepts used in this thesis, we reproduce here some definitions formulated by Lerahl and Jackendoff. The GTTM specifies a number rules regarding the formal analysis of metrical structure[1], grouped in three categories:

1. The Well-Formedness Rules (WFR), which specify the requisite properties for structural description. In other words, a structure is not acceptable unless it complies with these rules.

2. The Preference Rules (PR), which steer the choice of a structural description towards one that maximally correlates with expert listeners' formalisation of any piece.

3. The Transformational Rules (TR), which lay out means of relating distorted structures to well-formed descriptions.

[1] Although we only focus here on the metrical structure, we note that the GTTM is applicable to several levels of musical structure, the metrical structure being only one of them.

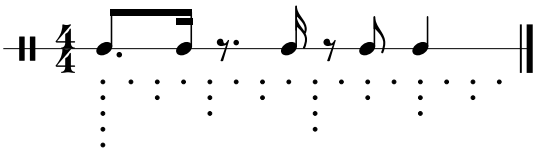


FIGURE 2.1: **A simple rumba clave rhythm pattern and the corresponding metrical structure**. Each horizontal line of dots represents an underlying metrical level.

Although we refer the reader to the original publication for in-depth detailed description, we reproduce here the metrical structure well-formedness rules as they describe the fundamental structuration of the metrical structure:

1. WFR1: "Every attack point must be associated with a beat at the smallest metrical level present at that point in the piece" (pp97)

2. WFR2: "Every beat at a given level must also be a beat at all smaller levels present at that point in that piece" (pp97)

3. WFR3: "At each metrical level, strong beats are spaced either two or three beats apart." (pp97)

4. WFR4: "The tactus and immediately larger metrical levels must consist of beats equally spaced throughout the piece. At subtactus metrical levels, weak beats must be equally spaced between the surrounding strong beats." (pp97)

With the GTTM, Lerdahl and Jackendoff lay out a formal description of the metrical structure as a multilayered structure that follows hierarchical constraints. Figure 2.1 illustrates the hierarchical structure of metrical levels for the rumba clave rhythm. The dots below the stave represent the beats corresponding to each metrical level. Note that this arrangement is compliant with the well-formedness rules. For each layer, we refer to the corresponding beat rate as a *metrical level pulse rate* in the remainder of this thesis.

Beat and Downbeat

Multiple meanings have been given to the term *beat*. Lerdahl and Jackendoff use the word *beat* to describe the pulses corresponding to each metrical level (e.g. WFR2). On

the other hand, it is also common to use the term *beat* to refer to the pulse of the metrical level to which listeners synchronise when foot-tapping, nodding or perhaps dancing along the music, which is known as the *tactus*. In such a scenario, beat and tactus are identified. Note that in this case it is common the refer to '*the* beat' instead of simply 'beat'. The MIR research community adopts the latter naming convention and the task of 'beat tracking' consists in retrieving the positions of *the* beat, i.e. tactus.

Identifying the beat to the tactus implies that the term 'beat' cannot be used to describe the pulse of other metrical levels (as opposed to Lerdahl and Jackendoff convention). The term 'downbeat' refers to a metrical level of longer period than the beat, which is typically aligned with the beginning of a rhythm cycle and/or with harmonic changes. In score notation the downbeat commonly corresponds to the beginning of a bar. Note that although the terminology is different, this definition is not incompatible with the GTTM (e.g. the downbeat is one of the metrical levels). The concepts of beat and downbeat effectively load some metrical levels with a distinctive semantic meaning.

The relation between beat and downbeat (e.g. the ratio of their pulse rates) is therefore informative of the metrical structure in spite of the fact that it does not fully describe it. On this ground, and this will be described further in section 2.4, numerous authors in the MIR community identify meter tracking with joint beat and downbeat tracking.

Hierarchical structure

An important aspect of metrical structure is its hierarchical organisation. For instance, in the GTTM, this property is encoded in the well-formedness rules 1 and 2. The hierarchical relationships between metrical levels are then a fundamental descriptor of the metrical structure. Figure 2.2 shows the association between some common terms used to describe metrical structures and the hierarchical relations they imply. These terms define relations between the metrical level immediately above and below the tactus (or the beat) level respectively. Then, a metrical structure is labeled either as duple or triple if the metrical level above the tactus corresponds to the grouping of two or three beats respectively. Similarly, a metrical structure is labeled either as simple or compound if the beat is subdivided in two or three equal parts respectively. The terms *binary* and

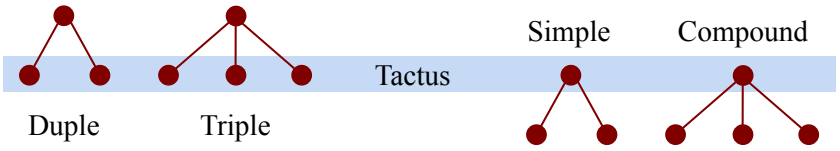


FIGURE 2.2: **Common metrical hierarchy terminology.** Left to right, these labels describe the hierarchical relations between the metrical level immediately above and below the tactus level respectively. In both cases, the terms describe a subdivision into two or three equal parts from one metrical level to the next.

ternary are also used to described a subdivision in two or three equals parts respectively. Combinations of these terms may then be used to describe metrical structures e.g. 'simple duple' for describing a structure that would be easily scored as $\frac{2}{4}$, or 'triple compound' for describing a structure that would be easily scored as $\frac{9}{8}$. Note that this description is only concerned with the metrical levels immediately neighbouring the tactus level and are not informative nor restrictive of the hierarchical relationships to further levels. For instance, a compound metrical structure could include a lower level that is either a binary or ternary subdivision of the level immediately below the beat (see Figure 4.3 for an example of full metrical structure representation). In addition, this nomenclature only covers beats grouped and subdivided in a constant number of units, e.g. in compound metrical structure all beats are subdivided in three equal parts. The term *odd meter* is used to describe structures in which beats are subdivided or grouped in more than one number of units. Figure 4.4 provides an example of such a structure in which a $\frac{5}{8}$ bar is made of groupings of two and three eighth notes. Because the interval between two pulses is then not constant, this type of metrical structure can be described as *non-isochronous*.

2.1.2 Note Values and Time Signature

The concept of note values is used to represent the relative duration of sounded musical events and rests. A sounded musical event may for instance be a note or any other sound object and *rests* are understood here as any silent event [42]. In the western notation system the *whole note* is the note value of longest duration[2]. All other note values are obtained by successively subdividing the whole note in two equal parts, as illustrated on Figure 2.3. As a consequence, a whole note is subdivided in two *half notes*, each of which is then subdivided in two *quarter notes* and so on. Note that in the American

[2]Although it can be extended using dots and ties

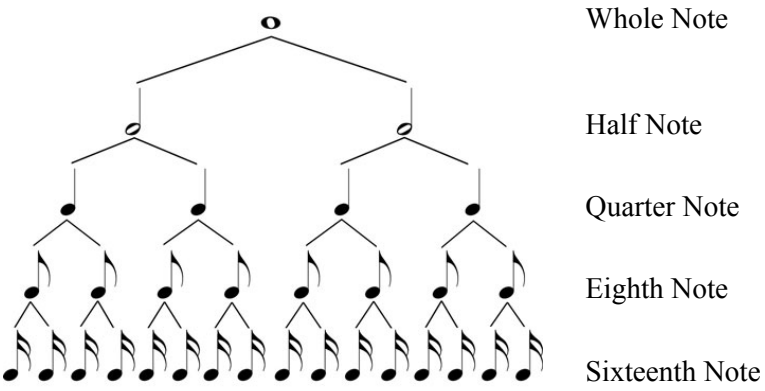


FIGURE 2.3: **Basic note values and their labels**

terminology for note values used here, each note value label describes how many of these need to be added up to equal the duration of a whole note (i.e. 8 eighth notes in a whole note). The British terminology provides equally functional, but not as self-explanatory labels.

In the score notation system, the time signature is presented using two numbers. The denominator specifies the time subdivision unit, or note value, to be counted, and the numerator specifies how many of these units constitute a bar. Note that the number used at the denominator to identify the note value corresponds to the naming system given in Figure 2.3. A half note is identified with number 2 because two half notes add up to a whole note, the quarter note is identified with number 4 as four quarter notes add up to a whole note and so on. For instance $\frac{2}{4}$ time signature means that there are 2 quarter notes per bar and $\frac{6}{8}$ time signature means that there are 6 eighth notes per bar.

The time signature is often identified to the meter of a piece, so that a number of automatic metrical structure analysis systems aim to retrieve the time signature, see for instance [43]. This direct association of time signature and metrical structure is subject to a few nuances, however. By convention, the time signature suggests a typical specification of a portion of the metrical structure. For example, $\frac{3}{4}$ time signature a priori implies a simple triple meter. In other words, there are three beats in a bar and each beat is subdivided in two equal parts (cf. Figure 2.2). On the other hand $\frac{6}{8}$ a priori implies a compound duple meter. In other words, there are two beats per bar and each beat is subdivided in three equal parts. We note at this point the importance of time signature

interpretation convention on this example: $\frac{3}{4}$ and $\frac{6}{8}$ are mathematically equivalent in that there are the same number of units in each case (3 quarter notes equates to 6 eighth notes), but not metrically equivalent given the convention. Moreover, note that the note value specified at the denominator of the time signature does not necessarily specify the note value associated to the metrical level intended to be the beat (i.e. tactus). In $\frac{3}{4}$, the quarter note is typically assumed to correspond to the beat, but the eighth note is not in $\frac{6}{8}$. In addition, the score notation for a given piece of music is not unique. Therefore, given that the score is a cue for the performers, the composer or arranger is responsible for choosing the notation that will result in a performance suitable to his intention. There exist more than one notation convention depending on era or genre. For instance it is common practice in Jazz to use $\frac{4}{4}$ time signature, which traditionally implies a simple duple meter, to score compound meter pieces. Jazz performers are used to interpret the score appropriately.

Finally, the time signature does not specify the entirety of the metrical structure. It specifies a typical hierarchical structure around the beat (i.e. tactus) level, but for instance does not specify lower metrical levels. Besides, the time signature does not imply definitive constraints on the metrical structure. Accidentals may be used, and therefore imply a metrical structure that is different from the typical structure associated to the current metrical structure. Similarly, composers are free to choose to notate music as they wish, and it is not rare to come across compound meter pieces (or sections thereof) scored in $\frac{4}{4}$ with use of eighth notes triplets although $\frac{12}{8}$ would be a naturally suitable time signature. To conclude, it is undeniable that there typically is a correlation between the metrical structure of a piece and the time signature used in a score but there is no strict equivalence between these two concepts. This motivates our choice to characterise the metrical structure using metrical levels and their hierarchical organisation rather than time signatures.

2.1.3 Tempo

The notion of tempo was historically introduced to describe the pace of music, i.e. how fast it feels. On musical scores, tempo was indicated using words that evoke the intended

feeling, such as *presto* meaning 'very fast'. Often, the words used as tempo indications have a connotation that goes beyond just the pace and also suggest a certain attitude to be given to the music, e.g. *largamente* that means 'slow and dignified'. The interpretation of these words in terms of speed and expressivity of execution was the conductors' or performers' responsibility. As a result, their choices may not have matched the intent of the composer. With the invention of the metronome, there has been a way for composers to specify the intended speed of execution more objectively and more reliably. The metronomic indication, typically given in beats per minute (BPM), then indicates the pulse rate of a given metrical level. This is typically specified on a score by marking which note value is associated to the metronomic indication.

Since then, the original notion of tempo and the metronomic indication tend to be confused. However, these concepts are not rigorously equivalent. The original notion of tempo describes the pace of the music; in other words the perceptual effect that the piece is intended to have on a listener. On the other hand, the metronomic indication only provides a technical point of reference for the pulse rate of a given metrical level. The metronomic indication is commonly (but not necessarily) associated with the pulse rate of a metrical level labelled as the *beat*. In perceptual terms this is often associated with the *tactus rate*, which is the pulse rate at which people would tap or nod along to the music. The tactus rate is then implicitly associated to the metronomic indication, which is itself identified with the tempo. However, it has been shown that the tactus rate alone is not sufficient to characterise the pace of music [44]. In addition, it has also been shown that different people can latch on a tactus rate that correspond to different metrical level pulse rates [21, 45]. In the context of MIR research, the notion of tempo is understood as equivalent with the tactus rate, and measured with a rate in BPM similar to a metronomic indication. As a result, the large field of research labelled as automatic tempo estimation is effectively performing tactus estimation (cf. section 2.5). The estimation of the pace of music, however, has received far less attention; see for example [46–48].

2.2 Onset detection

The task of detecting the time instants at which musical events occur, known as '*onset detection*', is a necessary first step for a variety MIR tasks such as automatic transcription, and generally all rhythm related features extraction such as beat tracking, tempo estimation and metrical structure estimation. Relatively comprehensive reviews of the underlying principles of onset detection as well as of the variety of methods proposed in the literature[3] are given in publications such as [1, 49]. The intent in this chapter is not to provide an exhaustive description of the onset detection body of work but rather to present the fundamentals principles of onset detection, briefly review the families of onset detection functions calculation methods and highlight what aspects are relevant to the work presented in this thesis.

2.2.1 General paradigm of onset detection

Music, as an acoustic signal, may be viewed as constituted of a number of organised events that unfold in time. Conforming to this view, the musical event may then be seen as the atomic constituent of music. A musical event could typically be a note, although not limited to it. For instance a musical event could also be characterised by variations of timbre. The investigation of what are the cognitive processes at play to enable the human listener to isolate musical events is beyond the scope of this work, but we note that in the context of audio-based MIR research the notion of musical event implies the existence of a change of some nature measurable in the signal over time to isolate musical events [1]. The temporal evolution of a sonic (musical) event is canonically represented in several phases: attack, decay sustain and release, as illustrated in Figure 2.4. The notion of onset intends to describe the time instant at which a musical event starts. The term *transient* is also commonly used when describing temporal evolution of sonic events. It is not easy to define, and should not be confused with the attack, however. A transient is commonly defined as a short time interval "*during which the signal evolves quickly in some nontrivial or relatively unpredictable way*" [1] whereas the attack is the time interval during which the amplitude envelope increases. Thus the initial transient

[3] at the exception of the more recent neural networks methods

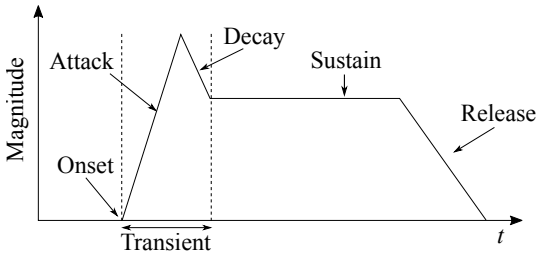


FIGURE 2.4: Schematic magnitude envelope of a note onset, attack, decay, sustain and release.

may include the decay phase, and the release may also be considered as a transient. The task of onset detection then consists in detecting the onset of an event, that is to say the the time at which its occurence starts. Typically, an onset will be located during a transient (if there is one). A critical time resolution for the location of onsets is given by the fact that two transients are perceived as two separate event only if they are more than 10ms apart [50].

The general scheme of onset detection algorithms comprises three steps, for which we use here the nomenclature introduced by Bello *et. al.* in [1]: (optional) *Pre-processing, Reduction* and *Peak-picking*. The scheme is illustrated in Figure 2.5 and the constituent steps are described below.

- **Pre-processing** may be applied to the raw audio signal with the aim of improving the performance of the subsequent stages. Examples of such pre-processing could consist in analysing the signal independently in several frequency bands [51–54]. These frequency bands can for instance be chosen to approximately reflect human hearing filter distribution [55].

- **Reduction** is the process that aims at deriving from the audio signal a function in which the occurence of musical events is clearly manifested. Such a function is referred to as the *Onset Detection Function* (ODF). Onsets are expected to be revealed as peaks in this function. The ODF is typically sampled at a lower rate than the audio sampling rate by several orders of magnitude. Given that the smallest perceivable time interval necessary for the human ear to resolve two separate events is about 10 ms, ODF sampling rate is typically of the order of 200Hz. The computation of the ODF is a key element of onset detection. The main

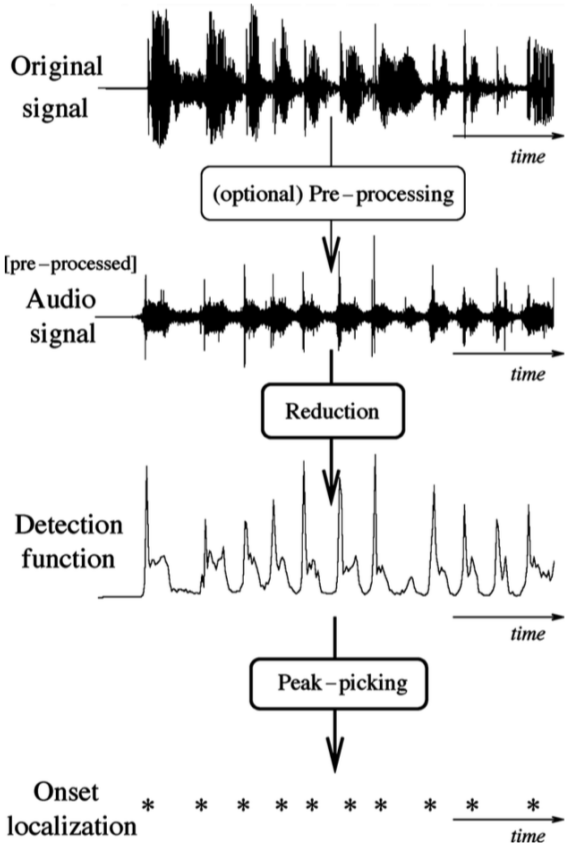


FIGURE 2.5: Onset detection scheme. Figure reproduced from [1]

approaches to the computation of onset detection functions are briefly described in section 2.2.2.

- **Peak-picking** is the final step in the onset detection scheme. The goal of onset detection is to retrieve the time instants at which onsets occur. The onset locations are estimated by peak-picking the ODF in which they are expected to materialise as peaks. Retrieving discrete onset locations is beyond the scope of this work, so it will not be discussed here. We refer the interested reader to relevant literature, e.g. [1, 49].

2.2.2 Onset Detection Functions

A vast number of methods have been proposed to compute onset detection. We do not intend to offer here an exhaustive record of all the methods that have been proposed.

Instead, we present here the main strategies that underpin the computation onset detection functions reported in literature. We group them in four categories: the temporal features, which operate directly on the audio waveform, the spectral features, which operate on the magnitude spectrogram, the spectral features with phase, which exploit the phase information of the spectrogram and finally the probabilistic models, which exploit statistical models of the audio or spectral signal.

Temporal features

As a first approximation, it appears that onset occurrences coincide with transients in the audio signal. On this premise, early onset detection methods relied on detection functions that aim at capturing onsets from the amplitude envelope of the audio signal [56–58]. In its simplest form a detection function could be computed by rectifying and smoothing the audio signal with $\mathcal{W}$ is a window, or smoothing kernel, of length L centred around 0:

$$\Phi_E(n) = \frac{1}{L} \sum_{i=-\frac{L}{2}}^{\frac{L}{2}-1} |s(n+i)|\, \mathcal{W}(i) \tag{2.1}$$

where $s(n)$ denotes the n^{th} sample of the audio signal. A variation of this consists in using the energy instead of the rectified signal:

$$\Phi_{E_2}(n) = \frac{1}{L} \sum_{i=-\frac{L}{2}}^{\frac{L}{2}-1} [s(n+i)]^2\, \mathcal{W}(i) \tag{2.2}$$

Given that the target of this type of methods is to capture rapidly increasing amplitude envelopes, a possible refinement consists in computing the derivative of the rectified signal or of the local energy, i.e. $d\Phi_E/dn$ [52]. Taking in account that the loudness is perceived logarithmically [59], a further refinement consists in computing $d\log(\Phi_E)/dn$ [53]. Such temporal features are effective for clean signals with strong transients (typically stemming from percussive sounds) but do not give satisfactory results for other type of signals.

Spectral Features

Methods using the spectral representation of the audio signal to compute the detection function were later proposed. They typically produce more robust onset detection functions than temporal features, which tend not to be used anymore in modern systems.

In the spectral domain, transients result in sort-term broadband energy bursts. In musical signals, most of the energy is located at low frequencies so that broadband events are easier to capture at high frequencies [60]. Let $\mathbf{X}(m,n)$ be the short-term Fourier transform (STFT) spectrogram of the audio signal, obtained using a L samples window where m is the frequency bin index and n is the frame index. The spectrum may then be re-weighted with a frequency-dependent weighting function $\nu(m)$ to give more importance to high frequency content (and therefore enhancing transients):

$$\Phi_{HFC}(n) = \frac{1}{L} \sum_{m=-\frac{L}{2}}^{\frac{L}{2}-1} \nu(m) \left|\mathbf{X}(m,n)\right|^2 \tag{2.3}$$

If $\nu(m) = 1$, $\nu(m) \left|\mathbf{X}(m,n)\right|^2$ is the local energy. Masri propose to weight each bin proportionally to its frequency $\nu(m) = |m|$, thereby resulting in a *high frequency content* (HFC) detection function [61]. By construction, this is geared towards music with strong transients.

A more general approach consists in measuring the distance between successive magnitude spectrogram frames. Onset detections calculated on this basis are labelled *spectral difference* or *spectral flux* depending on authors. A number of variations of the spectral flux have been proposed, with various pre-processing and metric employed to compute the difference. Using the L_1 norm of the difference between successive frames, the spectral flux is given by [61]:

$$\Phi_{SF}(n) = \sum_{m=1}^{\frac{L}{2}} Y(|\mathbf{X}(m,n)| - |\mathbf{X}(m,n-1)|) \tag{2.4}$$

where L is the window length used in the processing of the Fourier spectrogram, $\mathbf{X}(m,n)$ is the spectrogram coefficient of the m^{th} bin of the n^{th} frame, and Y is the half-wave

rectifier function:

$$Y(x) = \frac{x + |x|}{2} \tag{2.5}$$

Half-wave rectification is introduced to account only for energy increases in successive frequency bins, in order to capture onsets rather than offsets. Duxbury used a variation of the spectral flux based on the L_2 norm of the rectified difference [54].

$$\Phi_{SF_2}(n) = \sum_{m=1}^{\frac{L}{2}} \left[Y\left(|\mathbf{X}(m,n)| - |\mathbf{X}(m,n-1)| \right) \right]^2 \tag{2.6}$$

Spectral flux type methods are known for their robustness in capturing onsets from polyphonic recordings, even in the absence of very sharp onsets. The onset detection function $SF(n)$ effectively is the sum of (possibly squared) difference $|\mathbf{X}(m,n)| - |\mathbf{X}(m,n-1)|$ for all frequency bins m. Because this difference is computed frequency bin wise it captures transitions between pitched notes because the fondamental frequency and harmonics, i.e. the distribution of energy on the frequency axis, change from one note to another. Naturally, the presence of a strong transient, i.e. of a short term broadband energy burst, also results in a large difference because it can be expected that the energy of a large number of frequency bins will increase during the transient. In other words, the spectral flux is sensitive to transients *and* harmonic changes.

In the basic formulation of equations (2.4) and (2.6), the difference is computed between adjacent frames. However, depending on the hop size chosen for the processing of the audio spectrogram this might or might not provide enough context for an optimally robust computation: if the hop size is very small, the overlap between windows is then very large, which results in small differences between successive frames. The spectral flux calculation can then be extended to compute the distance between frames that are μ frames apart:

$$\Phi_{SF_3}(n) = \sum_{m=1}^{\frac{L}{2}} Y(|\mathbf{X}(m,n)| - |\mathbf{X}(m,n-\mu)|) \tag{2.7}$$

If $\mu = 1$, equation (2.7) is equivalent to (2.4). When $\mu > 1$, the overlap between the frames considered for the calculation is smaller, which yields larger differences. Note

that the L_2 norm formulation of (2.6) can be generalised to μ frames apart calculation too.

The spectral flux is computed between frequency bin m and the same frequency bin μ frames apart. This means that deviations in frequency that displace energy maxima from one frequency bin to another over time would be registered as a large value in $\Phi_{SF}(n)$. In practice, small frequency deviations such as vibrato or pitch instability would result in spurious peaks in the onset detection curve. Böck introduced a further improvement on spectral flux, labelled *superflux*, by using a maximum filter to implement robustness against local frequency deviations (e.g. vibrato) [62]. First, a pre-processing stage is applied so that the magnitude spectrogram is filtered using a bank of 138 triangular filters logarithmically distributed, aligned with the western scale and with central frequency separated one quarter tone from each other; thereby covering the $[27.5, 16000]$ Hz frequency range:

$$\mathbf{X}_{log,filt}(f,n) = \log_{10}\left(|\mathbf{X}(m,n)| \cdot B(m,f) + 1\right) \tag{2.8}$$

where $\mathbf{X}_{log,filt}(f,n)$ is the filtered spectrogram, $B(m,f)$ the filter bank and f the index of the current filter. Robustness against local frequency deviations is implemented using a maximum filter. For each frequency bin in the scaled and filtered spectrogram $\mathbf{X}_{log,filt}(f,n)$, the energy is set to the maximum value in the neighbourhood (along frequency axis):

$$\mathbf{X}^{max}_{log,filt}(f,n) = \max\left(\mathbf{X}_{log,filt}(f-\rho : f+\rho, n)\right) \tag{2.9}$$

where ρ is the width of the maximum filter, set to $\rho = 1$ in [62]. Finally, the superflux detection function $\Phi_{SF^*}(n)$ is computed with respect to the maximum filtered spectrogram, in a similar fashion to the spectral flux:

$$\Phi_{SF^*}(n) = \sum_{f=1}^{T} Y\left(\mathbf{X}_{log,filt}(f,n) - \mathbf{X}^{max}_{log,filt}(f,n-\mu)\right) \tag{2.10}$$

Foote introduced a measure of audio novelty related to spectral difference methods in [63].

First, the distance between feature vector frames (e.g. power spectrogram) is computed, and a distance matrix is formed[4]. A "*novety curve*" is then obtained by correlating a Gaussian tapered checkerboard kernel along a diagonal of the distance matrix. The novelty curve shows sharp peaks at time instants corresponding to large differences. The kernel size controls the time span over which distances are integrated to compute the measure of novelty. Therefore, when the kernel is small (i.e. of a length in the order of 0.5s), the novelty curve effectively corresponds to an onset detection function. On the other hand, when the kernel size is chosen larger (i.e. length in the order of seconds), the peaks in the novelty curve are expected to signal structural segment boundaries. In fact, this method is seldom used for onset detection, but mostly finds applications in the task of structural segmentation. It is therefore described in greater detail in section 2.10.

Spectral Features with Phase

All the spectral methods described above only exploit the magnitude spectrogram. We present here methods making use of the phase information of the complex spectrogram. The Fourier transform computes the projection of a given signal on the basis of sinusoidal functions, for which the phase advance is related to the instantaneous frequency:

$$\omega(m,n) = \left(\frac{\varphi(m,n) - \varphi(m,n-1)}{2\pi d} \right) \omega_s \tag{2.11}$$

where $\varphi(m,n)$ is the 2π unwrapped phase of the m^{th} frequency bin at the n^{th} time frame, d the hop size and ω_s the sampling frequency. For a stationary sinusoid, the instantaneous frequency is expected to be approximately constant, which implies from (2.11) that the phase advance should be constant too.

$$\varphi(m,n) - \varphi(m,n-1) \approx \varphi(m,n-1) - \varphi(m,n-2) \tag{2.12}$$

[4]The euclidian distance as well as cosine distance are suggested by Foote as possible distance measures

This could equally be characterised by the second order phase difference being approximately equal to zero in stationary parts (from (2.12)):

$$\Delta\varphi(m,n) \triangleq \varphi(m,n) - 2\varphi(m,n-1) + \varphi(m,n-2) \approx 0 \tag{2.13}$$

In spectrogram frames corresponding to transients, the input signal is not well approximated by a stationary sinusoid, hence the instantaneous frequency not being well defined and therefore $\Delta\varphi(m,n)$ tends to be larger than zero in magnitude. As a result, it is expected that $\Delta\varphi(m,n)$ would peak around transients locations and close to zero in steady state regions. Since $\Delta\varphi(m,n)$ is defined for each frequency bin, a simple way to construct an onset detection function consists in computing the mean absolute phase deviation [64]:

$$\Phi_p(n) = \frac{1}{M}\sum_{m=1}^{M} |\Delta\varphi(m,n)| \tag{2.14}$$

where M is the number of frequency bins in the spectrogram. Bello proposed a slightly more intricate method to compute an onset detection from phase difference in [65]. It consists in treating the M deviations $\Delta\varphi(m,n)$ as a probability distribution for each frame. In stationary phases of the signal, deviations tend to be zero, and therefore the distribution strongly peaks around this value. Reciprocally, during transients, the phase deviations tend to be larger, but not necessarily consistent in magnitude, which means that the distribution is then flattened. An onset detection is then obtained by computing the inter-quartile range and the kurtosis of the distribution, i.e. measuring how flat it is. By only relying on the phase information, these methods are equally sensitive to the phase advance of components with no significant energy (e.g. silence), which tend to be noisy. This effect can be minimised by combining phase and amplitude information [64].

Integrating further the notion of combination of phase and magnitude, Bello *et al.* propose a method to compute the onset detection directly in the complex domain [66]. The difference between the observed complex value of spectrogram bin $\mathbf{X}(m,n)$ and the value that can be predicted by phase advance from the previous frame $\hat{\mathbf{X}}(m,n)$ is calculated:

$$\Gamma(m,n) = \left[|\hat{\mathbf{X}}(m,n)|^2 + |\mathbf{X}(m,n)|^2 - 2|\hat{\mathbf{X}}(m,n)||\mathbf{X}(m,n)|\cos\left(\Delta\varphi(m,n)\right)\right]^{\frac{1}{2}} \tag{2.15}$$

The distances are then summed along the frequency axis to form the onset detection function:

$$\Phi_c(n) = \sum_{m=1}^{M} \Gamma(m, n) \tag{2.16}$$

Since this can be viewed as a measure of the stationarity of the signal, here again the underlying working principle is that onsets are associated to a non-stationarity of the signal and therefore yield large differences that materialise as peaks in the detection function.

Probabilistic and Machine Learning Methods

Another approach consists in describing the music signal with a probabilistic model so that probabilistic and statistical methods for onset detection can be constructed. Here again the underlying working principle consists in estimating the onset locations via the estimation of the likely times of abrupt change in the signal. One class of approaches consists in considering two statistical models $\mathcal{A}$ and $\mathcal{B}$. Each sample of the signal $s(n)$ is assumed to derive from either $\mathcal{A}$ or $\mathcal{B}$. The log-likelihood ratio of the models is then defined as

$$R = \log \frac{p_{\mathcal{B}}(s)}{p_{\mathcal{A}}(s)} \tag{2.17}$$

where $p_{\mathcal{A}}(s)$ and $p_{\mathcal{B}}(s)$ are the probability density functions of the two models. Assuming that the signal follows one model and then switches to the other, the log-likelihood ratio R will change sign. In the case of music signals, the probabilistic signal models are generally unknown and are therefore typically estimated from the data. Assuming that appropriate models can be learnt from the observed signal, this change of polarity can be used to detect onsets, see for example Jehan [67] and Thornburg and Gouyon [68]. An alternative approach consists in using a single probabilistic model and detecting instants at which the signal does not follow the model, i.e. '*surprising*' moments. In this context, a model of the signal can be built from the observed signal so that predictions of its evolution can be formulated. It is then expected that onsets will emerge as events which make prediction of the evolution of the signal after the onset difficult, i.e. surprising events [69].

In the recent years the use of a range of variations on Deep Neural Networks (DNN) has emerged as a new trend for onset detection. Neural networks were first employed to perform onset detection by peak-picking a hand-crafted onset detection function [70]. Lacost and Eck then proposed to learn the onset detector directly from spectral data [71]. In other words, the neural network learns how to compute an onset detection function. Eyben [72] introduced the use of Recurrent Neural Networks (RNN) trained on Mel-scaled magnitude spectrogram, later improved by Böck [73]. Inspired by the analogy between edge detection in image processing and the task of detecting onsets from spectral representations of the audio signal, Schlüter subsequently employed Convolutional Neural Networks (CNN) for onset detection [74, 75]. Recent comparison of onset detection methods revealed that current state of the art performance is achieved by neural networks methods [72, 75].

Interestingly, after introspection of their CNN, Schlüter notes that the network seems to learn to detect percussive (i.e. wide band and short term energy burst) and harmonic (i.e. change of harmonic energy distribution) onsets [75]. That is to say the network learns to detect features that closely ressemble what hand-crafted spectral flux based methods are designed to detect. Schlüter then argues that the superiority of CNN over hand-crafted functions resides in the fact that the network learns hundreds of variations of these basic detectors, which would be impossible to replicate manually.

2.2.3 Difficulties in onset detection

From the descriptions provided in section 2.2.2, it becomes apparent that different methods for calculation of onset detection functions rely on different properties or assumptions about the audio signal. It is then to be expected that performance would depend on the audio signal to which these methods are applied. This intuition is confirmed by numerous authors reporting different performance when applying the same onset detection algorithms to different datasets (with different audio characteristics) [1, 49, 62].

Very quick and drastic change, such as a percussive sound, are typically referred to as '*hard*' onsets, while onsets resulting in slow and/or small change in the audio are commonly referred to as '*soft*' onsets. Given that the notion of onset is associated with

the presence of a rapid change in the audio signal, the detection of soft onsets is more difficult than that of hard onsets. In fact, the detection of 'soft' onsets remains an open research question. Since the calculation of an onset detection function is the first step of the vast majority of rhythm feature extraction and processing methods, soft onsets are problematic for most of rhythm related MIR tasks. Chapter 5 is concerned with the estimation of the reliability of rhythm feature extraction methods, and soft onset constitute one of the sources of failures that will be considered.

2.2.4 Onset detector choice motivation

Comparing a number of handcrafted onset detection functions, it was reported by numerous authors that temporal features and high frequency content typically perform best on percussive sounds while methods like phase deviation perform well for pitched sounds (including non percussive). In comparison, the spectral flux method appears to be a good all rounder, exhibiting the most stable performance over a variety of types of audio signals, see for example [1]. Superflux brings an improvement in performance over spectral flux, while preserving the all-rounder quality [62].

In addition, Tian performed an evaluation combinations of pairs of a variety of standard and then state-of-the-art handcrafted onset detectors (i.e. not based on machine-learning methods) in [76]. The parameters space was exhaustively explored and results for the best performing configurations are reported. The constituent onset detection methods were also evaluated individually for comparison. The outcome of this study is two-fold. On the one hand it was shown that combining onset detectors instead of using the constituent methods led to an increase in performance. On the other hand, it appeared that the superflux method is the best performing method, and is also a constituent method of the top 6 composite methods. Moreover, it appeared that composite methods that do not comprise superflux exhibit lower performance than superflux alone. This result suggests that superflux is providing a robust and performant onset detection function.

Competitive, if not state-of-the-art results have been reported for onset detection functions employing a probabilistic framework, such as log-likelihood and neural network

based methods [1, 72, 75]. The gain in performance in comparison to methods like superflux comes at the price of increased computational complexity, and perhaps more importantly is dependant on the fitness of the probabilistic model to the audio signal. In practical terms, it means that a large quantity of (annotated) training data is required to estimate a suitable model.

In the remainder of this thesis, we chose not to use deep learning based methods in order to limit the complexity of the overall system. Note that this choice is also in line with the bottom-up unsupervised approach take here. In the light of the results discussed in this section, the superflux method was chosen to generate an onset detection function that is used for further processing. The symbol $\Phi(n)$ is used to denote the onset detection function in the remainder of this thesis.

2.3 Rhythmogram

This section is dedicated to the rhythmogram, which is a useful feature for the automatic analysis of metrical structure. First, a definition of the rhythmogram feature as well as of the related terminology is given. In a second part, the main rhythmogram calculation methods are presented. Lastly, an introduction to the interpretation of the rhythmogram with respect to musical concepts is provided.

2.3.1 Definitions and Terminology

Let us define the *rhythmogram* as a two dimensional frequency domain representation of the onset detection function, by analogy with the *spectrogram* that is the frequency domain transform of the audio signal[5]. As such, the horizontal and vertical axes respectively represent time and frequency. By construction, the frequencies of the rhythmogram effectively represent the periodicity rates of the onset detection function. In the context of rhythm studies, it is natural to consider frequencies in the rhythm range, that is to say approximately in the [0.2,13] Hz range [19]. Beats per minute (BPM) is more suitable

[5]Note that although this definition is limited to computations based on onset detection functions, this approach is also applicable to features derived from discrete onset positions, such as Inter-Onset Intervals (IOI) [77]

a unit to measure frequencies in the rhythm range, and is proportional to Hz with 1Hz = 60BPM. A number of authors have used the term *tempogram* to refer to this type of feature, see for instance [78–80], while some other prefer the term *rhythmogram*, see for example [81, 82]. Since they describe the same feature, the two terms are effectively interchangeable. However, it will become apparent in the remainder of this thesis that the rhythmogram captures more than strictly tempo information. On that account the term rhythmogram will be preferred over tempogram from now on. The symbol **R** will be used to denote the rhythmogram in equations.

Just like the frames of the spectrogram are spectra of segments of the audio signal (delimited by the windows), each frame of a rhythmogram is a *rhythm periodicity spectrum* of a segment of onset detection function. Periodicity spectra, in the broad sense of the term, i.e. a spectrum representing the periodicities present in the onset detection function, are widely used in the literature. Again, the terminology used depends on the authors and could for instance be *beat spectrum* [83], *spectral rhythm pattern* [84] or *beat histogram* [85]. For conciseness, in the remainder of this thesis, the rhythm periodicity spectra (i.e. the rhythmogram frames) will be referred to simply as 'periodicity spectra', and notated with symbol **r**.

2.3.2 Calculation Methods

Given an onset detection function, several methods have been proposed to compute a rhythm periodicity spectrum. We group the different strategies proposed in the literature in two categories. On the one hand, methods estimating periodicities by measuring similarity between two instances of the onset detection function shifted by a given time lag are grouped in the the 'Self-Similarity Lag' category. On the other hand, the 'Frequency Domain' methods project the onset detection function of a basis of vectors covering an appropriate frequency range.

Self-Similarity Lag

The most popular approach consists in calculating the autocorrelation of the onset detection function for a range of lags corresponding to rhythm periodicities. After an early implementation using this method on an onset detection function generated from symbolic data by Brown [86], it has widely been applied to onset detection functions derived from audio [43, 85, 87–92]:

$$\mathbf{r}_A(l) = \frac{\sum_{n=1}^{L} \hat{\Phi}(n)\hat{\Phi}(n-l)}{I} \tag{2.18}$$

where $\hat{\Phi}(n) \triangleq \Phi(n)\mathcal{W}(n)$ is the windowed onset detection function, $\mathcal{W}(n)$ a window of length L, l is the lag and I is a normalisation factor. In its simplest form, the autocorrelation is obtained with $I = 1$. It is however common to normalise it against the first element ($I = \hat{\Phi}(1)$) or to use the unbiased autocorrelation, i.e. $I = L + 1 - l$. The computational complexity of such a calculation is N^2. The Wiener-Khinchin theorem allows a more efficient computation of the autocorrelation with a computational complexity in the order of $N\log(N)$, using the FFT of the raw signal (the ODF here):

$$\begin{aligned} \Omega(\omega) &= FFT\{\Phi(t)\} \\ \mathbf{r}_A(l) &= iFFT\{\Omega(\omega)\Omega^*(\omega)\} \end{aligned} \tag{2.19}$$

In this thesis, we compute the ACF using the Wiener-Khinchin theorem. Note that since the first step consists in computing the FFT of the ODF, the computation of the ACF is related to the FFT-based computation described in (2.20).

Foote and Uchihashi propose to estimate periodicities from "self similarity" of the features derived from the audio signal [83]. This is achieved by first building a similarity matrix and then compute sums of the matrix diagonal elements. The distance from any diagonal to the main diagonal effectively represents a time lag. In that sense, the summation over the l^{th} diagonal is analogous to the computation of an autocorrelation with lag l.

Frequency Domain

The Fourier transform is also used to analyse periodicities in the onset detection function [43, 84, 93]. The periodicity spectrum is then straightforwardly obtained by computing the Fourier transform (FT) of the windowed onset detection function:

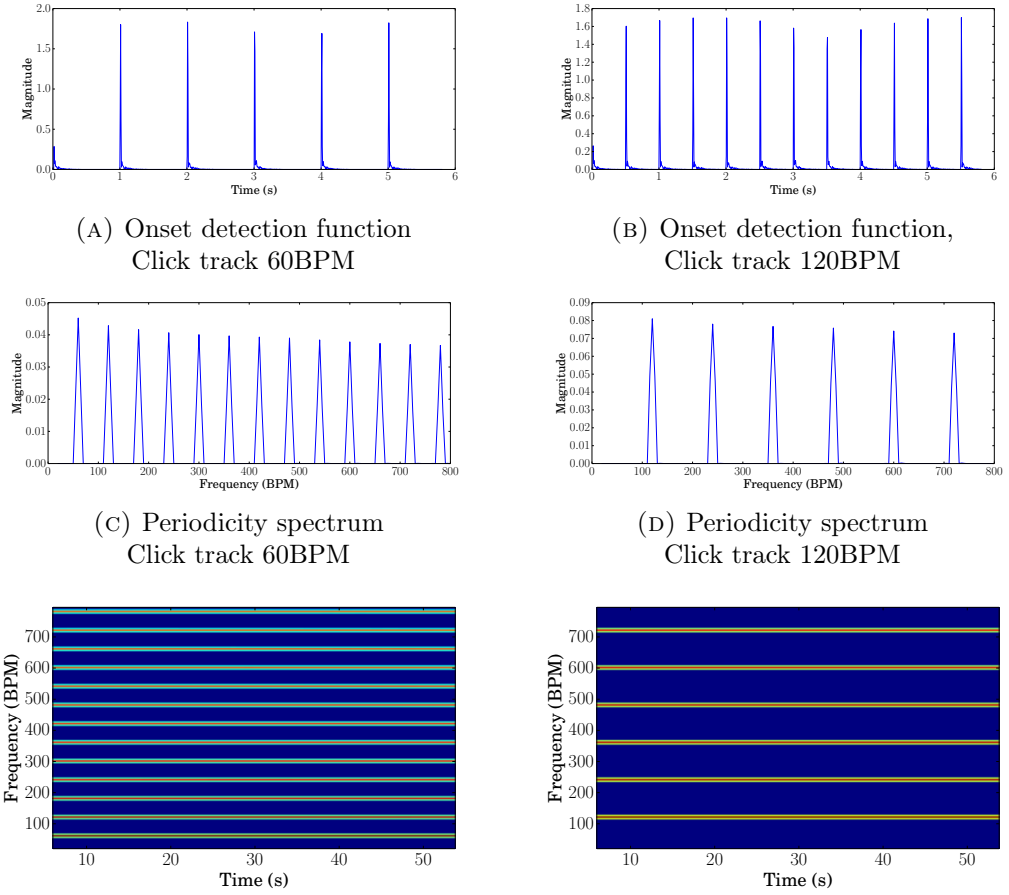
$$\mathbf{r}_F(m) = \mathrm{FT}\left\{\Phi(n)\mathcal{W}(n)\right\} \tag{2.20}$$

where m denotes the frequency bin index.

Finally, a number of authors have proposed to use resonator filter banks to estimate periodicities in the onset detection function [52, 89]. Each bin of the periodicity spectrum then represents the magnitude response of a filter of the corresponding resonance frequency. While standard filter banks, such as comb filters [52], have traditionally been used, Large developed a model of interconnected neural oscillators, named Gradient Frequency Neural Networks (GFNN) with the intent of modelling human auditory system [94]. The architecture he proposed is effectively a single layer filter bank, and should therefore not be confused with deep neural networks. This model was later extended by Lambert [95]. So far GFNN have only been applied to simple elementary signals (e.g. click track or son clave pattern) and are yet to be tested with realistic polyphonic music recordings.

2.3.3 Interpreting a Rhythmogram

The intent in this section is to give the reader a sense of how a rhythmogram can be interpreted. Examples of gradual complexity are shown to expose fundamental properties of rhythmograms as well as highlighting some differences between the two classes of computation methods. First, two computer-generated elementary signals are considered. They consist of a percussive (cross-stick) sound repeated at a regular interval of 1s and 0.5s respectively. For conciseness, we refer to these as 'click tracks'. Figure 2.6 shows a 6s excerpt of the superflux onset detection, the Fourier periodicity spectrum corresponding to the first rhythmogram frame as well as the Fourier rhythmogram for each track. The Fourier transform of a periodic function of period τ is also periodic in the frequency



(A) Onset detection function
Click track 60BPM

(B) Onset detection function,
Click track 120BPM

(C) Periodicity spectrum
Click track 60BPM

(D) Periodicity spectrum
Click track 120BPM

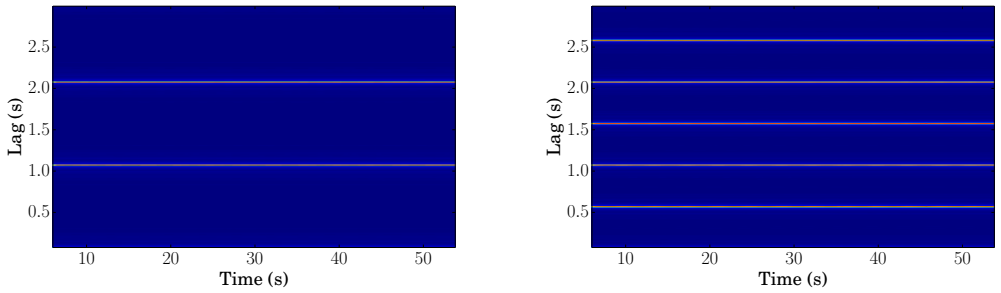
(E) Fourier Rhythmogram
Click track 60BPM

(F) Fourier Rhythmogram
Click track 120BPM

FIGURE 2.6: Example rhythmograms of cowbell sounds.

domain with period $1/\tau$. This property is clearly observed on Figure 2.6 (A) and (C) and on Figure 2.6 (B) and (D) respectively, that illustrate the characteristic behaviour of the Fourier transform of a train of impulses. The period of the audio signal being stable for its entire duration, the periodicity spectra constituting each rhythmogram frame are approximately identical. As a consequence, a clear structure of horizontal lines is observed in the rhythmograms, where each line corresponds to a peak in the periodicity spectrum. In the case of the 60BPM click track, a line in the rhythmogram is observed at 60BPM. As series of lines are also observed at integer multiples of this frequency. A similar observation can be made for the case of the 120BPM click track, where all the harmonics of 120BPM rate are observed.

Similarly, the autocorrelation of a periodic function is also periodic. The main difference with the Fourier transform being that the autocorrelation function peaks at multiples of



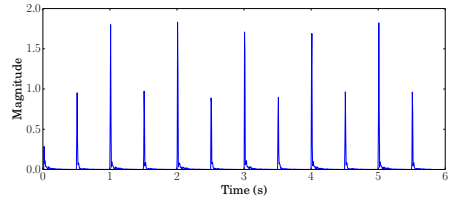
(A) ACF rhythmogram of a 60BPM (i.e. $\tau = 1.0$s) click track

(B) ACF rhythmogram of a 120BPM (i.e. $\tau = 0.5$s) click track

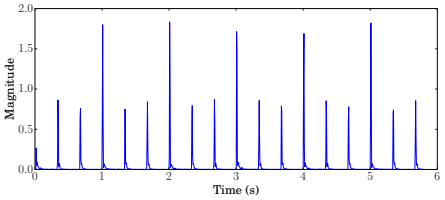
FIGURE 2.7: ACF rhythmograms of two percussive audio signals

the period lag. Figure 2.7 illustrate this on the same two audio tracks as Figure 2.6, which consist of a percussive sound repeated at a period of 1.0s (60BPM) and 0.5s (120BPM) respectively. Figure 2.7 (A) exhibits a horizontal line at 1.0s lag, corresponding to the periodicity of the audio signal as well as a second line at 2.0s lag, corresponding to first multiple of the audio signal periodicity. In a similar fashion, a horizontal line at lag 0.5s corresponding to the periodicity of the audio signal as well as lines at all the multiples of this periodicity are observed on Figure 2.7 (B). Note that periodicity multiples can equivalently be described as frequency sub-multiples, or sub-harmonics. It is then interesting to point out that the Fourier transform of a periodic signal includes harmonics of the signal frequency (i.e. inverse of its periodicity), while the ACF of the same signal includes sub-harmonics. As originally suggested by Peeters, these complementary properties offer potential to estimate periodicities of the signal [84]. This aspect will be addressed in more details in Chapter 4.

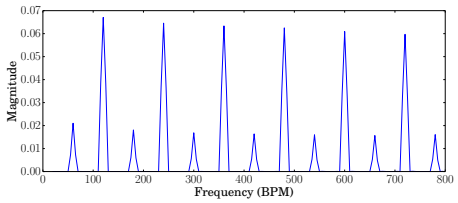
A number of authors have observed that the distribution of energy in rhythmograms of musical recordings is related to the metrical structure of music, see for example [43, 83, 84]. The hierarchical organisation of the metrical structure of music implies that there exists a sense of periodicity at a number of levels in parallel, and these periodicities are typically related to the metrical level pulse rates [41]. Let us now consider two other synthetic audio signals to illustrate how the rhythmogram may reveal the metrical structure. Both tracks use a single percussive sound (the same as before) played at two loudness levels, controlled by the MIDI velocity parameter: 'loud' (velocity = 127) and 'quiet' (velocity = 60). The first track is composed of the repetition of the sequence



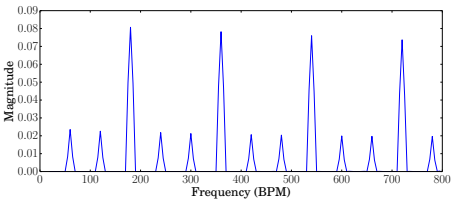
(A) Spectrogram of a 2Hz percussive beat with every second note accented



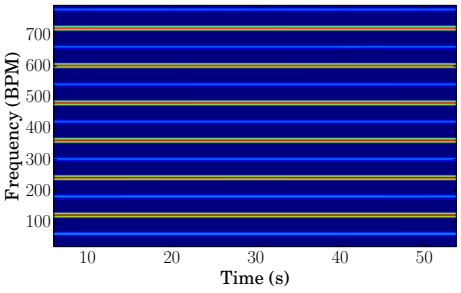
(B) Spectrogram of a 3Hz percussive beat with every third note accented



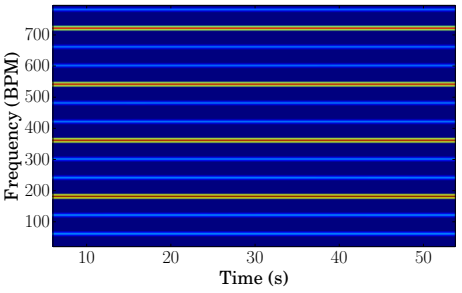
(C) Spectrogram of a 2Hz percussive beat with every second note accented



(D) Spectrogram of a 3Hz percussive beat with every third note accented



(E) Spectrogram of a 2Hz percussive beat with every second note accented



(F) Spectrogram of a 3Hz percussive beat with every third note accented

FIGURE 2.8: Spectrogram of a 2Hz percussive beat with every second note accented

{loud, quiet}. The interval between each onset is 0.5s, so that the period of the sequence is 1s. Similarly, the second track is composed of the repetition of the sequence {loud, quiet, quiet}. The interval between each onset is 0.5s, so that the period of the sequence is 1.5s. Because each track has two periodicities, with different relative periodicities, they stand for elementary examples of two different metrical structures. Figure 2.8 (A) and (B) show a 6s excerpt of the corresponding onset detection functions, where the two velocity levels of the audio signal translate in two magnitude levels. Similarly the periodicity spectra, and therefore rhythmograms shown in Figure 2.8 (C) to (F) exhibit two levels of peaks and lines respectively, thereby reflecting the two levels of periodicity. The distribution of high and low peaks suggests that the distribution of energy in the rhythmogram reveals the hierarchical relationships between metrical level pulse rates.

If the relationship between metrical structure and energy distribution in the rhythmogram seems to be reasonably straightforward on elementary synthetic examples, the question of the generalisation of this relationship to full musical recordings arises. Since the metrical structure of a musical piece is typically made of 4 or 5 metrical levels, it can be expected that the distribution of energy in the rhythmogram would be more complex than in the elementary examples presented so far. Figure 2.9 shows the rhythmogram of an excerpt of a commercial pop song (Lady Gaga's 'Do What You Want'). The distribution of energy is indeed more complex than the synthetic elementary examples previously shown. However, it is still possible to discern patterns of distribution of energy at different rates, which suggests that the metrical structure of the music still influences the distribution of energy in the rhythmogram in complex musical mixtures. Moreover, the energy distribution in the rhythmogram changes around the 80s timestamp. This song is in simple duple meter (typically scored in 4/4) but contains some metrical structure changes. Considering the beat rate of 97BPM, the section ending around 80s relies heavily on sixteenth notes subdivision, whose rate is then 388BPM (=97×4). This subdivision is dropped in the section starting at 80s, so that eighth notes are the shortest subdivision used. This change appears to be captured by the rhythmogram as the energy around 388BPM and around 776BPM significantly decreases around the change. As a result, this observation suggests that the rhythmogram is able to capture metrical structure changes over time, i.e. metric modulations. This question will be explored in greater details in Chapter 7. On the other hand, this example also demonstrate the difficulty to explicitly extract the metrical level rates from the rhythmogram. Although the sixteenth note subdivision is not used anymore after the change, there is still energy around 388BPM. This could be explained as being a harmonic of the eighth note rate (194BPM), as well as of the quarter note (97BPM) and so on. As a consequence, this suggests that identifying metrical level pulse rates from the rhythmogram is not straightforward. Exploiting the complementarity of self similarity lag and frequency domain methods pointed earlier in this section, Chapter 4 is concerned with addressing the questions of the relationship between the energy distribution in the rhythmogram and the metrical level rates and of the determination of the metrical level rates from the rhythmogram.

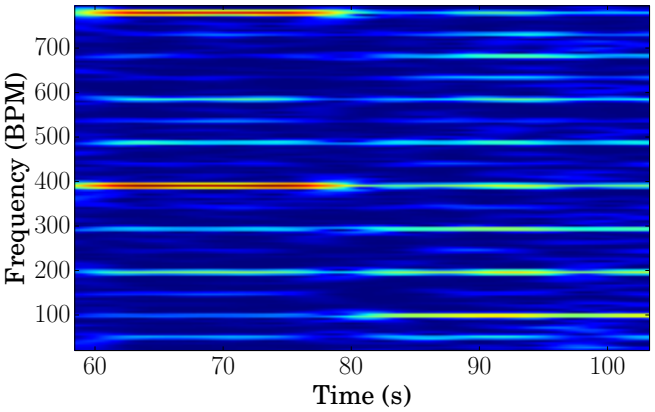


FIGURE 2.9: Fourier rhythmogram of an excerpt of Lady Gaga's 'Do What You Want'

2.4 Metrical Structure Estimation

Automatic extraction of the metrical structure of music from audio recordings is a complex and challenging task, which is by no means a solved problem in the current state of the art. In this section we briefly describe work related to automatic metrical structure extraction. Note that the methods presented here all rely on music theoretical concepts presented in section 2.1.

2.4.1 Canonical Metrical Structure Estimation Pipeline

The canonical approach is a two-stages process. The first stage consists in estimating the positions of onsets of musical events from the audio, either in a discrete (onset locations) or continuous (via an onset detection function) fashion. This problem is an active field of research on its own, we refer to section 2.2 for a brief overview. The second stage focuses on estimating the metrical structure from the onset locations representation derived at the first stage, see for example [15, 28, 43, 96]. In order to account for the fact that the perception of metrical structure is affected by a range of different musical features (e.g. harmony, melodic contour etc.) [26], methods using several onset detection function-like curves derived from a number of features (e.g. timbral or harmonic features) have been proposed, see for example [97–99]. This effectively corresponds to using several variants of the first stage of processing in parallel. The adoption of deep learning methods to address MIR tasks has been significant over the last few years and

is still steadily progressing. Although deep learning has been employed to address the first and second stages independently so far, an emerging trend in recent deep learning work is the development of "end-to-end" systems, which apply deep learning directly on audio, or on the audio spectrum, and output the high level estimate to solve the task under consideration [100, 101]. As a consequence, it may be foreseen that "end-to-end" deep learning methods for metrical structure estimation may be proposed in the close future, therefore departing from this canonical two-stages process. In this section we only discuss the metrical structure estimation step, i.e. the second stage.

2.4.2 Metrical Structure Extraction Strategies

The formal description of the metrical structure (as proposed for instance in the GTTM) implies the existence of periodicities in the structure of music. Each metrical level operates at a different period length (cf. Figure 2.1). As a result, the automatic estimation of metrical structure typically involves a form of estimation of these periodicities and/or their hierarchical relationships. Metrical structure estimation strategies can broadly be classified in two categories. On the one hand, the metrical structure may be characterised by the estimation of the temporal location of metrical cycles. On the other hand, the metrical structure may be characterised by the analysis of metrical periodicities and their mutual relationships.

Cycle Tracking Methods

Works addressing the task of metrical structure estimation by tracking cycles typically focus on two to three metrical levels, i.e. two to three cycles, among the downbeat, tactus (or beat, cf. section 2.1) and tatum. All these approaches consist in tracking the positions of the pulses for each metrical level tracked. In this scenario, the description of the metrical structure therefore equates to the identification of the pulses of two to three metrical levels. The hierarchical organisation of the metrical structure is implicitly encoded in the problem definition: it is assumed that the downbeat cycle is made of a number of beat pulses, which are themselves made of a number of tatum pulses. The estimation of beat and downbeat constitutes the majority of works, see for example

[37, 99, 102]. The *tatum*, which stems from 'temporal atom' and represents the metrical level with the shortest period present in the music, is sometimes considered as a third metrical level of interest [15, 39, 40].

Probabilistic state space models have become the standard approach to address this type of problem, see for example [15, 16, 27, 103, 104]. In this framework, metrical level pulses are modelled via latent variables and the audio signal, or features thereof (typically an onset detection function) are the observations. The task of metrical structure inference then consists in estimating the most probable latent state sequence given the observations. Hidden Markov Models are one form of state space models that may be used of metrical structure estimation, e.g. [15]. The bar pointer model proposed by Whieteley et al. includes a further variation of this class of approaches [27]. In particular, on top of latent variables representing the current position in the metrical cycle (the bar), called the *pointer*, and the speed at which the pointer progresses within the bar, called *instantaneous tempo*, this model includes a rhythmic pattern variable that represents probable locations of onsets. As a result this last variable is conceptually similar to the rhythm patterns considered by Dixon in [105] and implicitly incorporates information about the metrical structure. Further improvements on this model include the optimisation of efficiency by designing a more effective state space [96] or usage of particle filters to handle high dimensionality state-spaces, which is not possible using exact inference (e.g. via HMM) [28].

Although these concepts are formulated in western music terms, the notion of tracking metrical cycles of different periodicities generalises to other musical cultures [35]. The notions of beat, downbeat and tatum must then be transposed to the relevant units for the music considered, but the fundamental principle remains identical. For instance, Srinivasamurthy applied such an analysis framework on carnatic music, tracking the *sama* and *aksara* in order to characterise the tala cycle [80]. In Indian music tradition, the *aksara* is the smallest time unit of the cycle, so in that respect is analogous to the *tatum*. The *sama* is "the first aksara" of the cycle, that is to say the starting point of the cycle, which is analogous to the *downbeat*. One difficulty arising from such transposition is that the length of the relevant cycles might be dependant on the musical culture

considered and potentially exceed the period length range for which such systems were originally designed. In the case of indian music, the longest cycle can exceed the typical length of downbeat cycle (in western music) by an order of magnitude. For this reason, Srinivasamurthy proposed an extension of the bar pointer model to track long metrical cycles [106], of durations up to a minute (when a bar in western music is typically a few seconds long). However, such long cycles are typically associated with the structural segmentation of a piece rather than metrical cycles and given the perceptive thresholds observed in psychological experiments, it is questionable whether such long cycles can be perceived as metrical cycles [19]. In fact, Srinivasamurthy named this extended model the "Section Pointer Model", which reveals his awareness of the issue.

Periodicities Estimation Methods

Another approach to metrical structure analysis consists in analysing metrical periodicities and their mutual relationships. In this scenario, the temporal information (i.e. the temporal position of the beats, downbeats etc.) is not considered, but it is easier to track a large number of periodicities (i.e. more than 2 or 3) than with a cycle tracking method.

The first step of processing is the calculation of a feature that reveals the metrical periodicities present in the music, such as the *beat spectrum* [83], *inter-onset histograms* [92], or *periodicity spectra* [107]. We refer to section 2.3 for a more detailed description of these features and their computation. However, we recall that these feature share a common property: periodicity rates are captured as peaks. The raw periodicity features may then be normalised so that the peaks are identified with respect to a given rate, for instance the tactus rate (or tempo). This type of approach was adopted for instance by Peeters [93] or Robine who uses this representation to build *meter class profiles* as vectors of thirteen dimensions representing the relative strength of pulses at rates related in a fixed set of integer ratios to the tempo (which is required as prior knowledge) [43]. Further analysis can then be performed using these features, normalised or not.

A range of metrical structure related tasks have been tackled using such features. One trend is concerned with rhythm-based classification [84, 92]. It is to be expected that the

metrical structure of music has an influence on such a classification, but the classification task in itself does not explicitly reveal the metrical structure. Gouyon proposed a more direct estimation of metrical structure properties by performing a classification based on a dichotomy between duple and triple meter [90]. Robine used the meter class profiles to classify musical excerpts with respect to time signature tags [43]. The explicit extraction of the metrical structure — i.e. associating periodicities to metrical level pulse rates — is comparatively rarely addressed, see for example [108]. Authors often observe that some peaks in the periodicity features correspond to metrical level pulse rates, and use these features for a variety of tasks on the basis of this assumption. A formal evaluation of this assumption has not been carried out, however. This observation forms one of the motivations for the study carried out in Chapter 4. It is likely that the lack of available data for carrying such a study is, at least partly, responsible for this. The metrical structure annotation dataset introduced in Chapter 3 enables such an evaluation.

Metric Modulations

In the vast majority of the works referenced above, changes of metrical structure over time (i.e. metric modulations) are not considered. Often the metrical structure is assumed constant over the duration of the piece, or excerpt under scrutiny. There exist, however, a few works addressing the issue of abrupt changes of metrical structure [109, 110]. Latent state space models are theoretically capable of tracking metric modulations. This capability is however dependant on architecture design, and more importantly on model parameter settings. For instance, the transition probabilities are often set so that they implement robustness against octave jumps [15]. This type of setting also implements the inability, or at least greatly limits the ability to track metric modulations. As a result of these design choices, metric modulations tracking capability is compromised. Nevertheless, it is to be noted, that Witheley et al. demonstrated that the bar pointer model can track metric modulations [111]. However, their evaluation is only carried out on two symbolic data examples, which is not sufficient evidence to assess the general robustness of the system. In particular, it may be hypothesised that the gain in sensitivity to metrical changes may lead to a large number of spurious (metrical change) detection

in sections of consistent meter. Automatic tracking of metric modulations has otherwise seldom been addressed.

Besides, latent state space models used in metrical cycles tracking systems, require a number of parameters to be set in order to function, such as the acceptable tempo range, the length of a metrical cycle or the rhythm patterns. It is clear that these parameters are critical for the description of the metrical structure. They are typically set manually or learnt from data in a supervised fashion. This implies that the range of musical variety covered is highly dependant on either the training data or the manual inputs. A limited performance on music which properties differ from the training data is to be expected. Therefore, a large amount of training data is required in order to derive robust models. On the other hand, section 2.3 suggests that the rhythmogram may be capable of capturing metrical structure changes, and its processing does not require prior knowledge. As a result of all these observations we propose in Chapter 7 a method to detect metric modulations from audio recordings in an unsupervised fashion, on the basis of a rhythmogram feature.

2.5 Tempo estimation

The estimation of tempo is not an objective of this thesis. Nevertheless, because it is related to some aspects considered here (cf. Chapter 5), we provide a brief overview of the paradigm of tempo estimation, related work and challenges. Note that this brief overview is limited to estimation of tempo from audio recordings, i.e. estimation of tempo from symbolic representations of music is not considered here.

2.5.1 Tempo Estimation Paradigm

In the context of audio-based MIR research, the tempo is understood as the tactus rate (cf. section 2.1.1). As a consequence the task of tempo estimation consists in extracting the rate at which listeners tap along a musical audio recording. Because it relates to human perception, this definition of tempo is also often referred to as *perceived* tempo, by opposition to the *notated* tempo, which is the metronomic indication on a score. Given

this definition, tempo estimation algorithms are evaluated against annotations of the *perceived* tempo, which are typically collected by asking a listener (or a group thereof) to tap along to the music. The tempo annotation is then derived as the rate of tapping. Algorithms performance is evaluated against the annotated value. In a typical scenario, the tempo is regarded as correct if it lies within a tolerance window around the annotated value, e.g. 8% of the annotated value in the MIREX audio tempo estimation task[6].

2.5.2 Related work

The first step of a tempo estimation algorithm is the computation of an onset detection function. One strategy consists in extracting discrete onset positions by peak-picking the ODF. The time interval between each pair of onset position then defines an inter-onset interval (IOI) [39, 112–115]. Informations about the tempo may then be extracted from appropriate[7] IOI histograms [112]. This range of methods relies on the explicit determination of onset positions, which is an error-prone step.

An alternative to the explicit estimation of discrete onset positions consists in using the ODF directly. A tempo estimate may then be derived by the analysis of periodicities in the ODF. Periodicities may be revealed by the computation of a periodicity spectrum or rhythmogram, as already described in section 2.3. In particular, the Fourier transform [116–119], ACF [93, 118, 120–122], and bank of filters [15, 52] have been used to capture periodicities in the context of tempo estimation. As shown in section 2.3, periodicity spectra obtained with these methods tend to exhibit several peaks, i.e. reveal several periodicities. The final step of tempo estimation consists in picking the periodicity corresponding to the tempo. The various methods proposed are typically differentiated by the approach taken to address this critical decision step. The periodicity selection often relies on the assumption that the strongest peak in the periodicity spectrum corresponds to the tempo. Retrieving the tempo can then be performed by picking the rate of the strongest peak. However, this assumption does not always hold; so that a peak that does not correspond to the tempo may be the strongest. This leads to a common difficulty

[6] http://www.music-ir.org/mirex/wiki/2016:Audio_Tempo_Estimation

[7] A limit to the maximum IOI is generally set, so that only a range of intervals meaningful for rhythm analysis is considered

in tempo estimation, known as the *octave error* because the erroneous tempo estimate is usually related to the annotated tempo by a factor of two (or three if compound or triple meter). A number of approaches have been proposed to implement robustness against octave error, such as incorporating a metrical structure informed prior [121] or transposing methods for pitch estimation to tempo estimation [89] to name but a few.

Since the tempo is defined as the tactus rate, tempo estimation is closely related to the task of *beat tracking*, which consists in retrieving the beat positions from audio recordings. Beat tracking is typically performed by tracking the beat period (i.e. tempo) and the beat alignment (also know as beat *phase*). Some systems estimate the beat period and alignment separately [88, 112, 121], while some other estimate them jointly [15, 52]. In other words beat trackers also perform tempo estimation whereas tempo estimation system do not necessarily perform beat tracking (because they do not estimate the beat alignment).

Following a similar trend as a number of other MIR tasks, tempo tracking systems have started using deep learning [9, 123, 124]. Böck introduced a deep-learning method that consistently outperforms previous state of the art methods in [9]. This publication also offers a compact summary of performance of various methods on 10 different datasets. It clearly illustrates the gain in performance, but also that before the introduction of this method, deep learning methods were competitive with method based on hand-crafted features, such as [15, 121, 125, 126], but not significantly more performant. In this context, the method proposed by Davies in [121] exhibits competitive performance and its Vamp Plugin implementation[8] was used in Chapter 5 and Chapter 7.

2.6 Non-Negative Matrix Factorisation

Non-Negative Matrix Factorisation (NMF) provides a framework to approximate a 2-dimensional non-negative matrix $\mathbf{R} \in \mathbb{R}_{\geq 0}^{M \times N}$ by the product of two non-negative matrices

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

$\mathbf{W} \in \mathbb{R}_{\geq 0}^{M \times K}$ and $\mathbf{H} \in \mathbb{R}_{\geq 0}^{K \times N}$ [127] such that:

$$\mathbf{R} \approx \mathbf{WH} \tag{2.21}$$

In this context, the columns of $\mathbf{W}$ are commonly referred to as *template* or *dictionary* vectors while the rows of $\mathbf{H}$ as the corresponding *activations* and K is the number of templates used. K is typically chosen so that $MK + KN \ll MN$, thereby reducing the dimensionality of the data. In the context of this thesis, NMF is applied to the rhythmogram, hence the choice of symbol $\mathbf{R}$. However, the NMF framework can be applied to any 2-dimensional matrix. It has consequently been applied in a wide variety of domains such as financial data analysis [128], image classification [129] or bioinformatics [130]. In the audio and musical domain, NMF has become a standard technique for source separation, see for example [131, 132] and polyphonic transcription, see for example [133–136] in which cases the matrix $\mathbf{R}$ to be approximated is typically a spectrogram representation of the audio signal.

Typically, NMF is performed by assigning a cost function to the difference between the original and reconstructed matrices, $\mathbf{R}$ and $\mathbf{WH}$ respectively, which is reduced using gradient-based optimisation with respect to $\mathbf{W}$ and $\mathbf{H}$ [127]:

$$\min_{\mathbf{W},\mathbf{H} \geq 0} D(\mathbf{R}|\mathbf{WH}) \tag{2.22}$$

where D is typically the Euclidian distance or a variant of the Kullback-Liebler (KL) divergence, defined for two matrices $\mathbf{A}$ and $\mathbf{B}$ as:

$$D_{KL}(\mathbf{A}|\mathbf{B}) = \sum_{i,j} \left(\mathbf{A}_{i,j} \log \frac{\mathbf{A}_{i,j}}{\mathbf{B}_{i,j}} - \mathbf{A}_{i,j} + \mathbf{B}_{i,j} \right) \tag{2.23}$$

While enforcing the non-negativity constraints usually requires rather complex optimisation algorithms, Lee and Seung proposed multiplicative update rules derived from a gradient descent method; the main advantage of these rules being that they are easy to implement [137]. The update rules, using the KL divergence, are formulated as follows:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T(\frac{\mathbf{R}}{\mathbf{WH}})}{\mathbf{W}^T\mathbf{J}} \tag{2.24}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{(\frac{\mathbf{R}}{\mathbf{WH}})\mathbf{H}^T}{\mathbf{J}\mathbf{H}^T} \tag{2.25}$$

Where $\mathbf{J} \in \mathbb{R}_{\geq 0}^{M \times N}$ is a matrix of ones, $\odot$ denotes the Hadamard product (element-wise multiplication) and the division is understood element-wise.

The non-negativity constraint has been shown to be useful to make matrix factorisation learn templates that represent parts of the data [127]. For example, in the context of monophonic transcription, it is to be expected that the templates learnt by NMF would relate to the spectral representation of individual notes [133–136]. This fundamental property explains the popularity of NMF over other matrix factorisation techniques when semantically interpretable templates are desirable. For instance, techniques such as Principal Component Analysis (PCA) or Vector Quantisation (VQ), which are also forms of matrix factorisation [138], do not enforce non-negativity constraints. As a result, coefficients cancellations allow factorisations in which the templates do not represent parts of the data.

The generalised β-divergence offers a generalised closed form for a family of divergences parametrised by β. It includes Euclidean distance ($\beta = 2$), Kullback-Leibler (KL) and Itakuro-Saito (IS) divergences as limit cases as $\beta \rightarrow \{1, 0\}$, respectively :

$$D_\beta(x|y) = \begin{cases} \frac{x^\beta}{\beta(\beta-1)} + \frac{y^\beta}{\beta} - \frac{xy^{\beta-1}}{\beta-1} & \beta \in \mathbb{R}\backslash\{0,1\} \\ x\log\left(\frac{x}{y}\right) - x + y & \beta = 1 \\ \frac{x}{y} - \log\left(\frac{x}{y}\right) - 1 & \beta = 0 \end{cases} \tag{2.26}$$

The β-divergence was proposed for use in NMF as a generalised measure of the reconstruction error in [139] and later in [140]. Let us denote β-NMF the NMF factorisation obtained when minimising the β-divergence reconstruction error $D_\beta(\mathbf{R}|\mathbf{WH})$. A set of multiplicative update rules, parametrised by β can then be derived [141]:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T \left(\mathbf{R} \odot (\mathbf{WH})^{(\beta-2)}\right)}{\mathbf{W}^T (\mathbf{WH})^{(\beta-1)}} \tag{2.27}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{\left(\mathbf{R} \odot (\mathbf{WH})^{(\beta-2)}\right) \mathbf{H}^T}{(\mathbf{WH})^{(\beta-1)} \mathbf{H}^T} \tag{2.28}$$

where $\odot$ denotes the element-wise multiplication, the division is also element-wise.

Penalty terms Υ can be used to encourage certain behaviours in NMF, such as sparse activation [142] or co-ocurrence constraints [143]. The cost function to minimise with respect to $\mathbf{H}$, $\mathbf{W}$ and Υ in order to learn the factorisation is then:

$$D_\beta(\mathbf{R}|\mathbf{WH}) + \alpha\Upsilon \tag{2.29}$$

where α is a parameter to control the weight of the penalty. The multiplicative update rules that can be derived to solve this optimisation problem depend on the penalty that is applied. For example, with the application of sparse activation constraints, the β-NMF multiplicative update rules given in equations (2.27) and (2.28) typically become [144]:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \left[\frac{\mathbf{W}^T \left(\mathbf{R} \odot (\mathbf{WH})^{(\beta-2)}\right)}{\mathbf{W}^T (\mathbf{WH})^{(\beta-1)} + \alpha\Psi_{\mathbf{H}}}\right]^{\varphi(\beta)} \tag{2.30}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left[\frac{\left(\mathbf{R} \odot (\mathbf{WH})^{(\beta-2)}\right) \mathbf{H}^T}{(\mathbf{WH})^{(\beta-1)} \mathbf{H}^T + \alpha\Psi_{\mathbf{W}}}\right]^{\varphi(\beta)} \tag{2.31}$$

where $\Psi_{\mathbf{H}}$ and $\Psi_{\mathbf{W}}$ typically describe the gradient of the penalty term, α controls the weight of the penalty and $\varphi(\beta)$ is a parameter that varies with β and the penalty used to ensure descent of the cost function at each iteration.

A range of penalised NMF methods are considered for the detection of metric modulations in Chapter 7.

2.7 K-means Clustering

K-means clustering aims at partitioning a large number of data points (the observations) into a fixed number of clusters K defined beforehand. Each observation is assigned to the

cluster with the closest mean. The cluster centroid, which is the mean of the observations it contains, may then be regarded as a prototypical representation of the cluster content. The partition of the observations is optimised by minimising a within-cluster cost that is a distance between the cluster centroid and all the cluster elements, typically the squared euclidian distance. Let $\mathbf{R} \in \mathbb{R}^{M \times N}$ be the observations matrix, whose columns are N observations vectors of dimension M so that $\mathbf{R} = (\mathbf{r}_1, ..., \mathbf{r}_N)$, $\mathcal{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_K\}$ be the set of cluster centroids and $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_K\}$ be the set of clusters. The assignment of each one of the N observation vectors to exactly one of the K cluster seeks to minimise the function

$$\sum_{\mathbf{r} \in \mathcal{R}} \min_{\mathbf{z} \in \mathcal{Z}} ||\mathbf{r} - \mathbf{z}_i||^2 \tag{2.32}$$

where $|| \cdot ||$ denotes the euclidian distance operator.

Solving this optimisation problem is computationally difficult but algorithms such as the standard Lloyd's algorithm guarantee a quick convergence to a local minimum in the observation data space [145]. It is structured as follows:

1. Initialisation step: Choose K cluster centroids $\{\mathbf{z}_1, ..., \mathbf{z}_K\}$.
2. Assign each observation $\mathbf{r}$ to the cluster having the nearest centroid.
3. Update each cluster centroids $\mathbf{z}_i$ to be the mean of the clusters generated in step 2: $\mathbf{z}_i = \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{r} \in \mathcal{C}_i} \mathbf{r}$. Then compute the difference between the old and new centroid.
4. Repeat steps 2 and 3 until the set of cluster centroids $\{\mathbf{z}_1, ..., \mathbf{z}_K\}$ does not vary anymore. In practice, this is implemented by stopping the iterative updates when the variation of the cluster centroids falls below a given convergence threshold.

Several approaches for choosing the initial centroids are possible, two of which are most commonly used. The first consists in arbitrarily picking K observations from $\mathbf{R}$ and use these as cluster centroids. The second consists in randomly assigning each observation $\mathbf{r}$ to a cluster and then computing the cluster centroids as in the update of step 3. This has the effect of locating the initial means near the centre of the data distribution. Given that Lloyd's algorithm only converges towards a local minimum, the clustering result depends on the initialisation. Arthur and Vassilvitskii proposed an improved initiation strategy,

named "K-means++", that they found to lead to better clustering results [146]. The idea in their approach is to choose initial cluster centroids that are more uniformly distributed in the data space. They implement this constraint by performing the initialisation step as follows:

1. (a) Choose the first centroid $\mathbf{z}_1 = \mathbf{r}$ where $\mathbf{r}$ is picked uniformly at random in the observations space.

   (b) Take a new centroid $\mathbf{z}_i$, choosing $\mathbf{r}$ with probability $\frac{d(\mathbf{r})^2}{\sum_{\mathbf{r}\in\mathcal{R}} d(\mathbf{r})^2}$ where $d(\mathbf{r})$ is the shortest distance from an observation to the closest centroid already chosen.

   (c) Repeat step (b) until K centroid are chosen.

The uniformity of distribution of the centroids is effectively enforced by step 1 (b) in which the probability for choosing an observation as the new centroids is proportional to the squared shortest distance between the observation and the closest centroid already chosen. In other words, this strongly favours observations that are located far apart from existing centroids in the observations space, thereby globally favouring uniformity of the centroid locations distribution. The following steps are identical to Lloyd's algorithm.

K-means is used as a comparison method for clustering in Chapter 7. It was chosen over other clustering techniques because it is a standard clustering algorithm. We use the K-means ++ algorithm in all our experiments.

2.8 Harmonic and Percussive Sound Separation using Median Filtering

Fitzgerald proposed a method to separate harmonic and percussive parts of an audio recording by applying median filtering to the audio spectrogram [147]. The working principle of this technique effectively consists in separating energy distributions that form vertical and horizontal lines in the magnitude spectrogram respectively. Applying median filtering across successive frames suppresses vertical lines and thereby can be seen as an enhancement of the horizontal structures. Conversely, applying median

filtering across the frequency axis suppresses horizontal structures and can then be seen as an enhancement of vertical structures. Fitzgerald applied this technique to percussive/harmonic separation on the grounds that percussive events typically result in short broadband energy bursts (i.e. vertical lines) and steady state harmonic sounds result in a series of horizontal lines representing the energy of the partials. Harmonic and percussive sound separation is beyond the scope of this thesis but this technique generalises to any multi-dimensional feature, hence its use to enhance horizontal structures in a rhythmogram feature (cf. Chapter 7). We briefly describe the calculation procedure below, following [147].

Given an input vector $x(n)$, the median filter $\mathcal{M}$ can be defined as:

$$\mathcal{M} : x(n) \mapsto \text{median}\left[x\left(n - \frac{\ell - 1}{2} : n + \frac{\ell - 1}{2}\right)\right] \tag{2.33}$$

where ℓ is odd and defines the number of samples over which the median filtering is applied.

Let us notate the audio magnitude spectrogram $\mathbf{X}$, so that $\mathbf{x}_n$ is the n^{th} spectrogram frame and $\mathbf{x}_m$ the m^{th} frequency slice. Then, a percussion-enhanced spectrogram frame $\mathbf{p}_n$ is computed by median filtering $\mathbf{x}_n$:

$$\mathbf{p}_n = \mathcal{M}\left\{\mathbf{x}_n, \ell_{perc}\right\} \tag{2.34}$$

where ℓ_{perc} is the length of the median filter used to filter out horizontal lines and therefore enhance percussive (i.e. vertical) structures. A percussion enhanced spectrogram $\mathbf{P}$ is then constructed by concatenating the percussion-enhanced frames $\mathbf{p}_n$. Similarly, a harmonic-enhanced spectrogram $\mathbf{H}$ is constructed by concatenating harmonic-enhanced spectrogram slices $\mathbf{h}_m$, obtained by median filtering spectrogram frequency slices $\mathbf{x}_m$:

$$\mathbf{h}_m = \mathcal{M}\left\{\mathbf{x}_m, \ell_{harm}\right\} \tag{2.35}$$

where ℓ_{harm} is the length of the median filter used to enhance harmonic (i.e. horizontal) structures.

The harmonic and percussion suppressed spectrograms can then serve as a basis to compute masks to be applied to the original complex spectrogram $\hat{\mathbf{X}}$. Two masking strategies were proposed. First a hard, or binary, masking, where each spectrogram bin is assigned either to the harmonic or percussive part, hence each element of the masks being defined as:

$$\mathbf{M_H}(m,n) = \begin{cases} 1 & \text{if } \mathbf{H}(m,n) > \mathbf{P}(m,n) \\ 0 & \text{otherwise} \end{cases} \tag{2.36}$$

$$\mathbf{M_P}(m,n) = \begin{cases} 1 & \text{if } \mathbf{H}(m,n) < \mathbf{P}(m,n) \\ 0 & \text{otherwise} \end{cases} \tag{2.37}$$

Alternatively, a soft masking strategy can be employed, based on Wiener Filtering:

$$\mathbf{M_H}(m,n) = \frac{[\mathbf{H}(m,n)]^v}{[\mathbf{H}(m,n)]^v + [\mathbf{P}(m,n)]^v} \tag{2.38}$$

$$\mathbf{M_P}(m,n) = \frac{[\mathbf{P}(m,n)]^v}{[\mathbf{H}(m,n)]^v + [\mathbf{P}(m,n)]^v} \tag{2.39}$$

where v is the element-wise exponent, typically set to 1 or 2. The complex harmonic and percussion-enhanced spectrograms are then computed as

$$\hat{\mathbf{H}} = \hat{\mathbf{X}} \odot \mathbf{M_H} \tag{2.40}$$

$$\hat{\mathbf{P}} = \hat{\mathbf{X}} \odot \mathbf{M_P} \tag{2.41}$$

where $\odot$ indicates element-wise multiplication and $\hat{\mathbf{X}}$ is the complex spectrogram. This last operation enables the inversion of the filtered complex spectrogram back to an audio signal.

Harmonic-Percussive separation was originally introduced to be applied to the audio spectrogram, for source separation. In this thesis it applied to the metergram, and

used as a pre-processing step in our metrical structure based segmentation procedure presented in detail in Chapter 7.

2.9 Markov Models

First introduced in the 1960s, Hidden Markov Models (HMM) have since proven to be precious tools for analysis of time series data. In the audio domain, they are particularly well-known for speech recognition applications [148, 149]. In the music domain, HMMs have become a standard approach for chord recognition, see for example [150–153], but have also been used for key estimation [154], structural segmentation [155] and transcription [133, 156]. In this thesis, a HMM is used as part of the metric modulation detection scheme introduced in Chapter 7. A brief description of Markov chains and Hidden Markov Models (HMM) is given in this section in order to specify the notions, terminology and notation used in this thesis. For a more exhaustive and detailed description of HMMs we refer the reader to Rabiner's reference tutorial [149].

2.9.1 Markov Chain

Let us consider a system that may be described as being in a particular state at any time. The set of all K possible states for the system is labelled *state space*:

$$\mathcal{S} \triangleq \{\psi_1, \psi_2, \cdots, \psi_K\} \tag{2.42}$$

At every time step t, a change of state occurs (possibly back to the same state). In a probabilistic framework, the probability of being in state k at time t is conditioned on all preceding states. A system satisfying the property such that the state at any time q_t is conditioned only on the preceding state q_{t-1}, which is expressed as:

$$P\left(q_t = \psi_j | q_{t-1} = \psi_i, q_{t-2} = \psi_k, \cdots\right) = P\left(q_t = \psi_j | q_{t-1} = \psi_i\right) \tag{2.43}$$

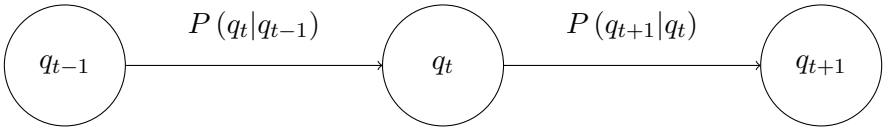


FIGURE 2.10: **A Markov chain over three time steps.** The arrows represent conditional dependence. q_t is the state at time step t, and $P(q_{t+1}|q_t)$ is the transition probability from state q_t to state q_{t+1}.

is therefore called a first order *Markov chain*. This system having only a first order temporal conditional dependence, the state transition probabilities can be defined as:

$$p_{ij} = P(q_t = \psi_j | q_{t-1} = \psi_i) \tag{2.44}$$

where $1 \leq i, j \leq K$ and $p_{ij} \in [0, 1]$. The transition probabilities moreover obey the standard stochastic constraint:

$$\sum_{j=1}^{K} p_{ij} = 1 \tag{2.45}$$

A state transition probability matrix Θ, in which each entry is a p_{ij}, then characterises all possible transitions. A schematic representation of a first order Markov chain is given in Figure 2.10. Note that systems incorporating a conditional dependence on the N preceding states are then known as N^{th} *order Markov chains*.

2.9.2 Hidden Markov Model

The Markov chain could be called an observable Markov model since the output of the process is the state at each time step. In other word, the states of the model itself are observable. In the context of musical analysis, say for example chord recognition, the states could correspond to chord labels. Therefore a Markov chain represents observable chord labels. However, in audio-based processing the chords labels are not directly observable and must be inferred from the audio signal or suitable features thereof. The hidden Markov models extend the concept of Markov chain by involving two layers. The lowest layer is visible and therefore represents the *observations* (i.e. the audio signal, or features thereof). The highest layer is a Markov chain representing the underlying states of the system (the musical feature of interest, e.g. chords), which is not visible, hence the name 'hidden'. It is then considered that the observations may be explained by the

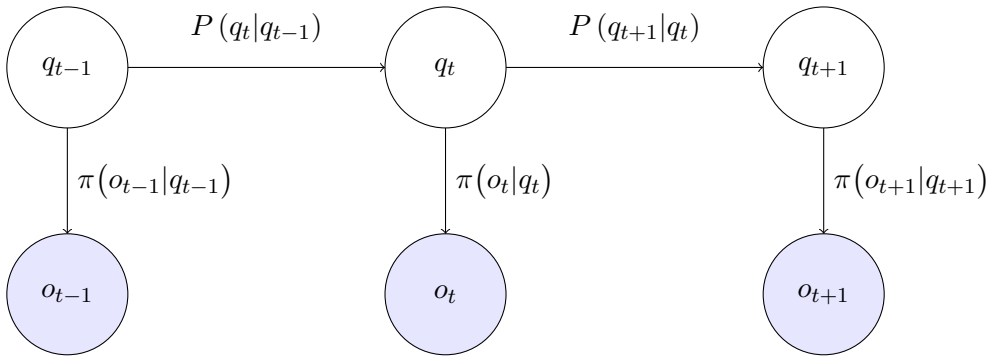


FIGURE 2.11: **Illustration of a Hidden Markov Model**. The hidden layer of the model corresponds to the Markov chain of Figure 2.10. The lower layer represents the observations. Arrows represent conditional dependence. At each time step, the observations are conditioned only on the current state.

hidden states of the system. The two layers are connected to each other via *emission probabilities*, which provide a statistical description of how observations relate to hidden states. Furthermore, the observation at time t, notated o_t, is conditioned only on the state of the Markov chain at this time, with emission probability $\pi\big(o_t|q_t\big)$. This structure is illustrated in Figure 2.11.

In a typical MIR scenario, the goal is to uncover the hidden state sequence given the observations (i.e. the chord sequence from the signal). This is a difficult problem in general. However, the Viterbi algorithm offers an efficient way to find the most likely state sequence given the observations and the HMM [149, 157, 158].

The choice of the model parameters, i.e. state space, state transition probabilities and emission probabilities, is highly task-dependant. While choosing an appropriate state space is usually straightforward, deriving state transition probabilities and emission probabilities is not trivial. Typically these probabilities can either be set manually, perhaps using prior or expert knowledge, or learnt from data. Again, these choices are highly dependant on the purpose of the use of the HMM and on the data available.

One type of use for a HMM is *post-filtering* [7, 150]. In this setting, a rough estimate of the information to be extracted has already been obtained from a given processing pipeline. The HMM is then used to filter this rough estimate with the aim of obtaining a cleaner, and therefore more accurate and/or meaningful output. HMM post-filtering is particularly effective for stabilising stuttering estimates. By construction of this use case,

the desired output (i.e. a state sequence) closely ressembles the input of the filtering stage (i.e. the observations). A HMM for post-filtering metrical structure change estimates is described in section 7.4.5. We refer the reader to relevant literature for the description of other classes of use cases for HMM, e.g. [149].

In this thesis, a HMM is used a post-filtering stage of a segmentation algorithm, in order to eliminate spurious segmentation. As will be described in further details in Chapter 7, in this case the hidden states represent the different metrical structures present in a piece. Then, the decoded state sequence characterises the segmentation of the piece and state transitions correspond to metric modulations.

2.10 Structural Segmentation

At the most granular level music is made of sound events that may for instance be individual notes occurring over time. The grouping, combination and organisation of these events results in higher levels of structure, such as motifs, patterns, phrases and sections. Their relative organisation then defines the overall layout of a piece. Such a structure is broken down in parts with a musical role. For instance in classical music these parts could consist of exposition, development and recapitulation of a movement while in popular music the parts typically correspond to the verse, chorus and bridge of a song. Note that, in this sense, the organisation of musical events considered as structural segmentation is distinct from that considered as the metrical structure [41]. The task of *structural segmentation* is concerned with automatically retrieving this structural organisation, i.e. the musical form, from audio recordings[9]. In this sense, retrieving the structural segmentation is beyond the scope of this thesis. However, because the metrical structure detection approach introduced in Chapter 7 borrows from it, the general principles of structural segmentation as well as the corresponding evaluation metrics are briefly presented in this section.

[9] Although this task can also be performed on symbolic representations, we recall that focus in this thesis on analyses on audio recordings.

2.10.1 Musical Dimensions

Musical structure is multifaceted. Composers have a large number of musical dimensions on their palette to materialise the musical form. Attempting to produce a comprehensive list of such attributes is vain, but one can cite harmony, melody, rhythm, timbre, instrumentation, repetition, dynamics, audio effects, lyrics or key as examples. Unless prior knowledge is available for the pieces under study, it is impossible to know a priori what musical dimension(s) have been used by composers and/or producers to structure the piece. This is why it is common when attempting to retrieve structural segmentation from musical audio to make assumptions as to what these dimensions might be. The first necessary step in the automatic processing then consists in extracting features that characterise the musical dimensions of interest.

When attempting to retrieve timbre-based segmentation, the Mel-Frequency Cepstral Coefficients (MFCCs) is a commonly used feature [63, 159]. Similarly, the chromagram may be used to retrieve structure based on harmony [160] and so may the rhythmogram for rhythm-based segmentation [161]. However, since the musical form is typically materialised by the combination of more than one musical dimension, methods that combine different features [82, 162–164] or operate directly on spectral representation of audio [155, 165, 166] have been introduced with success.

2.10.2 Musical Structure Analysis Strategies

Once the relevant features have been extracted from the audio recording, the musical form is to be retrieved from them. A number of approaches have been proposed to achieve this goal. On the ground of common underlying concepts, they can be loosely grouped in three categories that were devised in previous work [7, 167], which we briefly summarise below.

Novelty-based Segmentation

Novelty-based methods for structural segmentation rely on the *contrast* between successive sections. It is assumed that a change of section coincides with a change in some

musical dimensions, and therefore in the feature(s) of interest. The goal is then to retrieve the locations of the changes in time in order to determine the boundaries between two successive sections or segments. The classic approach to novelty-based segmentation was introduced by Foote in [63]. In a first step, a self-similarity matrix (SSM) of the evolution of the feature of interest over time is computed. Because this method is applied to the rhythmogram later in this thesis, let $\mathbf{r}$ be the N frames feature vector. Each element of the self-similarity matrix $\mathbf{B} \in \mathbb{R}^{N \times N}$ is the defined as:

$$\mathbf{B}(i,j) = \|\mathbf{r}_j - \mathbf{r}_i\| \tag{2.46}$$

where $\mathbf{r}_j$ and $\mathbf{r}_i$ represent the feature vector corresponding to the j^{th} and i^{th} frame respectively and $||\cdot||$ denotes the euclidian distance operator. Though the self-similarity matrix may be computed using other distance measures. Foote suggested the cosine distance as a possible alternative that is invariant to the norm of the frame vectors considered. Nevertheless, the euclidian distance is widely used in the literature and is used here to facilitate comparison with existing work.

A novelty function is then computed by correlating the main diagonal of the SSM with a checkerboard kernel. Peaks in this novelty curve are expected to indicates significant changes in the feature vectors. The elementary kernel κ is defined as:

$$\kappa = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.47}$$

The size of the kernel defines the timespan over which the novelty calculation is performed at every step. The size of a block kernel matrix κ_{Block} can be adjusted by computing the Kronecker product, notated $\otimes$, of the elementary kernel κ with a matrix of ones, which size determines the size of the resulting kernel. For example, using a 2×2 matrix of ones:

$$\kappa_{\text{Block}} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \tag{2.48}$$

In order to minimise edge effects, Foote recommended to smooth the block kernel using a radially tapered Gaussian window:

$$\kappa_{\text{Gauss}}(i,j) = \exp\left(-\xi^2\left(i^2+j^2\right)\right) \cdot \kappa_{\text{Block}}(i,j) \tag{2.49}$$

where $\xi > 0$ allows for the adjustment of the tapering. Foote suggests choosing ξ to equate to half the number of columns in κ_{Block}. The kernel may finally be normalised in order to compensate the influence of its size:

$$\kappa_{\text{Norm}}(i,j) = \frac{\kappa_{\text{Gauss}}(i,j)}{\sum\limits_{i,j} |\kappa_{\text{Gauss}}(i,j)|} \tag{2.50}$$

The novelty curve is then obtained by sliding the checkerboard along the main diagonal of the SSM and summing the element-wise product of κ_{Norm} and $\mathbf{B}$:

$$\zeta(n) = \sum_{i,j} \kappa_{\text{Norm}}(i,j)\mathbf{B}(n+i,n+j) \tag{2.51}$$

Segment boundaries are expected to yield local maxima in the novelty curve and may therefore be recovered by peak picking. As such, the novelty curve is analogous to an onset detection function in which peaks reveal musical event onsets. An example of application of the Foote method on the metergram is given in Figure 7.4.

A number of variations on this method have been proposed in the literature. Their description is beyond the scope of this thesis, we refer the interested reader to [7] (Chapter 4) for a more exhaustive description.

Repetition-based Segmentation

Repetition is a fundamental aspect of music and of particular importance in musical form. Musical sequences such as phrases, themes or patterns are often repeated. Musical form may also be defined by repetition. For instance, in popular music the chorus is typically repeated several times over the course of a piece. Note that in contrast with novelty-based methods that only produce segment boundaries, repetition-based methods additionally describe the relation between parts. As such they provide a structural

analysis that goes beyond the sole segmentation. A major drawback of repetition-based methods, however, is that they cannot detect segments that are never repeated. Due to this limitation, repetition-based methods are not applicable to the problem addressed in Chapter 7. Nevertheless, a few pointers to relevant literature are given in the following.

A range of methods rely on the detection of repetitions for retrieving the structural segmentation of a piece, see for example [168–170]. SSM are also applicable in this scenario as repetitions result in off-diagonal path structures [7]. The number and length of repetitions is related to the number and length of off-diagonal paths. The SSM can then be processed in order to perform repetition-based structural segmentation [171]. Departing from the SSM-based methods, Weiss proposed a variant of sparse convolutive Non-Negative Matrix Factorisation (NMF, described in section 2.6) to identify repeated patterns [172]. The NMF decomposition then learns the repeated patterns and their temporal activation reveals the musical structural.

Homogeneity-based Segmentation

Sections of a music piece tend to exhibit some degree of homogeneity, with respect to some musical dimensions such as tempo, instrumentation or key. In other words, the musical structure may be characterised by a relative consistency of some musical descriptors within sections. Homogeneity based structural segmentation retrieval then consists in grouping similar frames of a given feature vector in contiguous clusters.

A variety of methods have been proposed to achieve this. Levy used a Hidden Markov Model (HMM) to perform the clustering [155]. Given that this formulation of structural segmentation is effectively a clustering problem, a comparison to standard clustering techniques such as K-means clustering is also performed. Mc Fee proposed to use graph theory as an alternative approach to clustering [173]. Note that, by construction, segments belonging to the same cluster are somewhat similar. In this sense, formulating the segmentation task as a clustering problem enables an estimation of similarity between different segments.

SSM are also usable in this context as homogeneous regions of the audio signal translate into blocks in the SSM. A number of authors have therefore approached the task of homogeneity-based structural segmentation by exploiting this property [7]. Blocks on the main diagonal represent the succession of homogeneous regions (i.e. sections) while off-diagonal blocks reveal the similarity between non-adjacent blocks. As such, off-diagonal blocks may also be indicators of repetition. In a SSM-based scenario, retrieving the segmentation consists in recovering the block structure. Enhancement of the block structure has been proposed to facilitate segmentation retrieval in [174]. Kaiser also proposed to use NMF to learn a decomposition of the SSM in order to cluster blocks, and therefore reveal similarities [175].

In Chapter 7, we introduce a method to track zones of stable metrical structure, and by extension metric modulations, that is analogous to homogeneity-based segmentation, using the metergram as a basis feature. In particular this is achieved by introducing a variation of sparse NMF.

2.10.3 Evaluation metrics

Structural segmentation is known as a challenging and multifaceted task. As a result, a number of evaluation metrics, offering a variety of viewpoints, have been introduced. We describe in this section a range of metrics used to evaluate segmentation algorithms. All segmentation algorithms are expected to produce a structural division of a music piece as an output. As such they are expected to produce, in some shape of form, either segment boundaries timestamps or segments temporal extent, or locations. On top of specifying regions or time instants corresponding to segment or segment boundaries respectively, some algorithm also aim at labelling the sections, producing labels such as “verse” and “chorus” or “A” and “B” for instance. In the work presented here, we do not address the issue of the semantic label assignments. In other words we do not intend to evaluate if a segment labelled as a “verse” effectively corresponds to a verse and not another label (e.g. a chorus). As a result the metrics we present here are not sensitive to the labels of the segments. Some of them are sensitive to segments clustering, however. Then it

does not matter if verses are effectively labeled as "verses", as long as all verses receive the same label - i.e. the belong to the same cluster.

As we will see later in Chapter 7, all of the metrics described below individually only provide a narrow insight into the quality of the segmentation, but their combination enables a far greater depth of analysis.

2.10.3.1 Boundaries Retrieval

Let us first introduce metrics that measure the ability of a system to accurately retrieve segment boundary positions. It is to be noted at this point that such metrics implicitly assume a model of structural segmentation whereby segments are delimited by boundaries that are time instants of zero duration. In other words a transition from one segment to another is modelled as a sudden change, which constitutes a limitation for this metric. However, it is common practise when using these metrics to allow a certain tolerance window in the evaluation to account for the fact that segment boundaries might not be as sharp as a time instant of zeros duration.

Hit rate

Segment boundaries estimated by the algorithm are regarded as correct — or to be a *'hit'* — if they are within a tolerance window from a boundary in the ground truth. Values for the tolerance window commonly found in MIR literature are 0.5s [164] and 3s [155]. The MIREX structural segmentation task[10] evaluates algorithms using the hit rate metric both with 0.5s and 3s tolerance. Given the matches between the ground truth and estimated boundaries, the boundary retrieval precision recall and F-measure rates are calculated.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.52}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.53}$$

$$\text{F-measure} = 2 \cdot \frac{Precision \cdot Recall}{precision + recall} \tag{2.54}$$

[10] http://www.music-ir.org/mirex/wiki/2016:Structural_Segmentation

Where TP, FP and FN are the number of true positives, false positives and false negatives respectively. Note that these are standard information retrieval metrics, which use is therefore not limited to segmentation.

Median Deviation

As opposed to the Hit rate metrics, which assignes a binary value to each boundary (hit or non-hit), the median deviation aims at providing a quantitative measure of the distance between ground truth and estimated boundaries. Two median deviations can be computed: the *median true-to-guess* (TTG), which is the median distance (in seconds) from boundaries in ground truth to the closest estimated boundaries and the *median guess-to-true* (GTT), which is the median distance (in seconds) from estimated boundaries to the closest boundaries in ground truth [164].

The choice of the median rather than the mean has the advantage of being robust against the presence of outliers. As a result the *median true-to-guess* and *median guess-to-true* metrics provide a quantitative measure of the most prominent trend in terms of segment boundaries location.

2.10.3.2 Frames clustering

Structural segmentation may be interpreted as a clustering process: the audio frames belonging to a given segment are seen as belonging to a cluster. As a consequence the pairwise precision, recall and f-measure, which are standard metrics for cluster quality evaluation, can be used to evaluate the segmentation as suggested by Levy and Sandler [155]. Pairs of frames from the machine-estimated segmentation and the reference segmentation are compared. The pairwise precision rate, *ppr*, pairwise recall rate, *prr*, and pairwise F-measure, *pfm* are calculated as

$$ppr = \frac{|P_e \cap P_a|}{P_e} \tag{2.55}$$

$$prr = \frac{|P_e \cap P_a|}{P_a} \tag{2.56}$$

$$pfm = 2 \cdot \frac{ppr \cdot prr}{ppr + prr} \tag{2.57}$$

where P_e is the set of similarly-labelled pairs of frames estimated by the machine and P_a is the set of similarly-labelled pairs of frames annotated in the human-generated reference ground truth.

This metric effectively quantifies the amount of overlap between segments generated by the machine and the human ground truth. As such, it is very complementary to the boundaries retrieval metrics introduced previously, which exclusively quantify the accuracy of the boundary locations.

2.10.3.3 Normalised conditional entropies

Here again, the structural segmentation is represented as a sequence of frame labels, each label denoting the frame membership to a cluster (i.e. a segment). Let A and E be the sequences of annotated and machine-estimated segmentation, respectively. Using the conditional entropy as an evaluation metric for structural segmentation was originally proposed by Abdallah in [176]. The conditional entropy $S(A|E)$ measures the amount of ground truth segmentation information missing from the estimated segmentation. Similarly, the conditional entropy $S(E|A)$ measures the amount of spurious information in the estimated segmentation. Just like the *mutual information* metrics presented in [176], the conditional entropy does not have an upper bound and the scale of the metric depends on the number of segments present in a song. The more segments and the more uniform their distribution, the higher the conditional entropy. This mathematical property has the disadvantage of not allowing a meaningful comparison of two musical pieces with a different number of segments.

Lukashevich introduced the over and under segmentation scores, which are based on conditional entropy, with an additional normalisation applied so that a meaningful comparison of songs with a different number of segments is made possible [177]. The over-segmentation S_o and under-segmentation S_u scores are defined as:

$$S_o = 1 - \frac{S(E|A)}{\log_2 N_e} \tag{2.58}$$

$$S_u = 1 - \frac{S(A|E)}{\log_2 N_a} \tag{2.59}$$

where N_e and N_a are the number of estimated and annotated segment clusters respectively. We refer the reader to the original publication for a detailed derivation of these expressions [177]. Both scores range from 0 to 1. They are maximal when the annotated and estimated segmentations match perfectly and tend towards 0 when the frame labels are randomly attributed. Note that the interpretation of these scores is counter-intuitive: the over (resp. under) segmentation score is takes small values when the estimated structure highly over (resp. under) segments the piece and conversely.

2.11 Summary

The theoretical concepts and computational techniques that are either underpinning or used in this thesis were introduced in this chapter. Because we are concerned here with rhythmic properties of music, we first introduced the music theory concepts and corresponding nomenclature on which the work presented in all subsequent chapters relies.

A sizeable body of work focusing on the automatic analysis of rhythmic properties from musical audio recordings already exists. In sections 2.2 to 2.5 we review computational methods and representations that are relevant to the analysis of metric modulations, such as onset detection, rhythmograms, metrical structure and tempo estimation. Onset detection is typically the first processing step in a method for rhythmic analysis and is therefore used in the processing involved in Chapters 4 to 7. Similarly, we demonstrate in Chapters 4 and 5 that the rhythmogram captures information related to the metrical structure of music and is therefore exploited for the detection of metric modulations in Chapter 7. Since the estimation of the tempo and metrical structure are related to the task of metric modulation tracking, the corresponding processing principles and strategies presented in sections 2.4 and 2.5 are referred to in Chapters 4, 5 and 7.

The detection of metric modulations is formulated here as a segmentation retrieval problem, which will be described in greater details in Chapters 6 and 7. Although it is, to

the best of our knowledge, the first time this type of approach is proposed to address the detection of metric modulations, it bears formal similarities with the well known task of structural segmentation. For this reason, the paradigm of structural segmentation along with the standard evaluation metrics were presented in section 2.10. However, a number of mathematical models, such as Hidden Markov Models, and numerical optimisation techniques, such as Non-negative Matrix Factorisation, are employed in subsequent chapters of this thesis and therefore form a pre-requisite for the presentation of segmentation retrieval schemes. For this reason, these were presented in sections 2.6 to 2.9, prior to the introduction of structural segmentation.

Chapter 3

Datasets

Typically, the performance of a given algorithm is estimated by evaluating how well it reproduces a ground truth. In order to provide robustness against outliers and generalisability of results, the evaluation is typically carried out on a large number of examples from which statistics are derived. In this context, evaluating automatic musical features estimation systems in a formal and quantitative way requires a corpus of audio recordings as well as the corresponding reference ground truth, produced in relevance to a specific task (e.g. structural segmentation, beat tracking etc.). In this chapter, we present the datasets of audio recordings and reference annotations used for evaluation purposes in this thesis. In particular, we briefly describe contents and properties relevant to the tasks addressed using them. Some of these are standard datasets that have been widely used in MIR research for several tasks. In addition, we introduce two new datasets for the evaluation of metrical structure and metric modulation estimation algorithms respectively and describe their creation process and contents.

Music tends to be equivocal by nature, which means that an absolute ground truth might not always exist. As a consequence, human expert annotations are commonly used as a proxy for ground truth in order to evaluate musical feature estimation systems. However, it has been shown that different people may exhibit different reactions or behaviours and therefore potentially produce different annotations in response to music [21, 178]. In this context, producing a rigorous evaluation of algorithms against "ground truth" that contains variability itself is not trivial. Over the last two decades, it has been common

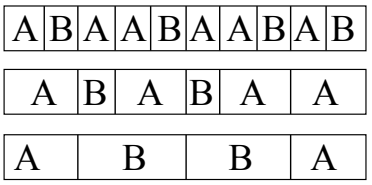


FIGURE 3.1: Simple example of several possible segmentations of the same piece

practice to collect one annotation per excerpt and evaluate the algorithm against it. But this is merely an evaluation of the ability of the algorithm to reproduce the annotator's interpretation of the piece. In response to this observation, it has been shown that collecting multiple annotations and assessing the level of agreement (or disagreement) between annotators enables more meaningful and more generalisable evaluation [10, 179, 180]. Note that although it has not been largely addressed in music-related tasks, this problem is not specific to music, see for example [181]. In Section 3.7, an analysis of the inter-annotator agreement in the newly introduced GTZAN-Met dataset is reported. The benefits of having multiple annotations for the interpretation of the results of the evaluation of algorithms using this dataset are shown in Chapter 4.

3.1 SALAMI Dataset

Emerging from the Structural Analysis of Large Amounts of Musical Information (SALAMI) project, the SALAMI dataset was created for evaluation of structural segmentation algorithms [182]. Most structural segmentation test sets provide annotations of the musical structure either as letter markings (eg. AABA) or functional annotations such as Verse, Chorus, Bridge etc. However, the structural segmentation of a piece can be interpreted in multiple ways [169], as illustrated on an example in Figure 3.1. When transcribing the structural segmentation of a piece, annotators effectively provide one possible interpretation of this structure. One may observe that the multiple segmentation interpretations highlight a hierarchical organisation of the structure of a piece, which is not captured by annotations at a single level (e.g. verse, chorus etc.). This is a difficulty for the evaluation of segmentation algorithms, as evaluating against annotations at a given (and unique) level of granularity may give distorted measure of the algorithm performance, or at least is a truncated analysis. The SALAMI dataset improves on this aspect by

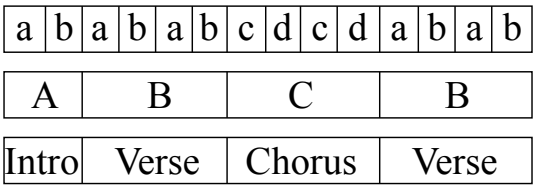


FIGURE 3.2: **Example of segmentation annotation from the SALAMI dataset (track 6, annotation 1)**. The annotation is produced at two levels of granularity denoted by lowercase and uppercase letters respectively. Functions, such as Intro, Verse and Chorus, are also annotated and are typically associated to segments denoted by uppercase letters

providing annotations of structural segmentation at two different granularity levels, denoted by lower case and upper case letters respectively. By this means, the annotations locate the segment boundaries, specify segments similarity and include a description of the hierarchical organisation of the structural segmentation (e.g. segment A is made of the sequence of sub-segment a and b). In addition, semantic functions, such as Verse Chorus, or Bridge are provided. They typically refer to segments described by uppercase letters. An example of annotation from the SALAMI dataset is given in Figure 3.2. Moreover, each track was annotated by one or two annotator(s), so that inter-annotator disagreement can be assessed in the latter case.

Initially collected as a unit, the SALAMI dataset is split in two parts that have distinct purpose: a publicly available part, open for researchers to evaluate their algorithms, and a hidden part used in the MIREX structural segmentation task. In the MIREX challenge, only the uppercase letters segmentation layer is used. The publicly available dataset[1] is made of 827 audio files and corresponding annotations. The results of the MIREX structural segmentation task on the SALAMI dataset are used to compare the segmentation results obtained in Chapter 7.

3.2 SMC Dataset

Holzapfel *et. al.* proposed a method to identify challenging tracks for beat tracking without relying on annotated ground truth in [3]. Essentially, it consists in measuring the disagreement between a committee of automatic beat trackers, thus considering as

[1] https://ddmal.music.mcgill.ca/research/salami/annotations

challenging the tracks that result in a high disagreement, and conversely. Using this approach, they selected a corpus of 270 music samples that are challenging for beat tracking amongst a database of 678 manually chosen excerpts of 40s length. The authors restricted their selection to western music "*because it is not always apparent how the notion of beat is used in music of other cultures*". To this set were added 19 excerpts deemed as easy using the same approach (i.e. resulting in a strong agreement between beat trackers). The resulting 289 excerpts were then manually annotated. In this process, annotators had the possibility to reject a piece if the annotation process seemed intractable to them. 72 pieces were indeed rejected this way. The remaining 217 excerpts constitute what we refer to as the SMC dataset[2] here. The musical genres represented are predominently classical music, romantic music, film soundtracks, blues, chanson and solo guitar compositions.

Holzapfel *et. al.* argue that the 72 pieces that humans could not successfully annotate are not helpful to improve the state of the art in beat tracking research. Given that automatic beat tracking consists in retrieving the positions of beats as perceived by human listeners, it is indeed unclear how to evaluate a beat tracker on a piece for which humans cannot find a beat. This is echoed by the discussion of the outcomes of the analysis of inter-annotator disagreement on the GTZAN-Met dataset detailed in Section 3.7, which seem to suggest that the notion of beats, and more generally of pulse, is not applicable to all musical styles, even within the western genres.

Nevertheless, the SMC dataset contains 198 excerpts that are challenging for automatic beat trackers but tractable by humans, therefore providing a valuable resource for the advance of beat tracking research. The SMC is used in Chapter 5 for a slightly different purpose. It provides excerpts that are challenging for beat trackers and are therefore likely to make algorithms fail. It also provides a subset of tracks for which beat tracking is expected to be successful. As such it provides a useful test set for evaluating the reliability estimation method proposed in Chapter 5.

[2]http://smc.inesctec.pt/research/data-2/

3.3 GTZAN audio dataset

Originally curated manually by Tzanetakis to evaluate a genre classification algorithm in 2002 [183], the GTZAN dataset has been used extensively in the last decade — Sturm counts at least 100 published papers using it [184]. The dataset is composed of 1000 audio music excerpts of 30 seconds duration from commercial recordings grouped in 10 genre classes, namely Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae and Rock, with 100 tracks in each, all presented as a mono PCM .wav file 16bit sampled at 22.05kHz. The tracks are labelled with their genre and an index ranging from 00000 to 00099 which results in track labels such as '*pop.00055*'. No other identifiers (such as track and artist name) are provided with the dataset.

Sturm carried out an exhaustive study of the content of the dataset in [184]. Using automatic fingerprinting, he was able to identify a very large portion of the recordings present in the dataset. A number of tracks not identified automatically were identified manually by a variety of contributors, but not all the dataset contents have been identified yet. This identification clearly reveals strong biases in the composition of the dataset that limit the representativity of some genre classes. For instance, most of the reggae tracks are by Bob Marley. Undeniably, Bob Marley had an enormous influence on the genre, but the generalisability of the classification performance of any algorithm on this reggae class is questionable as a result. Moreover, Sturm identified several types of singularities of the dataset (that he named "*faults*") such as mislabelling, distortions and repetitions[3] and demonstrated how they could affect the scope, if not the validity of the conclusions they are used to draw on genre classification [184].

With 1000 tracks, the GTZAN dataset is not among the largest publicly available datasets, but it contains a good degree of musical variety and a complete manual annotation is manageable. Two sets of key reference annotations[4,5] [185] as well as a set of tempo annotations[6] exist for the GTZAN dataset. The variety of annotations available

[3] In addition to the published papers, the "faults" and dataset content are listed online respectively:
http://www.eecs.qmul.ac.uk/∼sturm/research/GTZANtable2/index.html
http://vbn.aau.dk/files/206248560/GTZANindex.txt

[4] https://github.com/alexanderlerch/gtzan_key/blob/master/gtzan_key/KeyEnumeration.txt

[5] http://visal.cs.cityu.edu.hk/downloads/#gtzankeys

[6] http://www.marsyas.info/tempo/genres_tempos.mf

for this dataset make it particularly valuable. The musical excerpts being 30 seconds long, it may also be assumed that they will feature a relatively stable metrical structure rhythmic content. For these reasons, the GTZAN was chosen as the corpus for which to create a new set of metrical structure annotations, which we describe in Section 3.5.

3.4 GTZAN-rhythm annotations corpus

Simultaneously, but independently of the introduction of the GTZAN-Met dataset, a second corpus of rhythmic annotations for the GTZAN dataset was released by Marchand *et. al.* [186]. For conciseness, it will be referred to as GTZAN-Rhy in the remainder of this document The GTZAN-Rhy corpus consists of annotations of the beat and downbeat for every track. The beat subdivision[7] positions were annotated for the swung and compound meter (labeled as "*ternary*") excerpts only. A straight duple subdivision was assumed otherwise, and the corresponding eighth notes positions were not annotated. The annotations were produced semi-automatically: an estimate of beat and downbeat positions was automatically produced and corrected by the human annotator if necessary. A subdivision (duple or triple) of the beat was automatically generated and manually modified to align it to the actual position of the eighth notes, from which an estimation of the swing ratio was performed. The annotations were produced by two annotators (both researchers and practicing musicians) each one annotated half of the dataset. In order to give a quantitative estimate of the reliability of the annotations, 5% of the tracks (tracks numbered 95 to 99 for each genre) were annotated by both annotators and the agreement between them measured. An inter-annotator agreement F-measure of 0.91 for beat positions is reported.

3.5 GTZAN Metrical Structure annotations corpus

Sets of annotations for several metrical levels (typically beat and downbeat) are already available, see for example the Isophonics dataset[8] [187], the Million Song Dataset[9] [188]

[7] Typically corresponding to the eighth notes positions
[8] http://isophonics.net/datasets
[9] http://labrosa.ee.columbia.edu/millionsong/

or the GTZAN-Rhy dataset described in Section 3.4. To the best of our knowledge, there is no dataset for which audio recordings and annotations of all the metrical levels are publicly available, however. In this section, we introduce a corpus providing annotations of the pulse rate of every metrical level present in each track of the GTZAN dataset. It was first made public in [29]. For convenience, we will refer to it as GTZAN-Met in the remainder of this document.

Over the last two decades, it has been common practice to collect one annotation per excerpt and evaluate algorithms against it. Since music may be ambiguous, there may exist several interpretations of the same piece and it may not be trivial to state which one is more "correct" that the other(s). One way to account for the inherent ambiguity of music in the evaluation of automatic algorithms is to collect multiple annotations for the same piece. It has then be shown that doing so and assessing the level of agreement (or disagreement) between annotators provides extra insight and is therefore a step towards more complete, more meaningful and more generalisable evaluation [10, 179]. Note that although it has not been largely addressed in music-related tasks, this problem is not specific to music, see for example [181]. In order to handle the multiplicity of annotations, a number of methods to combine them in a single reference ground truth have been proposed in computer vision [189–192]. In such a scenario, the descriptor considered as ground truth may be, for instance, the intersection of all annotations, or the mean of all annotations. Flexer proposed to use the inter-annotator disagreement to estimate the upper limit of performance that any algorithm can possibly reach on a given dataset, and applied it to the task of music similarity estimation [180]. All aforementioned studies suggest that the provision of multiple annotations can bring an extra depth to the interpretation of the evaluation results.

Following this idea, multiple annotations were produced for every track of the dataset. The annotation procedure was carried out by a total of 11 different professional drummers, 6 of whom received formal academic training. Each annotator has a unique and anonymous identifier number. Annotator 1 annotated the entire dataset once while the others covered portions of the dataset of various sizes, depending on the time they could commit to the task. At the end of the annotation campaign, the dataset had been

TABLE 3.1: Contents of rhythmic annotations corpora for the GTZAN dataset

	GTZAN-Rhy [186]	GTZAN-Met
Contents	Beat, Downbeat and swing 8th notes positions	All metrical levels rates
Annotators	2 Researchers/authors	11 Professional Drummers
Total number of Annotators per track	1 (2 for 5% of tracks)	2 (3 for 60% of tracks)

entirely annotated twice and a bit more than 60% of the tracks by three different annotators. The musical excerpts being 30 seconds long, it was hypothesised that they will feature a relatively stable rhythmic and metrical content so that a single annotation is provided for the whole duration of each track. After having listened to all tracks in the dataset, we can confirm that this hypothesis is verified, as the overwhelming majority of excerpts feature stable rhythmic content. The respective contents of the GTZAN-Met and GTZAN-Rhy annotation corpora is summarised in Table 3.1.

The annotation process was carried out using a web interface, so that the annotations collection was easily centralised. Before starting the annotation task, annotators were shown the interface and received instructions about the process. The annotators were presented with one track at a time and asked to annotate the pulse rate of every underlying metrical level they could hear in the music. Deciding on the depth of the metrical structure — i.e. where to set the limits of meter and hyper meter and of the shortest subdivision — was left to the annotators' judgment. They could achieve this either by filling in the BPM value directly or by tapping the adjacent button to automatically measure the tapping rate. The annotations they provided had to match the metrical hierarchy format described in Section 4.2, i.e. that the pulse rates they annotate must be related to each other in integer ratios. Annotators had the possibility to mark the absence of meter or signal an intractable metrical structure by submitting a blank annotation (i.e. no metrical levels provided). The name of the track being annotated was obfuscated, so that annotators were not influenced by any preconception about a given genre they might have. As a consequence, mislabellings listed by Sturm in [184] do not have any influence on the annotation process. The annotators were listening to the music through studio quality headphones and were allowed to listen to the tracks as many times as they

needed. The annotators were not given speed or time constraints to annotate tracks, so that the quality of annotations did not suffer from time-induced pressure. In order to prevent concentration loss and decrease of annotation quality, they were, however, limited to sessions of two consecutive hours of annotations. They could do several sessions, provided they took at least a 30 minutes break between sessions, but were moreover limited to 5 hours of annotations per day because the task is tiring. Only one annotator reached this daily upper limit. Overall, they could annotate around 40 tracks per hour on average, which is equivalent to 25 hours of continuous work to annotate the whole dataset once.

3.6 Metric Modulations Dataset

To the best of our knowledge a dataset of audio music recordings containing metric modulation with the corresponding annotations does not exist. We introduce such a dataset here. Because creating a corpus of pieces featuring specific musical features is hard and extremely time consuming, and in order to minimise personal biases, suggestions of pieces containing metric modulations have been crowdsourced. The crowdsourcing request was broadcast in a variety of channels such as academic interest group mailing lists and social media. The suggestions obtained this way were then filtered. In particular, for the purpose of the study presented in this document (cf. Chapter 7) only the pieces featuring relatively abrupt changes from a section of relatively stable metrical structure to a segment of stable but different metrical structure were kept. This way, pieces containing gradual changes such as accelerando were not selected. In addition, we recall that musical pieces with soft onsets and/or with no clear metrical structure are challenging for automatic rhythm analysis systems (cf. Chapter 5) and that the extraction of the metrical structure typically relies on lower level features such as an onset detection function. Since the creation of this dataset is aimed at enabling the evaluation of metric modulation estimation systems, only pieces with hard onsets and clear metrical structure — i.e. for which the necessary underlying rhythmic descriptors can be reliably estimated — were kept so that the evaluation carried out using it actually focus on the

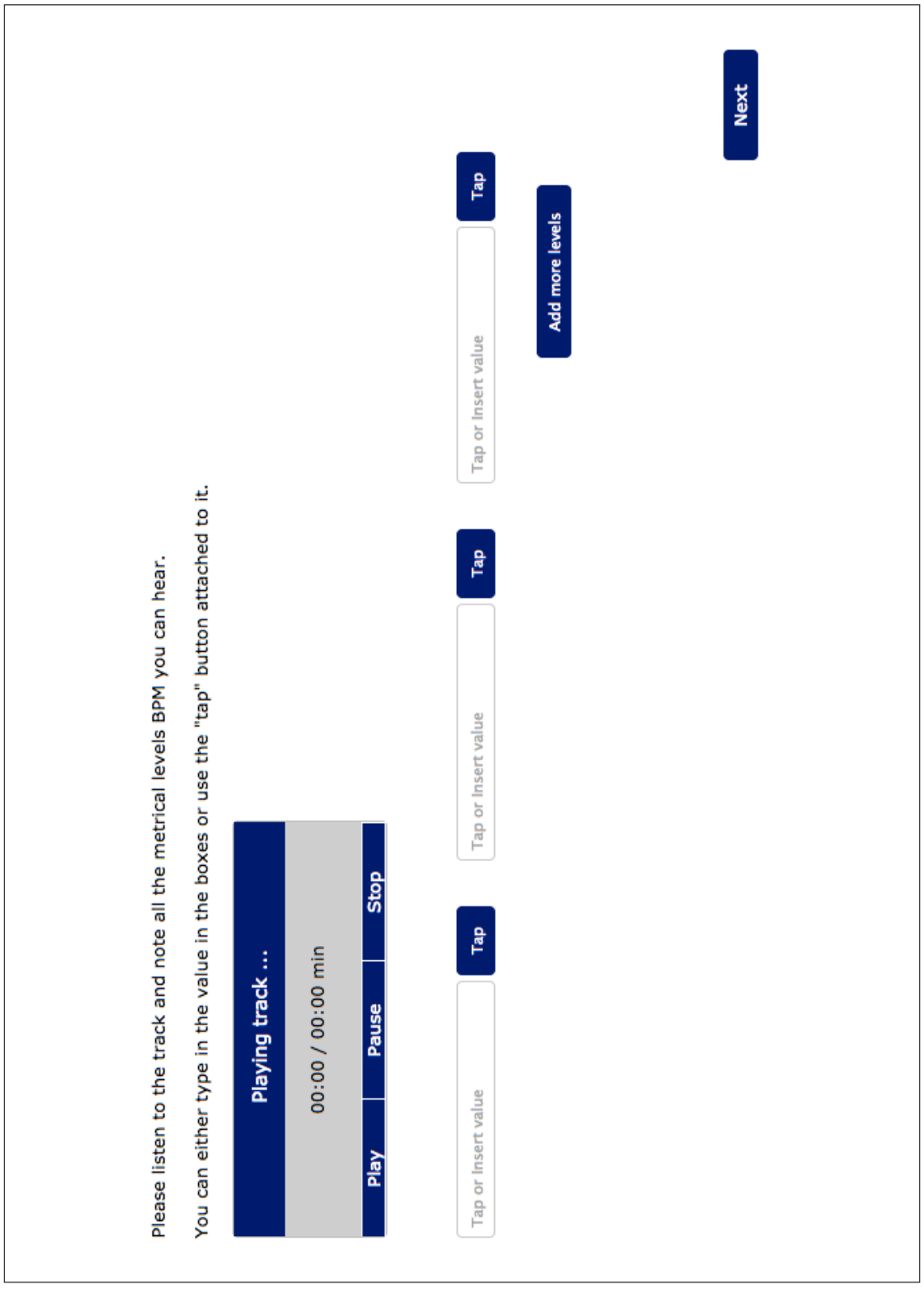


FIGURE 3.3: Metrical structure annotations collection interface

TABLE 3.2: Metric modulations dataset contents

	Avg/Track	Total
Number of tracks	-	67
Number of segments	6.6	445
Number of modulations	5.6	378

ability of a system to detect metric modulations. The complete list of tracks selected this way is given in appendix A.

Once a corpus of music pieces had been created, they were then annotated in two phases. Firstly, the beat and down beat positions were semi-automatically annotated. Beat and downbeat positions estimation was performed using the Vamp plugin implementation[10] of the algorithm presented in [121]. They were then exhaustively manually checked and corrected if necessary, using Sonic Annotator [193]. In addition, the segment boundary locations were annotated. The precise location of such a boundary may be very ambiguous. Nevertheless, in order to maximise the consistency of the annotations, the segment boundaries were annotated so that they coincide with the first down beat of the new segment.

In a second phase, all the metrical level pulse rates were annotated for each segment previously annotated. Recall that the pieces have been chosen because they contain modulations from one relatively stable metrical structure to different but stable metrical structure. As a result the metrical structure within a segment is assumed to be consistent. The metrical level pulse rates were annotated using the same procedure as for the creation of the GTZAN-Met dataset described in Section 3.5. A summary of the contents of the dataset is given in Table 3.2. A total of 67 tracks made of 445 segments were considered.

Producing multiple annotations for each track is expensive and time consuming and was not practically feasible for this dataset in the given timeframe. Therefore a rigorous assessment of the inter-annotator disagreement is not possible so far and left for future work. However, given that the metrical level pulse rates annotation procedure is identical to the one that produced the GTZAN-Met dataset, it may be assumed that inter-annotator agreement with similar properties would be observed.

[10] http://www.vamp-plugins.org/download.html

3.7 Inter-annotator agreement analysis

In this section, we present an analysis of the inter-annotator (dis)agreement that may be derived from the multiple annotations available for the GTZAN dataset. In the case of the GTZAN-Rhy corpus, the only quantitative information about inter-annotator agreement available was derived from an evaluation performed on a small subset of the dataset, as opposed to the GTZAN-Met corpus for which the information is retained on a much finer grained level, as each track received multiple annotations. As a result, most of the analysis of inter-annotator agreement will be carried out on the GTZAN-Met dataset and the GTZAN-Rhy corpus is used as a point of comparison when it is relevant. Nevertheless, the GTZAN-Rhy dataset provides rhythmic information that is not present in the GTZAN-Met corpus (e.g. swing ratio), which enables the analysis of its correlation with inter-annotator disagreement.

3.7.1 Quantifying the Inter-annotator agreement

In order to quantify the inter-annotator agreement the metrics introduced in Section 4.4.1 are used here. Then, for each track, the agreement between a pair of annotations is quantified by a F-measure score.

Figure 3.4 shows some statistics of the distribution of inter-annotator agreement per genre class for the GTZAN-Met corpus. The mean inter-annotator agreement F-measure across the entire dataset is 0.88, and is represented as a horizontal dashed line. This suggests a high average inter-annotator agreement at the corpus level. For most of the genres, the relatively narrow distribution of F-measure suggests a consistent the level of agreement. The agreement also tends to be very high with mean values often around or above 0.9 and median equal to 1.0 for 7 out of 10 genres. In comparison, in the case of the GTZAN-Rhy corpus, the authors report an inter-annotator agreement F-measure of 0.91 for beat positions. These results suggest a high level of agreement overall in both corpora. Nevertheless, significantly wider distributions and lower mean and median values characterise a significantly larger level of inter-annotator disagreement for the

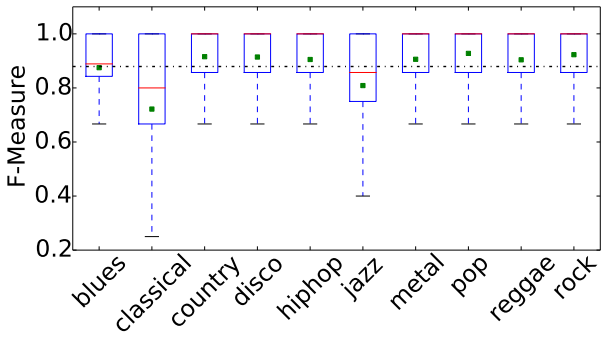


FIGURE 3.4: **Inter-annotator agreement for every genre**. The box extends from the lower to upper quartile values of the data, with a red line at the median while the green dot is the mean. The whiskers extend from the box to show the range of the data, excluding the outliers [11]. The black dashed horizontal line is the mean F-measure for the entire corpus.

tracks in the classical and jazz genre classes on the GTZAN-Met corpus. In the following sections, the cases of disagreement are investigated further.

3.7.2 Inter-annotator disagreement and metrical hierarchy

Are all the metrical levels equally likely to generate inter-annotator disagreement or are some of them more challenging to annotate than others? This is the question to be addressed in this section.

When annotating the metrical structure, it is necessary for the annotators to make a decision about the depth of the hierarchy, i.e. to decide what are the highest and lowest metrical levels. The highest level corresponds to longer periodicities and the potential ambiguity of its determination resides in the fact that several candidates might be relevant. Should the highest level correspond to one bar or a 2 bar pattern, for instance? This highest metrical level lies at the limit between the description rhythm and structural segmentation. Because it is hard to define a clear boundary between long metrical cycles and structural segmentation, it can be expected that different annotators make different judgments. In other words, it can be expected that the highest metrical level is particularly prone to inter-annotator disagreement.

[11] The upper whisker will extend to last data point less than $Q3 + 1.5 \times IQR$, where $IQR = Q3 - Q1$ is the inter-quartile range and $Q3$ and $Q1$ are the third and first quartile respectively. The lower whisker is similarly obtained with $Q3 - 1.5 \times IQR$. This configuration is used for all box plots in the remainder of this thesis

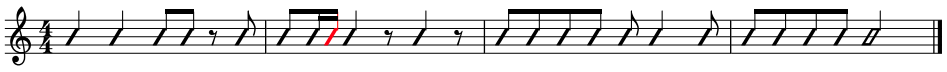


FIGURE 3.5: **Example of metrical accidental**. The red 16^{th} note only is the only note requiring the metrical structure to be extended below the 8^{th} level to describe the musical content, and it appears once: it may be considered as a metrical accidental

Similarly, the determination of the lowest level, which features the highest pulse rate, and also known as *tatum*, might be subject to some ambiguity. A similar difficulty was reported by Klapuri in the annotation of tatum [15]. Let us introduce the term *metrical accidental* to describe the difficulty with the lowest level. It is used in analogy with harmonic accidentals, in which case the occasional use of a note outside the current key (for example an F# in C major) is referred to as an accidental. Similarly, in the rhythmic counterpart, the occasional use of a subdivision that is not part of the current metrical structure is referred to as a metrical accidental. Figure 3.5 shows an example of what may be considered as a metrical accidental: the sixteenth note in the second bar, highlighted in red. It is the only instance of usage of a sixteenth note subdivision in this example, so that the metrical structure contains subdivisions up to the eighth note, and the sixteenth might be seen as an accidental. This example is relatively straightforward in that the sixteenth note is an accidental. However, in the grey area between clearly occasional and clearly non-occasional use of a subdivision, the decision is left to the annotators' judgment and can therefore be expected to be prone to disagreement.

As a consequence, we hypothesise that extreme levels (i.e. highest and lowest) are more likely to generate inter-annotator disagreement than other metrical levels. In order to test this hypothesis, all annotation pairs for which a disagreement between annotators is found are chosen. Disagreement on at least one level is characterised by an agreement F-measure inferior to 1.0. Then the position of the metrical level (relatively to the hierarchy it belongs to) resulting in the disagreement is tracked. Finally, the proportion of metrical levels leading to disagreement being either the lowest (highest pulse rate), highest (lowest pulse rate) or other levels for the entire corpus is computed. The results are presented in Table 3.3. Note that the number of "other" metrical levels varies depending on the

TABLE 3.3: Relative position of metrical levels leading to inter-annotator disagreement

Metrical Levels	Disagreement proportion
Highest level	28.7%
Lowest level	46.1%
Other levels	25.2%

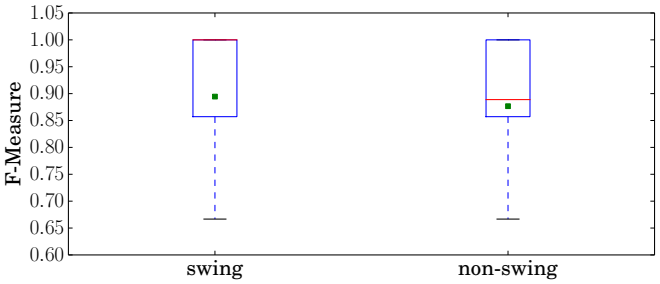


FIGURE 3.6: **Inter annotator agreement F-measure vs swing classes from [2]**.The box extends from the lower to upper quartile values of the data, with a red line at the median while the green dot is the mean. The whiskers extend from the box to show the range of the data, excluding the outliers.

tracks and annotators[12].

Nearly half of the metrical levels generating disagreement between the annotators correspond to the lowest metrical level. This result tends to support the idea that this type of disagreement relates to metrical accidentals and is consistent with Klapuri's observation [15]. Typically, in such a case one annotator might have included a high pulse rate in the metrical hierarchy when the other considered it as an accidental and did not include it. The highest level accounts for about a quarter of the disagreement cases, and the remaining 25% correspond to other levels. Overall, about 75% of the disagreement observed between annotators is relate to the extreme metrical levels.

3.7.3 On swing and inter-annotator disagreement

A duple subdivision of a time unit, e.g. the subdivision of a quarter note in two eighth notes, implies by default that the longer interval (e.g. the quarter note period) is subdivided in two parts of equal length (e.g. the eighth note period). A piece is typically labelled as '*swung*' when the subdivision of a time unit departs from the two equal parts schema, therefore generating a series of alternating long and short time intervals. The

[12]Different pieces may have different metrical hierarchy depth. In addition different annotators may have different interpretations of a given piece

swing may then be quantified by the ratio of the long to the short interval. The swing is typically viewed as an element of expressive timing, hence the swing ratio being controlled by the performers. Ratios typically vary from 1:1 to 2:1 while more extreme ratios (e.g. 3:1 and higher) are observed less often.

The model underpinning the GTZAN-Met annotations only captures isochronous pulse rates (i.e. is limited to swing ratios such of the form n:1 where $n \in \mathbb{N}$) and is thus not able to accurately capture swing. When presented a swing piece, the annotators therefore had to approximate the swung metrical level with the closest underlying isochronous metrical level. For example, let us consider swung eighth notes. Depending on the long-short ratio and the annotator judgment, they might either be approximated as straight eighth notes (long-short ratio 1:1), or triplet shuffle (long-short ratio 2:1), which implies an underlying triplet metrical level. This decision is left to the annotator's choice, and may therefore be expected to result in inter-annotator disagreement.

The swing data from the GTZAN-Rhy corpus was used in order to address the following questions: Is there more inter-annotator disagreement observed in presence of swing in the GTZAN-Met corpus? Does the swing ratio influence the agreement between annotators, and if so how?

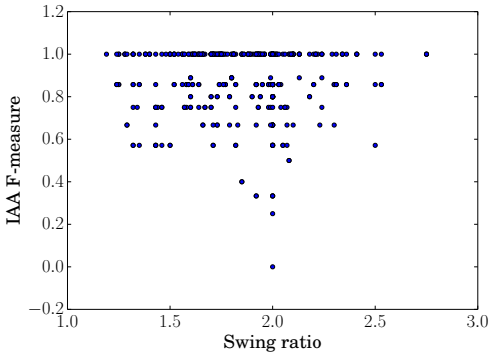


FIGURE 3.7: **Inter annotator agreement (IAA) F-measure vs. swing ratio from [2], for tracks with swing**. Each dot represents the inter-annotator agreement for one track

First, the tracks are classified as "swing" if the swing tag provided in the GTZAN-Rhy dataset is set to True and "non-swing" otherwise. Then, each track is associated with an F-measure characterising the inter-annotator agreement from the GTZAN-Met corpus.

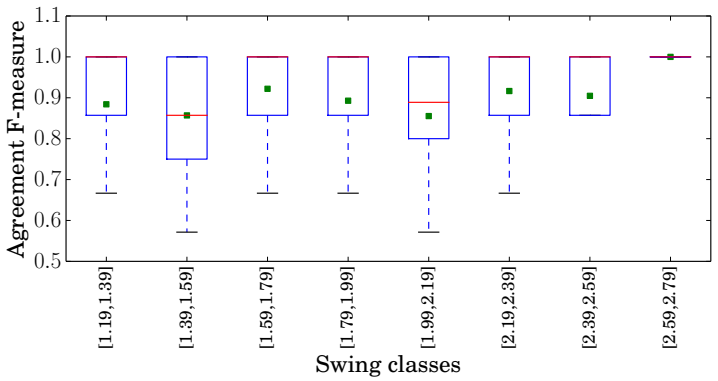


FIGURE 3.8: **Inter-annotator agreement per swing ratio class**. The box extends from the lower to upper quartile values of the data, with a red line at the median while the green dot is the mean. The whiskers extend from the box to show the range of the data, excluding the outliers. Note that [2.59,2.79] class only contains 3 scores ($F_m = 1.0$ in all cases).

Figure 3.6 shows the distribution of inter-annotator agreement for the swing and non-swing classes. The spread of the agreement F-measure distributions is comparable in the swing and non-swing classes. Although the mean F-measure in the non-swing class is slightly lower than in the swing class, the null hypothesis of equality of the mean F-measures is not rejected by a Mann-Whitney U-test (p-value = 0.16). As consequence, this classification does not provide any tangible evidence to support the idea that the presence of swing impacts the level of inter-annotator agreement.

In order to assess the influence of the swing ratio on inter-annotator disagreement, each track is associated with an F-measure characterising the corresponding inter-annotator agreement from the GTZAN-Met corpus and the swing ratio collected from the GTZAN-Rhy corpus. Figure 3.7 shows the scatter plot of the corresponding data. It does not reveal any salient trend of correlation between the swing ratio and the inter-annotator agreement. Figure 3.8 shows the same data, segmented by swing ratio classes and does not reveal any clear pattern either. Note that the [2.59,2.79] class only contains 3 scores (=1.0). In conclusion, it appears that there is no tangible evidence of a negative impact of the presence of swing nor of the swing ratio on inter-annotator agreement scores.

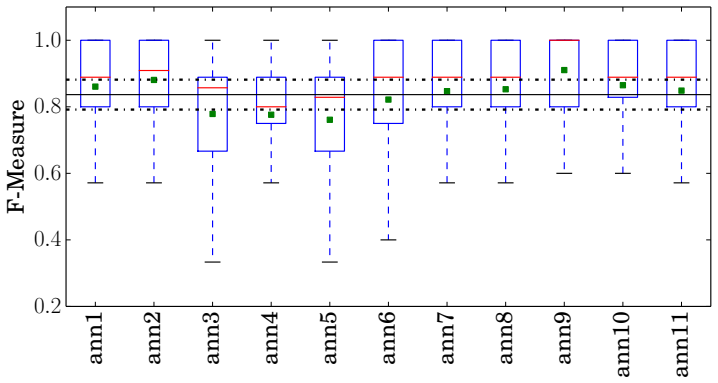


FIGURE 3.9: **Inter-annotator agreement per annotator**. The box extends from the lower to upper quartile values of the data, with a red line at the median while the green dot is the mean. The whiskers extend from the box to show the range of the data, excluding the outliers. The horizontal solid and dashed lines represent the mean and mean ± standard deviation of the mean F-measure across all annotators.

3.7.4 Annotators comparison

Being humans, it is to be expected that the annotators come with their personal bias and therefore might not all deliver the exact same performance. Then, do some annotators stand out by disagreeing with the others more than the average? The existence of such outliers is investigated in this section.

The annotators were free to chose how much time they wanted to commit to the annotation procedure. As a result, they did not all annotate the same number of tracks. While Annotator1 covered the entire dataset, the others typically annotated between 30 and 300 excerpts, presented to them in random order. For each annotator, the distribution of its agreement scores is shown in Figure 3.9. It can be observed that annotators 1,2 and 6-11 produce comparable distribution of agreement F-measure. Annotators 3-5 on the other hand seem to agree less with other annotators. In addition to the distribution of agreement F-measure per annotator, the mean and standard deviation of the mean F-measure for each annotator (the green dots in Figure 3.9) are computed. In this context, an annotator may be regarded as an outlier if his mean agreement F-measure is more than a standard deviation away from the overall mean [181], i.e. outside the zone delimited by the horizontal dashed lines in Figure 3.9. Again, Annotators 3-5 appear as outliers whose level of agreement with other annotators is notably low. Table 3.4 sums up the highest formal qualification of each annotator as well as the average speed at

TABLE 3.4: **Annotators education details and annotation speed**

Annotator	Formal Education	Annotation Speed (tracks/h)
Annotator 1	Dip	80.3
Annotator 2	-	55.6
Annotator 3	BMus	57.5
Annotator 4	-	31.6
Annotator 5	-	50.0
Annotator 6	MMus	49.8
Annotator 7	-	25.0
Annotator 8	BMus, PGDip	39.7
Annotator 9	-	25.0
Annotator 10	BMus	28.3
Annotator 11	BMus	75.3
Mean	-	40.2

which they annotated tracks. There does not appear to be any correlation between formal education nor annotation speed and the level of agreement of a given annotator with the others. It may then be hypothesised that the non-consensual annotations produced by Annotator 3-5 either correlate to attributes for which no data is available or simply result from the annotators' idiosyncrasies and personal biases.

3.7.5 Intra-corpus consistency

Some singularities of the GTZAN dataset can be leveraged to gain some insight on the quality of the annotations. In particular, we propose to use the 46 pairs or triplets of exact duplicates listed by Sturm [184] to assess the consistency of annotators; the hypothesis being that two exact duplicates annotated by the same person should ideally receive the exact same annotation.

We first detail the results of this analysis for the annotators of the GTZAN-Met corpus. For each annotator, the mean self-agreement F-measure (i.e. the mean of the F-measure obtained for each duplicate pair he/she annotated) as well as the number of duplicate pairs annotated are given in Table 3.5. Note that Annotator 1 covered the entire dataset, and therefore annotated all the duplicates. In contrast, the other annotators covered only a portion of the dataset, which was randomly presented to them. As a consequence the 10 other annotators have annotated between 0 and 7 pairs of duplicates each. The mean F-measures observed tend to be very high although for all annotators but Annotator 1 they

TABLE 3.5: **Annotators self-agreement as estimated from the exact duplicates listed by Sturm [184]**. The mean self-agreement F-measure and the number of duplicates pairs annotated are given for each annotator

Annotator	mean F_m	#pairs annotated
Annotator1	0.96	55
Annotator2	0.97	4
Annotator3	0.67	1
Annotator4	0.88	2
Annotator5	-	0
Annotator6	0.94	2
Annotator7	-	0
Annotator8	0.98	7
Annotator9	-	0
Annotator10	-	0
Annotator11	0.89	3

bear very little statistical significance given the small number of samples. Annotator 1 covered the entire dataset, spreading his workload over 8 days distributed over two weeks and the tracks were presented in random order. Given these conditions, a self-agreement F-measure of 0.96 suggests a very high level of annotation consistency.

Similarly, the exact duplicates may be leveraged to evaluate the consistency of the annotations of the GTZAN-Rhy corpus. First, the tempo annotation consistency is evaluated. The aim of this analysis is to verify that two annotations correspond to the same metrical level rather than assessing the accuracy of the annotated rate. The annotations for a duplicate pair are thus considered consistent if the two tempi are equal within an 8% tolerance, in accordance with the MIREX standard, which leaves room for some variability with respect to the rate, but guarantees that it corresponds to the same metrical level. Under this condition, only one case of inconsistency is observed between tracks *metal.00040* and *metal.00061*, which corresponds to an '*octave error*'. This then suggests a high level of consistency in the tempo annotations in this corpus.

Following the same procedure, the beat, downbeat and 8^{th} note position consistency is analysed using the MIREX standard so that two beat positions are considered consistent if they fall in a 70ms window. The F-measure for each pair of tracks is then computed and we present here the mean F-measure for all duplicates. F-measures equal to 0.69 and 0.58 are obtained for the beat and downbeat annotations respectively, which would tend to suggest that the consistency of the annotations is rather poor. This result may

TABLE 3.6: **Consistency of the annotations estimated from the exact duplicates listed by Sturm [184]**. Consistency scores are given by an F-measure with the exception of Tempo, for which a matching percentage is provided

Corpus	Annotation	non-Corrected	Corrected
GTZAN-Rhy	Tempo	98.1%	—
	Beat	0.69	0.93
	Downbeat	0.58	0.83
	8th note	0.89	0.96
GTZAN-Met	Met Structure (Ann1)	0.96	—

be surprising given that the tempo annotations are very consistent. In fact, not all the exact duplicates listed by Sturm are identical sample for sample. A large portion of the duplicate pairs exhibit a time offset of the order of tens to a couple of hundred ms between the two supposedly identical tracks. Two excerpts were defined as exact duplicates "*when two excerpts are the same to such a degree that their time-frequency fingerprints are the same*" [184]. This result is given within the time offset error marging of the fingerprinting algorithm, which is observed to be of the order of 100ms[13]. Such time offsets may not be problematic for MIR tasks that do not require a fine time resolution, but are large enough to be significant in comparison to the 70ms tolerance window used here. As a consequence, for all the duplicate pairs, the relative time offset between two excerpts was measured as the lag resulting in the maximum correlation between the two time domain waveforms. The evaluation was then run again with the time offset compensated. After correction, the overall consistency scores are significantly higher, as shown in Table 3.6, which suggests that the time offset was responsible for a large part of the inconsistencies measured in the non-corrected condition. Here again, with F-measures higher than 0.8, a high degree of consistency is observed. It may also be noted that the annotation of the downbeat seems to be less consistent than the beat level. Given that the downbeat is likely to be the highest metrical level annotated, the relatively higher inconsistency observed here is consistent with the observation that extreme metrical levels lead to greater inter-annotator disagreement reported in Section 3.7.2.

[13]Personal correspondance with B.Sturm

TABLE 3.7: **Tempo mismatches between the GTZAN-Rhy and GTZAN-Met corpora**. In the *All Ann* condition, all the annotators of the GTZAN-Met corpus disagree with the tempo annotation in the GTZAN-Rhy corpus. In the *Partial mismatch* condition, there is at least one annotator from the GTZAN-Met corpus agreeing with the annotation of the GTZAN-Rhy corpus and at least one disagreeing.

	All Ann		Partial mismatch	
Genre	Count	Ratio (%)	Count	Ratio (%)
Blues	2	7.7	8	14.8
Classical	12	46.2	16	29.6
Country	0	0.0	1	1.9
Disco	1	3.8	2	3.7
Hiphop	0	0.0	1	1.9
Jazz	5	19.2	9	16.7
Metal	3	11.5	3	5.6
Pop	1	3.8	4	7.4
Reggae	1	3.8	8	14.8
Rock	1	3.8	2	3.8
TOTAL	26	100	54	100

3.7.6 Inter-corpus consistency

In this section, we investigate the consistency of redundant annotations across the two corpora. The GTZAN-Rhy corpus contains annotations of the tempo as well as of the beat positions. The tempo rate is also expected to correspond to a metrical level rate in the GTZAN-Met annotations. Therefore, the tempo annotation is redundant across these two corpora and offers an ideal point of comparison.

First, for each track of the dataset, we verified if the tempo annotation from the GTZAN-Rhy corpus corresponds to a metrical level rate annotation in the GTZAN-Met corpus, allowing a tolerance of $\pm$ 8% of the tempo value. It results in a 95.2% match rate, which reveals a very high level of consistency across the corpora.

In a second step, the cases of tempo annotation mismatch (4.8%) were analysed. The distribution of these cases per genre class under two conditions are shown in Table 3.7. For a given track, in the *Partial mismatch* condition, some annotations of the GTZAN-Met corpus match the GTZAN-Rhy tempo and some others do not[14], while in the *All ann* condition, all the annotations of the GTZAN-Met corpus result in a mismatch with the tempo annotation of the GTZAN-Rhy corpus. Once again, it is observed that Jazz

[14]This necessarily implies a disagreement between the annotators of the GTZAN-Met corpus

TABLE 3.8: **Lists of tracks to use with care.** Tracks labelled as *intractable* and *divergent* were not successfully annotated by human experts. Tracks are labelled as *Suspicious* when all the annotators of the GTZAN-Met corpus agree between each other and disagree with the GTZAN-Rhy tempo annotation.

Intractable	Divergent	Suspicious
blues.00032	blues.00032	pop.00011
jazz.00026	jazz.00026	disco.00047
jazz.00030	jazz.00030	jazz.00019
jazz.00003	jazz.00003	rock.00014
reggae.00086	reggae.00086	blues.00038
classical.00080	classical.00080	classical.00057
classical.00038	classical.00038	classical.00047
classical.00033	classical.00033	classical.00042
classical.00040	classical.00040	classical.00056
classical.00041	classical.00041	classical.00037
classical.00055	jazz.00066	classical.00007
classical.00077		classical.00028
classical.00036		metal.00080
classical.00067		metal.00096
		metal.00092

and Classical stand out as the genres that generate the two highest mismatch rates in both conditions.

The mismatches in the *All ann* condition highlight consistent (and therefore interesting) discrepancies between the two corpora: all annotations in the GTZAN-Met corpus are in disagreement with the tempo annotation of the GTZAN-Rhy corpus. These cases of consistent disagreement were further divided in two sub-conditions whether there was a high or low inter-annotator agreement score (i.e. F-measure>(resp. <) 0.5) in the GTZAN-Met corpus. In the first sub-condition, all the GTZAN-Met annotators strongly agree against the GTZAN-Rhy annotation. This condition is realised for a total of 15 tracks, for which the GTZAN-Rhy tempo annotation is labelled *suspicious*. We then listened to all the corresponding tracks, which confirmed the suspiciousness of the annotations: the annotated tempo did not seem correct to us, not even when allowing for octave errors. In the second sub-condition, all GTZAN-Met annotators disagree with the GTZAN-Rhy annotation but also disagree between each other. It is likely that such divergent annotations are generated by musical content that is difficult to annotate. This condition is realised for 11 tracks labelled as *divergent*.

In addition, the posture adopted towards tracks which are difficult to annotate was

significantly different in the two annotation campaigns. In the case of the GTZAN-Rhy all tracks have received an annotation, whereas the annotators of the GTZAN-Met corpus had the possibility to provide a blank annotation to signify that they could not annotate a track, either because it is too difficult or because providing an annotation is not relevant (e.g. absence of pulse). Tracks are listed as *intractable* if at least one annotator decided not to provide an annotation. Table 3.8 provides a list of all the tracks mentioned in this section. In the process of creating a challenging dataset for beat tracking, Holzapfel *et al.* noted that including tracks that even humans cannot annotate is not helpful in designing better algorithms, and only kept tracks that are challenging for algorithms but successfully annotated by humans [3]. In a similar fashion, the tracks listed here as *intractable* or *divergent* are not successfully annotated by human experts and may therefore be used with care, if not disregarded, in algorithm evaluations.

3.8 Conclusions

All the datasets used for the experiments reported in this thesis were briefly described in this chapter. Two of these, namely the GTZAN-Met and the Metric Modulations Dataset, are original contributions. In addition, we have presented an analysis of rhythm annotations for the GTZAN dataset, as provided by two recently published corpora. Overall, the results show, first of all, that the annotations provided are of high quality, as the high annotation consistency and high inter-annotator agreement suggest. Similarly, a high level of consistency of annotations was revealed by the measurement of annotators self-agreement allowed by leveraging the presence of duplicates in the GTZAN audio dataset. Nevertheless the cases of annotation inconsistencies or of inter-annotator disagreement revealed interesting properties of the annotations.

It has been shown that the extreme metrical levels (i.e. highest and lowest) account for 75% of the cases of inter-annotator disagreement, which suggest that the depth of the metrical hierarchy is hard to determine unequivocally. In addition, it appeared that Jazz and Classical genres result in significantly more inter-annotator disagreement than other genres. The Jazz genre is known for its use of swing and syncopation. Comparing the

inter-annotator agreement measured on the GTZAN-Met corpus with the swing data from the GTZAN-Rhy corpus, no tangible evidence that either the presence of swing or the swing ratio has a systematic impact on inter-annotator agreement was found. The absence of data on syncopation did not enable the evaluation its effect.

The comparison of the inter-annotator agreement scores for each annotator showed that 8 out of 11 annotators tend to exhibit a similar behaviour while 3 others appear as relative outliers by agreeing less consistently and less on average with other annotators. No correlation between these relative discrepancies and available data, such as the annotators' formal education and annotation speed, was found. It may then be hypothesised that the less consensual behaviours observed either correlate with attributes for which no data is available or simply stem from the annotators' personal idiosyncrasies.

Finally, from the analysis of the inconsistency of annotations over the GTZAN-Met and GTZAN-Rhy corpora, a lists of tracks — labelled *intractable* and *divergent* — for which producing annotations is either irrelevant, impossible or at least very difficult was derived. In addition, the tracks for which the validity of the tempo annotations of the GTZAN-Rhy corpus seem to be strongly challenged are listed and labelled as *suspicious*. As such, these lists specify tracks which should be used with care for the evaluation of automatic algorithms. The fact that a large proportion of the *intractable* and *divergent* tracks belong to the Classical genre relates to a conclusion that was also drawn from the analysis of inter-annotator disagreement: it can be suspected that a model of metrical structure built on the notion of a relatively steady underlying pulse, such as the one considered here, may not be applicable to such music pieces. The elaboration of an alternative model of metrical structure which does not rely on the existence of a steady underlying pulse and which would be applicable to the currently intractable pieces is still an open question, which most likely constitutes a promising avenue for future research.

Chapter 4

On the Explicit Extraction of Metrical Structure From the Beat Spectrum

4.1 Introduction

Two main classes of approaches for metrical structure estimation have been identified in Chapter 2. Owing to their suitability for application in an unsupervised scheme[1], we focus in this chapter on periodicities estimation methods that rely on time-frequency transforms of the onset detection function. Typical frequency transforms are the Fourier Transform (FT) or the Auto-Correlation Function (ACF), which are used to compute beat spectra and rhythmograms (cf. Section 2.3). It is commonly observed that metrical level pulse rates relate to salient periodicities in the beat spectrum [43, 93, 194]. But it is also easily shown that not all the salient periodicities in a beat spectra relate to metrical level pulse rates (cf. Section 2.3), which is understandably an obstacle to the explicit metrical structure extraction from such features. Peeters proposed to multiply the ACF and Fourier rhythmograms to eliminate periodicities that do not relate to metrical levels pulse rates from the beat spectrum [93]. It was then qualitatively observed that the

[1] as opposed to cycle tracking methods that typically function in a supervised fashion

salient periodicities in the resulting spectrum indeed match more closely the metrical level pulse rates. However the use of this feature did not surpass the use of Fourier beat spectrum in a rhythm classification task [84]. To our knowledge, an explicit evaluation of the relationship between salient periodicities in the beat spectra and the metrical level pulse rates has never been carried out. This leads us to ask: To which extent do the peaks in the periodicity spectrum correspond to metrical level pulse rates? To which extent is it possible to explicitly extract the metrical levels pulse rates from a beat spectrum?

In this chapter we aim at addressing these questions. Carrying out an explicit evaluation requires the existence of reference data to compare the outputs of an algorithm to. Publicly available datasets containing metrical structure-related annotations typically do not contain annotations for more than two or three metrical levels while typical metrical structures commonly consist of 4 to 5 metrical levels. As a consequence, the evaluation presented here relies on the GTZAN-Met dataset, which was created in response to the lack of data at this level of detail and presented in Section 3.5.

In a first experiment, we quantify the match between peaks in the periodicity spectrum and metrical level pulse rates as well as the effect of the rhythmograms combination proposed by Peeters [93]. Since the metrical structure features a hierarchical organisation of the metrical level pulse rates (cf. Section 2.4), we propose a simple algorithm to estimate the metrical structure from the beat spectra by enforcing hierarchical constraints and evaluate its performance.

It has been shown that inter-annotator disagreement sets an upper bound on the performance possibly achievable by an algorithm [180]. In a second experiment we investigate how the performance of the algorithm relates to human experts (dis)agreement, as assessed on the GTZAN-Met dataset in Section 3.7. Finally, given that the beat rate is a metrical level, it ought to be part of the metrical pulse rates extracted using the aforementioned algorithm. On this premise, we derive a simple tempo extraction algorithm from the metrical levels pulse rates estimation system. It was submitted to the MIREX 2014 tempo estimation task and gave competitive performance.

4.2 Formalising the metrical structure representation

In this section we specify some notation used in the remainder of this chapter for the description of the metrical structure, based on the formal description that was provided in Section 2.1. We then relate this notation to its musical interpretation.

4.2.1 Notation

Figure 4.1 shows a hierarchical representation of several examples of metrical structure. It is similar to the dots representation of Figure 2.1 with the addition of explicit hierarchical links. Each horizontal level of nodes on the tree accounts for one metrical level of index $i \in [1, L]$, which is associated with a pulse rate ω_i measured in BPM (Beats Per Minute). The number of metrical levels necessary to represent the full hierarchy of a piece of music is therefore L. We define the sequence of metrical level pulse rates, sorted in ascending order as:

$$F = \langle \omega_1, \cdots, \omega_L \rangle \tag{4.1}$$

Hierarchical relationships are defined by $\lambda_i \in \mathbb{N}$, the ratio between the pulse rates of a metrical level and the next one. This is represented in Figure 4.1 by number of child nodes that each node generates. The sequence of metrical levels pulse rate ratios is then defined as:

$$\Lambda = \langle \lambda_1, \cdots, \lambda_i, \cdots, \lambda_{L-1} \rangle \tag{4.2}$$

where $\forall i \in [1, L-1], \lambda_i = \frac{\omega_{i+1}}{\omega_i}$. As a result, Λ is a representation of the hierarchical relationships between the layers of the metrical structure and is therefore independent of the absolute value of the metrical levels pulse rates, i.e. independent of tempo. This representation is also independent of the semantic role (e.g. beat, downbeat etc.) attributed to each metrical level.

Reconstructing a metrical level pulse rates sequence F from hierarchical sequence Λ is trivial and requires the provision of only one pulse rate. For instance, given the lowest pulse rate ω_0, the entire sequence is reconstructed with $\omega_i = \omega_1 \cdot \prod_{k=2}^{i} \lambda_k$.

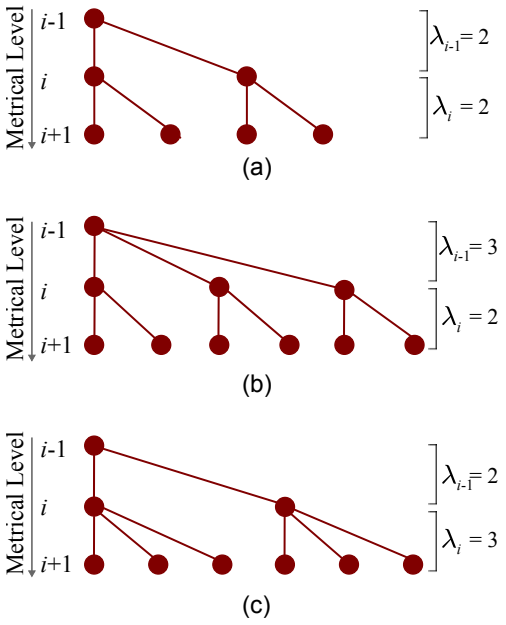


FIGURE 4.1: Tree representation for metrical hierarchy. (a) A simple duple hierarchy dividing the lower level into two groups of two. (b) A simple triple hierarchy dividing the lower level into three groups of two. (c) A compound-duple hierarchy dividing the lower level into two groups of three.

4.2.2 Relation to musical concepts

The metrical structure descriptors used here do relate to more traditional concepts, typically derived for score-based musicology, such as time signatures and note values, but there is no isomorphic mapping between the two domains. In particular the score representation of a given metrical structure is not necessarily unique. Figure 4.2 illustrates this fact by giving two different score notations for the same pattern. The time signature, tempo markings and therefore the note values used are different. However, the underlying organisation of musical events is the same in both cases, because they represent the same piece of music, which result in an identical metrical hierarchy (cf. Figure 4.3). Time signature is sometimes used as a proxy to characterise and classify metrical structure (see for example [43]) but the examples just provided illustrate the fact that the time signature alone does not fully define the metrical structure. The time signature typically specifies a canonical organisation for a part of the metrical structure. Let us consider the $\frac{12}{8}$ time signature as an example. It specifies that each bar is made of 4 beats and that the beat is subdivided in three equal part. Other elements, such as the potential subdivision of the eighth notes, are not specified by the time signature. Similarly, the

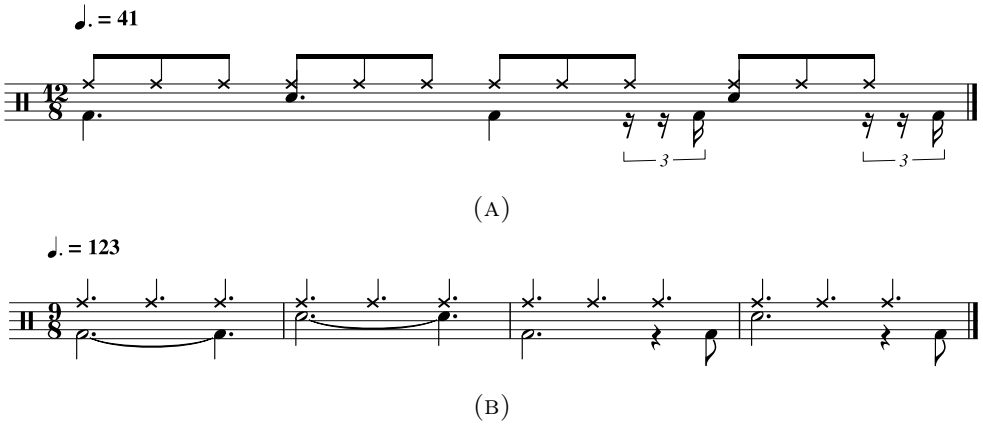


FIGURE 4.2: Two alternative score notations for the same drums pattern. The transcription is inspired from John Mayer's "Gravity".

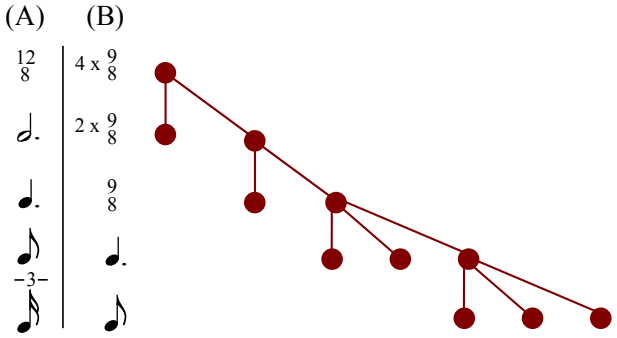


FIGURE 4.3: Metrical level pulse rates hierarchical structure for John Mayer's Gravity, with the two corresponding notation options given in Figure 4.2. Only one branch of the metrical pulse rates hierarchichy is developed for clarity.

$\frac{9}{8}$ time signature specifies that each bar is made of 3 beats, which are then subdivided in three equal parts but does not specify further subdivision levels. As a result, using score notation, the metrical structure can be fully specified only by the joint provision of a time signature *and* the note values used. This nuance explains how two different time signature can represent the same metrical structure.

On the other hand, the hierarchical organisation of the underlying metrical level pulse rates is independent of the score notation. In fact, this is an interesting property of this type of representation of the metrical structure of music. What is an ambiguous case from a score point of view maps to a unique representation. Note that this type of hierarchical organisation is related to the well-formedness and preference rules of the GTTM, which are briefly summarised in Section 2.1.1. Figure 4.3 illustrates such a structure on the musical example presented in Figure 4.2. For clarity, only the child nodes of one branch of each level are depicted — we refer the reader to Figure 4.1 for examples

of full development of the tree structure. In the example of Figure 4.2 and Figure 4.3, the hierarchical structure may be described by $F = \langle 10.25, 20.5, 41, 123, 369 \rangle$ and $\Lambda = \langle 2, 2, 3, 3 \rangle$. We also note that if the 16^{th} notes in case (A) (which correspond to the eighth notes in case (B)) were removed from the pattern, the depth of the metrical structure tree would be reduced by removing the last level (bottom of the tree in Figure 4.3), so that the hierarchical structure may be described by $F = \langle 10.25, 20.5, 41, 123 \rangle$ and $\Lambda = \langle 2, 2, 3 \rangle$. In other words, the use of subdivisions and the length of the bars or patterns directly affects the depth of the metrical structure tree.

Given a hierarchical organisation of metrical levels, the choice of the association of metrical levels with semantic roles (e.g. beat, downbeat etc.) leads to radically different score representations, as illustrated in Figure 4.3. In case (A), the metrical level of pulse rate 41 BPM is regarded as the beat, which leads to a $\frac{12}{8}$ time signature in which the eighth notes are subdivided in 16^{th} note triplets, whereas in case (B), considering the 123 BPM pulse rate as the beat rate leads to a pattern that runs over four $\frac{9}{8}$ bars with very long note durations. As a consequence, producing a time signature estimate from audio recordings requires an estimate of the periodicities present in the music in order to form a model of the metrical structure, and a semantic interpretation of this structure. In this chapter, we only investigate problems relating to the estimation of the hierarchical structure of metrical level pulse rates and do not consider the semantic interpretation.

4.2.3 Limitations

Describing the metrical structure via metrical level pulse rates is effectively a frequency domain representation. Implicitly, it assumes isochronous pulse rates, because a frequency cannot be defined otherwise. Such a representation is therefore not fit to fully describe non-isochronous elements. Music in odd time is often used as an example of non-isochronous metrical structure. Figure 4.4 shows a simple example of rhythm with a non-isochronous accent pattern.

In the example, the accenting pattern implies a 3+2 grouping of eighth notes as it groups together 3 and then 2 eighth notes, therefore creating a non-isochronous structure. The

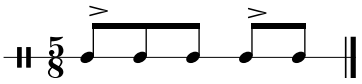


FIGURE 4.4: Example of odd time signature with non-isochronous accent pattern

hierarchical structure of isochronous metrical level pulse rates cannot describe the non-isochronous groupings organisation. Nevertheless it is important to note that the model can still account for a meter "in five" as the bar is subdivided in 5 equal parts (eighth notes), i.e. the ratio of pulse rates between two metrical levels equals 5. In other words, a reduced representation of non-isochronous metres is possible in this model. All the isochronous subdivisions are captured, only the phase of the non-isochronous grouping is lost (in the 5/8 example, the model captures the subdivision of the bar in 5 equal parts but cannot distinguish 3+2 from 2+3 groupings).

4.3 Periodicity Spectrum and Metrical Structure Estimation

In this section we first detail the computation of the periodicity spectra used in this chapter and the remainder of this thesis. Secondly, we present a simple algorithm to extract metrical level pulse rates from the periodicity spectrum. A flowchart summarising the structure of the algorithm is given in Figure 4.5. It can be broken down into three main processing steps. First, an onset detection function is computed from audio using the *superflux* method described in Section 2.2. Then, the analysis of the periodicities present in the musical signal is performed via the computation of rhythmograms. Finally, the metrical structure is estimated by peak-picking the periodicity spectrum, with the hypothesis that some of the peaks will correspond to metrical level rates.

4.3.1 Periodicity analysis

The autocorrelation function (ACF) based and Fourier transform-based rhythmograms are considered as features on which to base the periodicity analysis, notated $\mathbf{R}_A$ and $\mathbf{R}_F$ respectively. We refer to Section 2.3 for details of the computation of such features.

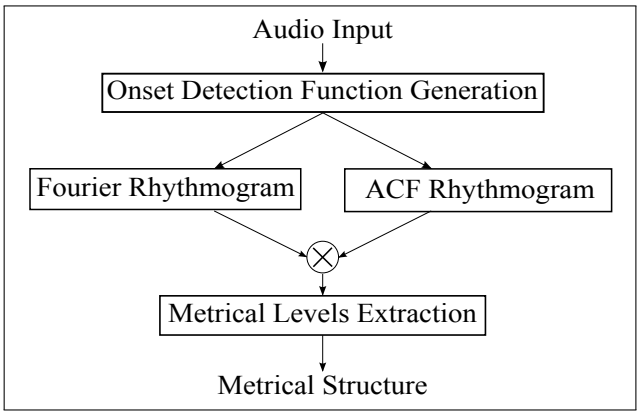


FIGURE 4.5: The feature extraction algorithm is divided in three major steps: computing an onset detection function, performing a periodicity analysis by combining two rhythmograms and finally extracting the metrical structure from the resulting periodicity spectrum.

All rhythmograms are computed using 12s Hann windows and 0.36s hop size. Using such long windows allow for the capture of long periodicities (e.g. at the bar level) at the expense of the inability to accurately capture changes happening at a scale typically shorter than the window length.

In addition, we consider the approach introduced by Peeters [93], which we briefly summarise in the following. Two rhythmograms $\mathbf{R}_A$ and $\mathbf{R}_F$ are calculated in parallel (i.e. using the same onset detection function). By construction, the Fourier transform generates harmonics at integer multiples of the periodicities present in the signal whereas the autocorrelation function generates subharmonics. The strategy proposed by Peeters consists in multiplying element-wise the Fourier transform and the autocorrelation function so that the harmonics and sub harmonics are cancelled by zero entries in the other periodicity spectrum, assuming that the peaks that are left (which must appear in the two functions) correspond to metrical level pulse rates.

The autocorrelation functions provides an analysis of periodicities against a lag l, so that the autocorrelation-based rhythmogram is natively a function of time and lag $\mathbf{R}_A(l, n)$. In order to be able to multiply the Fourier transform-based and autocorrelation-based periodicity spectra, it is necessary to rescale the two rhythmogram to a common frequency scale. In particular, the autocorrelation periodicity spectra (i.e. the frames of $\mathbf{R}_A$) are mapped to a frequency scale with $l = f_s/\omega$ where l is the lag in the ACF, and ω is the corresponding frequency and f_s is the onset detection function sampling frequency.

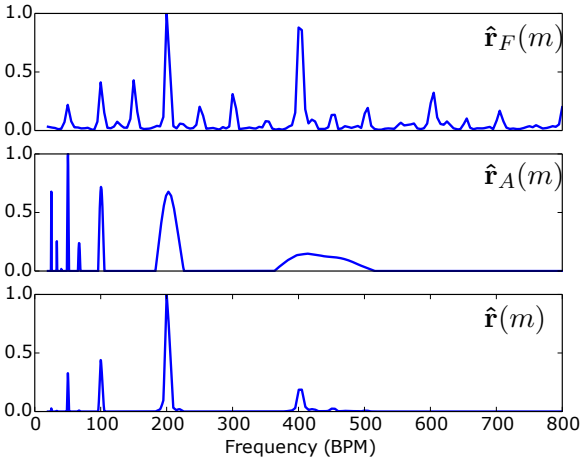


FIGURE 4.6: **Example periodicity spectra for the track blues.00053**. Respectively from top to bottom, Fourier transform based, $\hat{\mathbf{r}}_F(m)$, autocorrelation function based, $\hat{\mathbf{r}}_A(m)$ and the result of their multiplication, $\hat{\mathbf{r}}(m)$. Most of the harmonics in the Fourier and ACF spectra are rejected from $\hat{\mathbf{r}}(m)$.

The ACF is resampled so that the frequency scale matches the FFT bins (alternatively, both the FFT and ACF spectra can be resampled to a common frequency scale).

Given that the dataset used to carry the evaluation is made of short excerpts (cf. Section 3.3), the metrical structure is assumed to be constant. Therefore, the Fourier transform and frequency mapped autocorrelation function based rhythmograms, $\mathbf{R}_F(m,n)$ and $\mathbf{R}_A(m,n)$ respectively, can be summarised in average spectra $\hat{\mathbf{r}}_F(m)$ and $\hat{\mathbf{r}}_A(m)$ by summing frames:

$$\begin{aligned} \hat{\mathbf{r}}_F(m) &= \sum_n \mathbf{R}_F(m,n) \\ \hat{\mathbf{r}}_A(m) &= \sum_n \mathbf{R}_A(m,n) \end{aligned} \tag{4.3}$$

A composite spectrum $\hat{\mathbf{r}}(m)$ is produced by calculating the element-wise multiplication, or Hadamard product, denoted as $\odot$, of the spectra $\hat{\mathbf{r}}_F(m)$ and $\hat{\mathbf{r}}_A(m)$, and normalising the result:

$$\hat{\mathbf{r}}(m) = \frac{\hat{\mathbf{r}}_A(m) \odot \hat{\mathbf{r}}_F(m)}{\max_m \left(\hat{\mathbf{r}}_A(m) \odot \hat{\mathbf{r}}_F(m) \right)} \tag{4.4}$$

The three spectrum $\hat{\mathbf{r}}_F(m)$, $\hat{\mathbf{r}}_A(m)$ and $\hat{\mathbf{r}}(m)$ computed for a track from the GTZAN dataset are shown in Figure 4.6 in order to illustrate the properties discussed here. It

is clear that only the peaks that appear in the two periodicity functions are preserved in $\hat{\mathbf{r}}(m)$, while most of the others are eliminated by the multiplication with a coefficient close (or equal) to zero in the other spectrum.

In the current formulation, a summarised description of the periodicity content is used at a track level, as realised by equation (4.3). This is adequate for use cases in which consistency of metrical structure can be assumed, as it is the case here and in the original studies carried out by Peeters [84, 93]. However, this assumption may not hold in a more general setting. In particular, we will come back to this issue in Chapter 7 where we extend Peeters' multiplicative strategy to every frame of the rhythmograms in order to track the evolution of metrical structure over time.

4.3.2 Peak-picking algorithm

One hypothesis to be tested here is that metrical levels are represented by salient periodicities in the onset detection function, and are therefore revealed as peaks in the spectrum $\hat{\mathbf{r}}(m)$. If this is true, peak-picking the spectrum $\hat{\mathbf{r}}(m)$ should be enough to retrieve all the metrical level pulse rates. However, empirical experience has suggested this hypothesis might not always hold and Section 2.3 provides a theoretical explanation as to why this is the case. As a consequence, we devise three peak-picking steps to test this hypothesis.

First, a simple algorithm detecting local maxima if an element is larger than both of its neighbours is employed to find all the peaks in $\hat{\mathbf{r}}(m)$. Only the peaks higher than a given threshold $\epsilon_{\hat{\mathbf{r}}}$ are kept, i.e.

$$\hat{\mathbf{r}}(m_{peak}) > \epsilon_{\hat{\mathbf{r}}} \tag{4.5}$$

The threshold is set small ($\epsilon_{\hat{\mathbf{r}}} = 0.005$) so that at this stage all the peaks present in $\hat{\mathbf{r}}(m)$ are detected.

Secondly, from this list of peaks, the one with the largest magnitude is selected and its corresponding rate $\omega_{\max}$

$$\omega_{\max} = \underset{m}{\operatorname{argmax}} \left(\hat{\mathbf{r}}(m)\right) \tag{4.6}$$

is used as a reference. For instance, $\omega_{\max}$ is located around 200BPM in the example of Figure 4.6. It is hypothesised that metrical level pulse rates result in salient periodicities in the onset detection function, and thereby in salient peaks in the periodicity spectrum. Then it is to be expected that the corresponding peaks contain more energy than the related harmonics. $\omega_{\max}$ represents the rate containing the most energy in the spectrum, and is therefore assumed to represent a salient metrical level. Picking the most energetic rate minimises the likelihood of deriving $\omega_{\max}$ from a spurious peak and therefore maximises the robustness of the system in that respect. By construction of the metrical structure model used here, the rates of all other metrical levels ω_j should be related to $\omega_{\max}$ by integer ratios (cf. Section 4.2). Then, the pulse rate ω_j of the j^{th} peak in $\hat{\mathbf{r}}(m)$ is compared to $\omega_{\max}$ and is kept as a metrical level candidate if, and only if, it satisfies one of the following conditions:

$$\exists n \in \mathbb{N} : \begin{cases} \frac{\omega_j}{\omega_{\max}} \approx n & \text{if} \quad \omega_j > \omega_{\max} \\ \frac{\omega_{\max}}{\omega_j} \approx n & \text{if} \quad \omega_j < \omega_{\max} \end{cases} \tag{4.7}$$

and rejected otherwise. This operation effectively filters out peaks that cannot possibly be part of a metrical structure that obeys the constraints established earlier. It is labelled as Peak Filtering and referred to as 'PF' in the following.

Finally, a third peak-picking step based on a Peak-Picking Kernel labelled 'PPK' is introduced. Given the hierarchical description of the metrical structure provided in Section 2.1.1, it comes that finding all the peaks that are integer multiples of $\omega_{\max}$ using the peak filtering step PF is not sufficient to guarantee that they constitute a well-formed structure. In particular, the pulse rate of each metrical level and its immediate neighbour must be related by an integer ratio λ_i too. This peak-picking step, PPK, aims at guaranteeing that it is the case. Starting with $\omega_{\max}$, the relationship of each metrical level with its two closest neighbours are compared. The process is iteratively repeated until the list of peaks is exhausted. The comparison is performed both in ascending and descending pulse rates. Several candidate metrical structures can be tracked in order to deal with ambiguous cases, depending on the result of the comparisons at each step. At the end of the procedure, the candidates are weighted, and the heaviest is chosen. Algorithm 1 describes the details of the procedure for the ascending pulse rate search

in pseudo-code. The descending search algorithm is easily derived by symmetry and provided in pseudo-code in Appendix B.

Algorithm 1 Peak-picking kernel: $\mathcal{K}(\omega_j, \mathcal{M})$

Require: ω_j is the level under analysis and $\mathcal{M}$, the metrical structure candidates

1: **while** $\frac{\omega_{j+1}}{\omega_j} \notin \mathbb{N}$ **do**
2: $\quad\omega_{j+1} \leftarrow \omega_{j+2}$
3: $\omega_q \leftarrow \omega_{j+1}$
4: **if** $\frac{\omega_{q+1}}{\omega_j} \in \mathbb{N}$ **then**
5: $\quad$**if** $\frac{\omega_{q+1}}{\omega_q} \notin \mathbb{N}$ **then**
6: $\quad\quad\mathcal{M}_1 \leftarrow \mathcal{M}$
7: $\quad\quad\mathcal{M}_2 \leftarrow \mathcal{M}$
8: $\quad\quad\omega_j \leftarrow \omega_q$
9: $\quad\quad(\omega_j, \mathcal{M}_1) \leftarrow \mathcal{K}(\omega_j, \mathcal{M}_1)$ ▷ call peak-picking kernel
10: $\quad\quad\mathcal{M} \leftarrow \{\mathcal{M}, \mathcal{M}_1\}$
11: $\quad\quad\omega_j \leftarrow \omega_{q+1}$
12: $\quad\quad(\omega_j, \mathcal{M}_2) \leftarrow \mathcal{K}(\omega_j, \mathcal{M}_2)$ ▷ call peak-picking kernel
13: $\quad\quad\mathcal{M} \leftarrow \{\mathcal{M}, \mathcal{M}_2\}$
14: $\quad$**else**
15: $\quad\quad$append ω_{j+1} to $\mathcal{M}$
16: $\quad\quad\omega_j \leftarrow \omega_{j+1}$
17: **else**
18: $\quad$append ω_{j+1} to $\mathcal{M}$
19: $\quad\omega_j \leftarrow \omega_{j+1}$
return ω_j, $\mathcal{M}$

Lines 1 to 3 filter out metrical level candidates not related in integer ratio to ω_j. Once an integer ratio $\frac{\omega_q}{\omega_j}$ with $q > j$ is found, the second nearest neighbour ω_{q+1} is taken in account. A special case occurs when $\frac{\omega_{q+1}}{\omega_j}$ is an integer ratio but $\frac{\omega_{q+1}}{\omega_q}$ is not. This means that the metrical level ω_j could equally be subdivided in levels ω_q or ω_{q+1} whereas level ω_{q+1} is not a subdivision of level ω_q. This is in contradiction with the hierarchical model of metrical structure (cf. WFR1 and WRF2 in Section 2.1.1). It is therefore considered that levels ω_q and ω_{q+1} cannot coexist in a single metrical hierarchy. In such a situation, two hierarchy candidates are generated (lines 6 and 7) and constructed independently by calling two new instances of the peak-picking kernel (lines 9 to 13), one including metrical level ω_q and the other including level ω_{q+1}. If this special case is not encountered — which means that all neighbouring metrical level rates are related by integer ratios — ω_{j+1} is appended to the metrical structure, the index of level under analysis is incremented (lines 17 and 21), and the procedure is repeated until all peaks have been analysed.

At the end of this peak-picking stage, metrical structure candidates have been generated, and are represented by their sequence of metrical level pulse rates F. Note that if the condition of line 5 in algorithm 1 is never entered, only one candidate is generated. Finally, for each hierarchy candidate, each one of the metrical level pulse rate ω_i is associated with a weight $u_i = \hat{\mathbf{r}}(\omega_i)$ stored in $U = \langle u_1, \cdots, u_L \rangle$. Each hierarchy candidate is graded by the sum of the weights of its metrical levels $\Theta = \sum_{i=1}^{L} u_i$. The metrical structure with the largest cumulated weight Θ is considered as the most salient, and is therefore chosen as the final estimate. Note that here we only consider a scenario where a single metrical structure estimate is considered. However, our proposed model has the capability to produce several estimates and weight them. This can be interpreted as a mean to capture metrical ambiguity. Though this is beyond the scope of this work, one could imagine a scenario in which the metrical ambiguity could be evaluated by comparing the relative weights of the metrical structure candidates.

4.4 Evaluation

4.4.1 Evaluation metrics

For each track of the dataset, a pairwise comparison of every level pulse rate of the metrical hierarchy from the reference annotation (A) and the estimated feature (E) is performed. Let L_A and L_E be the depth of the metrical hierarchy (i.e. the number of metrical levels) of the reference annotation and estimated feature respectively. The goal of this comparison is to establish if a metrical level pulse rate in the reference annotation matches an estimated rate and reciprocally. In order to do so, a binary matrix $\mathbf{M}$ of size $L_A \times L_E$ capturing the matching information between extracted feature and annotation is built with each element $\mathbf{M}_{i,j}$ defined as:

$$\mathbf{M}_{i,j} = \begin{cases} 1 & \text{if } \left|\log_{10}\left(\omega_i^A\right) - \log_{10}\left(\omega_j^E\right)\right| < \xi \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

Consequently, each match between an annotation and an extracted metrical level is associated with the value 1 while mismatches are associated with 0. A tolerance ξ is

applied to account for the variability of human rating; its value is set to allow a tolerance window of 15% of the annotated value. Note that the intent here is not to evaluate the accuracy of the pulse rate annotation but rather to evaluate if a pair of annotated and estimated pulse rate correspond to the same metrical level. As a consequence the tolerance is set large enough to allow for imprecise rate estimation, yet small enough so that two unrelated pulse rates are distinguishable.

In this context, a false negative is characterised by a row of zeros in the matrix $\mathbf{M}$ because they correspond to levels being present in the annotation but not in the extracted feature. Likewise, a false positive is characterised by a column of zeros. The number of true positives is obtained by summing all the coefficients $\mathbf{M}_{i,j}$ of the matrix. Standard information retrieval metrics are then computed. For each track, Precision, Recall and F-measure as defined in Section 2.10.3 are calculated, therefore measuring the ability of the system to retrieve a metrical structure that corresponds to the reference annotations on each track. Average values of these scores across all tracks of the dataset are then calculated.

An example of such metrics is given below. It corresponds to the evaluation of the extracted metrical structure against one annotation for the track *rock.00029* from the GTZAN dataset.

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In this case, there are four true positives, i.e. four levels matching, indicated by the ones, one false negative indicated by the last row of zeros and no false positive as there is no column of zeros. It results in Precision=1.0, Recall=0.80 and F-measure=0.89.

4.4.2 Evaluation Dataset

The GTZAN audio dataset and the corresponding metrical structure annotations from the GTZAN-Met dataset are used to carry out evaluations in this chapter. Although

we refer the reader to Chapter 3 for a detailed description of the dataset and reference annotations, we recall that each one of the 1000 audio tracks of the dataset has been annotated by two to three different annotators, resulting in more than 2600 annotations. It is then possible to assess the inter-annotator (dis)agreement using the metrics described in Section 4.4.1. Instead of comparing annotation data (A) and an extracted feature (E), annotations produced by one annotator are compared with annotations produced by another. The F-measure is used as a figure of merit to assess the agreement for each track, 1 meaning perfect agreement (annotators have annotated a structure that contains exactly the same metrical levels) and 0 meaning complete disagreement (nothing in common in their annotations). By doing so, it comes that the average inter-annotator agreement F-measure for the entire dataset is 0.88, therefore setting the average upper limit of algorithm performance [180]. In the following, for each track, the estimated feature is evaluated against all the annotations available; from which are calculated average value per track and the dataset average values presented below. The tracks labelled as "*intractable*" were excluded from the evaluation (cf. Section 3.7.6).

4.4.3 Baseline method

Lartillot proposed a method to extract all the metrical level pulse rates from an ACF rhythmogram [194], which we also include in our evaluation. We use the implementation of the *mirmetre()* function from the Mirtoolbox[2]. The metrical structure estimation proposed in [194] comprises three steps that are very similar to the ones in the method presented here. First of all, an onset detection function is processed using a spectral flux method. Secondly an analysis of the periodicities present in this onset detection curve is performed by calculating an ACF rhythmogram (labeled "*autocorrelogram*" in the original publication). Finally, the metrical structure is estimated from the ACF rhythmogram. This last step is performed by peak-picking the rhythmogram frames and filtering the peaks on the basis of heuristic rules. In our experiment, we set the window length and hop size identical to the values used for the algorithm described in Section 4.3. All other parameters were set to default values. The metrical structure is returned

[2]Version 1.6.1 used in our experiments.
Available at: https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

in the form of a list of metrical level pulse rates. For each metrical level rate, the average value for the entire duration of the track is used for the evaluation.

4.4.4 Experiments

In a first experiment, the correspondence between salient periodicities in rhythmograms and metrical level pulse rates is assessed. This is achieved by evaluating the methods described above with respect to their ability to retrieve metrical level pulse rates.

In particular, three peak-picking steps are proposed in the algorithm presented in Section 4.3, canonically applied sequentially. The ability of the system to retrieve metrical level pulse rates is evaluated under different peak-picking conditions. Starting with only the raw peak-picking step, the evaluation of the algorithm is repeated, activating an extra layer of peak-picking step each time (i.e. adding PF and then PPK). Using only the first layer of peak-picking, which simply detects all the peaks in a periodicity spectrum, to perform estimation of metrical level pulse rates therefore allows quantification of the correspondence between peaks in the periodicity spectrum and metrical level pulse rates. Repeating the evaluation with the addition of extra layers of peak-picking then quantifies how the introduction of hierarchical constraints impacts the performance of metrical level pulse rates estimation.

In addition, we seek to evaluate the effectiveness of the multiplication of the periodicity spectra proposed by Peeters. The evaluation is thus performed using the ACF spectrum $\hat{\mathbf{r}}_A$ as well as the product spectrum $\hat{\mathbf{r}}$ under the same peak-picking conditions. Using the ACF spectrum only also enables comparison with the Lartillot's baseline method [194].

In a second experiment the algorithms performance is compared to the inter-annotator (dis)agreement. In particular, leveraging the multiple annotations available for this dataset, the algorithms performance is compared to the upper limit set by the inter-annotator disagreement [195]. For this purpose, we compute the Performance Relative to the Upper Limit (PRUL) implied by the inter-annotator disagreement as:

$$\text{PRUL} = 100 \cdot \frac{Fm_E}{Fm_A} \tag{4.9}$$

TABLE 4.1: System configurations ('methods') under evaluation defined by three parameters: the periodicity spectrum used 'PS', the activation of the Peak Filtering step 'PF', and the activation of the Peak-Picking Kernel 'PPK'. Results are presented for each method as well as for the baseline method [194] as Precision, Recall, F-measure and Performance Relative to the Upper Limit (PRUL) scores.

Method	PS	PF	PPK	Precision	Recall	F-measure	PRUL
Method 1	$\hat{\mathbf{r}}$	on	on	0.83	0.84	**0.82**	**93.2%**
Method 2	$\hat{\mathbf{r}}$	on	off	0.61	0.86	0.68	77.3%
Method 3	$\hat{\mathbf{r}}$	off	off	0.51	**0.96**	0.64	72.7%
Method 4	$\hat{\mathbf{r}}_A$	on	on	**0.86**	0.77	0.80	90.9%
Method 5	$\hat{\mathbf{r}}_A$	on	off	0.70	0.79	0.72	81.8%
Method 6	$\hat{\mathbf{r}}_A$	off	off	0.43	0.95	0.58	65.9%
Lartillot [194]	-	-	-	0.36	0.55	0.43	48.9%

where Fm_E is the average F-measure obtained for the automatically estimated metrical level pulse rates on one track and Fm_A is the average F-measure representing the agreement between all the annotators for the same track [3]. We recall that every track of the dataset was annotated by two to three different annotators. For each track, the metrical structure estimate produced by a given algorithm is compared to all the annotations available for this track, so that several F-measure scores are produced. We then present the average F-measure (Fm_E), precision and recall over all the annotations available for the track. Similarly, the inter-annotator agreement is evaluated by comparison of all the pairwise combinations for a track. This results in one F-measure score if there are two annotations (only one pair combination possible) or three F-measure score if there are three annotations. The average score over all the combinations is then computed for each track. On top of an average for the entire dataset, we present average values per genre cluster.

4.5 Results and discussion

For all methods under evaluation, only the metrical level rates in the range 30-800BPM were considered for evaluation. The 30BPM lower limit is chosen because periodicity spectra (in particular $\hat{\mathbf{r}}_F$) tend to be very noisy in the 0-25BPM range. The upper limit

[3] We have shown in Chapter 3 that some metrical levels are more prone to inter-annotator disagreement than others. However, it has also been shown that any level of the metrical structure can be subject to disagreement. In this context Fm_A represents an average inter-annotator agreement, irrespective of the position of metrical levels in the metrical structure.

is set to 800BPM in order to be greater than the fastest rate observed in the annotations of dataset. We recall from Chapter 3 that the average inter-annotator agreement F-measure for the whole dataset is $Fm_A = 0.88$. As a result and unless stated otherwise the PRUL for average scores at the dataset level is computed with respect to this value.

4.5.1 Metrical level pulse rates vs. salient periodicity

Results of the evaluation of the automatic metrical structure extraction for all methods considered in this chapter are presented in Table 4.1 as average precision, recall and F-measure scores for the entire dataset.

Methods 3 and 6 both have the PF and PPK steps deactivated so that only the first raw peak-picking step is active (cf. Section 4.3). The evaluation of method 3 and 6 provide an assessment metrical information captured by $\hat{\mathbf{r}}$ and $\hat{\mathbf{r}}_A$ respectively. Methods 3 and 6 exhibit similar performance in terms of recall with very high scores (0.96 and 0.95 respectively). This confirms the hypothesis that the metrical level pulse rates are captured as peaks in the periodicity spectra. In addition, method 3 scores higher than method 6 in terms of precision. Once again this result validates the assumption that irrelevant peaks would be rejected by the multiplication of $\hat{\mathbf{r}}_A$ and $\hat{\mathbf{r}}_F$. However, the rather low precision (0.51) also demonstrates that $\hat{\mathbf{r}}$ does not only contain peaks relating to metrical level pulse rates. In other words, the multiplication of periodicity spectra improves the correspondence between peaks in the periodicity spectra and metrical level pulse rates, but is not sufficient to eliminate all peaks that do not relate to a metrical level pulse rate. This last observation motivates the introduction of more elaborate peak-picking strategies.

Comparison of methods 2 and 3 reveals that the peak filtering step PF only brings a small improvement in F-measure, and therefore is not sufficient to extract a meaningful metrical structure on its own. Comparing the results of Method 1 and 2 clearly shows that constraining the peak-picking algorithm with a musically meaningful model for metrical structure (via the activation of step PPK) results in a substantial increase in performance (0.14 points of F-measure score). This is primarily achieved by increasing precision score at the expense of a very small decrease of recall, which means that the

PPK step effectively helps picking peaks that correspond to metrical level rates with a very little rate of error. Relating the F-measure to the average inter-annotator agreement, a PRUL of 93.2% is obtained for method 1, which suggest that the algorithm estimates closely approach human expert judgment on average. A similar trend emerges from comparison of methods 4, 5 and 6.

Lartillot's baseline method should be compared with methods 4, 5 and 6, as they all use ACF to estimate periodicities of the onset detection function. In all cases, this method is outperformed. Given that the onset detection function and periodicity estimation used in the baseline method are similar to the algorithm proposed here, it is assumed that the difference resides mostly in the metrical structure estimation steps — i.e. in the peak-picking strategy. As a consequence, the results corroborates the idea that peak-picking the periodicity spectra is a difficult and sensitive, yet crucial step. Lartillot's peak picking is achieved using heuristics that are not strongly rooted in music theory whereas our constraining of the metrical structure estimation with a musicologically motivated model proves to be instrumental in achieving a better level of agreement with human experts. Method 1, which involves the use of all the processing stages, delivers the best overall performance, with the highest F-measure.

From these results, the conclusions that can be drawn are threefold. Firstly, they suggest that the hypothesis according to which metrical level pulse rates are materialised by peaks in the periodicity spectrum is validated. On the other hand, it clearly appears that not all peaks in the periodicity spectrum relate to metrical level pulse rates. Secondly, the rhythmogram multiplication strategy proposed by Peeters appears to be effective to preserve the peaks that relate to metrical level pulse rates while eliminating some of the peaks that do not. The elimination is not total, however. Finally, a peak-picking strategy enforcing musically relevant hierarchical constraints (materialised by steps PF and PPK) is beneficial to perform accurate metrical structure extraction from the periodicity spectrum.

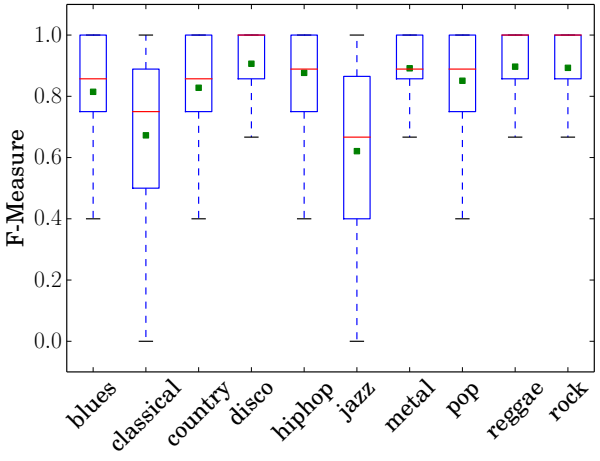


FIGURE 4.7: **Algorithm performance for every genre**. The box extends from the lower to upper quartile values of the data, with a red line at the median while the green dot is the mean. The whiskers extend from the box to show the range of the data, excluding the outliers.

4.5.2 Genre classes and inter-annotator agreement

The GTZAN dataset was initially introduced to evaluate genre classification algorithms [183]. Taking advantage of the genre class annotation of the dataset, we first present the evaluation results per genre class. The corresponding F-measure distributions are shown in Figure 4.7. Jazz and Classical genres stand out as they exhibit a lower mean and much wider distribution of F-measure than other genres, which suggests that the algorithm performs significantly worse on the tracks in these genre classes. However, it has been shown in Chapter 3 that inter-annotator agreement also depends on genre classes. We refer to Figure 4.7 and Figure 3.4 for comparison of the F-measure distributions. These two genres put aside, the inter-annotator agreement F-measure distribution is more consistent across genres than algorithm performance F-measure distribution is. So far, only the average PRUL for the entire dataset has been computed. We extend here the evaluation to the computation of the average PRUL for every genre class as it may be expected that its value varies across genres. The results are shown in Table 4.2.

At the exception of Jazz, the PRUL is over 90% for all genres. This result suggests that the algorithm performs with a good consistency in comparison to expert annotations, for all genres except Jazz. In other words, tracks belonging to the Jazz genre class seem to have properties that make them challenging for the algorithm more than for humans.

TABLE 4.2: F-measure of algorithm performance, inter-rated agreement and Algorithm Performance Relative to the Upper Limit (PRUL) imposed by inter-annotator agreement; by genre classes.

Genre	Fm_A	Fm_E	PRUL
Blues	0.87	0.81	93.1 %
Classical	0.72	0.66	90.9 %
Country	0.92	0.83	90.4 %
Disco	0.91	0.91	99.1 %
HipHop	0.91	0.88	96.8 %
Jazz	0.81	0.61	75.6 %
Metal	0.91	0.89	98.4 %
Pop	0.93	0.85	91.7 %
Reggae	0.90	0.90	99.2 %
Rock	0.92	0.89	96.8 %

The automatic algorithm used here relies on the premise that metrical level pulse rates correlate with salient periodicities in the onset detection function. This assumption seem to hold for most of the genres classes in this dataset, but is possibly challenged in the case of Jazz. Several hypotheses may be formulated to explain the relatively low performance obtained for tracks in this class. Jazz is known for making use of a sizeable amount of syncopation. Syncopation being produced by the use of rhythms that contradict the established meter [196], it is possible that more syncopation leads to less salient periodicities relating to the metrical structure and therefore to lower performance. Syncopation may also explain the relatively higher rate of inter-annotator disagreement on Jazz excerpts. Additionally, Jazz typically does not have a backbeat nor a very strong emphasis on the beats and patterns of strong and weak beats. Again, this is likely to diminish the salience of meter-related periodicities in an onset detection function and therefore explain the relatively lower performance of the algorithm on tracks belonging to the jazz class. It is probable that the lack of backbeat and more generally the fact that metrical structure is not necessarily clearly marked by percussive accents in Jazz makes its analysis harder for automated systems than for humans, which would explain the lower PRUL. However, the metadata available for this dataset does not allow these hypotheses to be directly tested, so we leave this investigation for future work.

The evaluation results on the tracks belonging to the classical genre class reveal another interesting case. Although, in absolute values, Classical displays the worst inter-annotator agreement score and the second worst algorithm performance, its PRUL is still above 90%, therefore comparable with Country and not far from the one obtained for pop. This result indicates that estimating metrical level pulse rates for these tracks is difficult for both human experts and the algorithm. Soft onsets are a known difficulty for automatic rhythm feature extraction from audio in the case of classical music [3]. By listening to the tracks and the annotators feedback, not only does this observation hold but it also appears that a number of tracks in the Classical class do not have a clear pulse and/or feature very strong use of expressive timing. By relying on the notion of pulse rates, the formal description of the metrical structure used here implicitly assumes the existence of a relatively stable and salient pulse. Furthermore, by relying on a onset detection function, the algorithm also implicitly assumes that musical events onsets are captured as peaks in this function. But these fundamental assumptions are contradicted by the musical properties that can be observed in a number of tracks in the Classical class, which explains why the algorithm as well as humans are unable to provide a satisfactory analysis of these pieces within this framework. This suggests such difficulties possibly cannot be overcome by simply improving the capabilities of an algorithm relying on such premises but would rather require the use of a different formal model of metrical structure on which to build an algorithm. The definition of such a formal framework is still an open question, however.

4.5.3 Alternative metrics

Previously, the metrical structure extracted for each track has been compared with all the annotations available for this track, therefore producing several F-measures which were then averaged to produce a unique score per track. By doing so, the variance in the scores for each tracks is disregarded. In this section, we introduce alternative metrics for further evaluation of the algorithm performance in order to assess how good is the best agreement as well as how bad is the worst. Two evaluation conditions are considered and respectively labelled 'Best' and 'Worst'. In the 'Best' evaluation condition, for each audio track, the

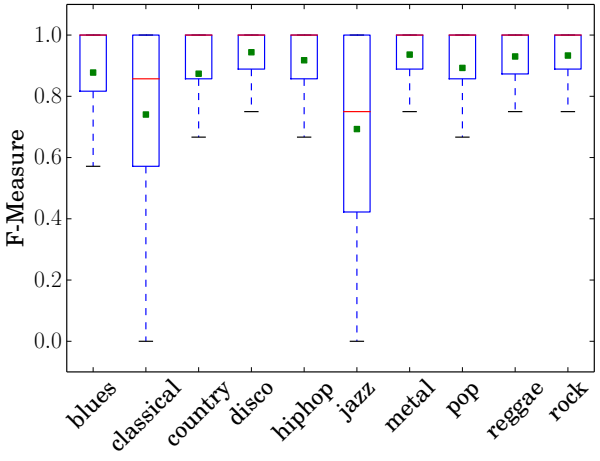


FIGURE 4.8: **Algorithm performance for every genre in the 'Best' condition**. The box extends from the lower to upper quartile values of the data, with a red line at the median while the green dot is the mean. The whiskers extend from the box to show the range of the data, excluding the outliers.

metrical hierarchy extracted by the algorithm is evaluated against all the annotations available for this track (exactly as it is done in Section 4.5.2). Then, for each track, only the highest F-measure is kept. As a result, only the best agreement between annotated and extracted metrical hierarchies is considered in this evaluation condition. Similarly, in the 'Worst' condition, only the lowest F-measure is kept, therefore considering only the worst agreement between annotated and extracted metrical hierarchies. The statistics of the results of the evaluation under these two conditions are shown, per genre cluster, in Figure 4.8 and Figure 4.9. As expected by construction, the 'Best' condition exhibits much higher performance than the 'Worst' condition and the average results shown in Figure 4.7 lie in between. Regardless of the evaluation conditions, Jazz and Classical remain performance outliers. This corroborates the observation made in Section 4.5.2 and the conclusions that can be drawn from them. In the 'Best' condition, Jazz and Classical genres aside, the median F-measure equals 1.0 and the mean in consistently well above 0.8 for all genres. Moreover the score distributions are very narrow, which accounts for a high level of consistency. This means that the estimated metrical structure always corresponds very well with at least one of the annotations, which suggests that the algorithm robustly produces meaningful estimates. In the 'Worst' condition, the distribution of F-measure scores tends to widen and the median and average values decrease with respect to the 'Best' condition or the average scores considered in Section

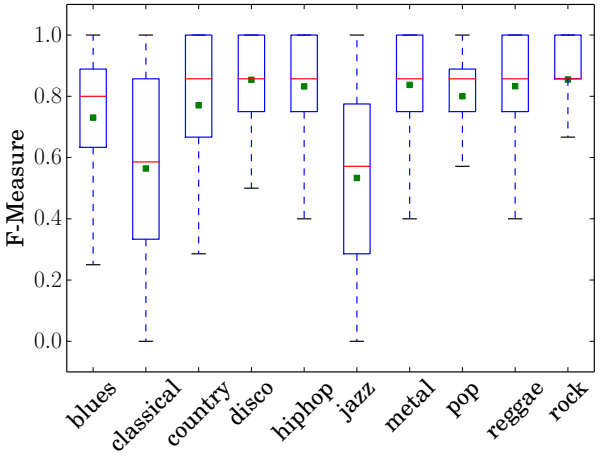


FIGURE 4.9: **Algorithm performance for every genre in the 'Worst' condition**. The box extends from the lower to upper quartile values of the data, with a red line at the median while the green dot is the mean. The whiskers extend from the box to show the range of the data, excluding the outliers.

4.5.2. However, it is interesting to note that the scores obtained in the 'Worst' condition are not dramatically low, which indirectly reveals the relatively good agreement between annotators.

Table 4.3 and Table 4.4 show the mean F-measure (Fm_E) obtained from the evaluation in the 'Best' and 'Worst' conditions respectively as well as the mean F-measure of inter-annotator agreement (Fm_A) for each genre. The calculation of Fm_A and of the PRUL are identical to those presented in Section 4.5.2. In the 'Best' condition, the PRUL exhibits values higher than 100%. It means that the best agreement between one annotation and the extracted metrical structure is greater that the average agreement between annotators, or $Fm_E > Fm_A$. This result shows the importance of taking into account the inter-annotator agreement. Evaluating the algorithm under the 'Best' condition effectively neglects the impact of the disagreement and therefore produces values that are not representative of the actual performance of the algorithm.

4.6 Extension to tempo estimation

The concepts of *tempo*, *tactus* and *beat rate* are often used interchangeably in MIR literature focusing on beat tracking and tempo estimation, although the limits of such a

TABLE 4.3: F-measure of algorithm performance in the 'Best' condition, inter-rated agreement and Algorithm Performance Relative to the Upper Limit (PRUL) imposed by inter-annotator agreement; by genre classes.

Genre	Fm_A	Fm_E	PRUL
Blues	0.87	0.88	101.1%
Classical	0.72	0.74	102.8%
Country	0.92	0.87	94.6 %
Disco	0.91	0.94	103.3%
HipHop	0.91	0.92	101.1%
Jazz	0.81	0.69	85.1 %
Metal	0.91	0.94	103.3%
Pop	0.93	0.89	95.7 %
Reggae	0.90	0.93	103.3%
Rock	0.92	0.93	101.1%

TABLE 4.4: F-measure of algorithm performance in the 'Worst' condition, inter-rated agreement and Algorithm Performance Relative to the Upper Limit (PRUL) imposed by inter-annotator agreement; by genre classes.

Genre	Fm_A	Fm_E	PRUL
Blues	0.87	0.73	83.9 %
Classical	0.72	0.56	77.8 %
Country	0.92	0.77	83.7 %
Disco	0.91	0.85	93.4 %
HipHop	0.91	0.83	91.2 %
Jazz	0.81	0.53	65.4 %
Metal	0.91	0.84	92.3 %
Pop	0.93	0.80	86.0 %
Reggae	0.90	0.83	92.2 %
Rock	0.92	0.85	92.4 %

view point have been pointed out [44]. Despite the intrinsic interest of this discussion, debating the difference between those terms is beyond the scope of this study. In the following we comply with the standard convention used in MIR literature and thus adopt the consideration that the beat rate describes the pulse rate of the metrical level labelled as 'beat' — which is therefore expected to be part of the metrical structure. We also adopt the consideration that the beat rate can be labelled as 'tempo'. Finally, we adopt the idea that listeners would tap their foot along the music at a given metrical level that can be labelled as the 'beat', therefore associating the tactus and the beat rate.

On this premise, a simple tempo estimation system based on the metrical structure estimation algorithm described above that was submitted to the MIREX 2014 tempo

estimation task is presented in this section. It effectively consists in selecting two metrical level pulse rates from the metrical structure estimate as tempo estimates. In the following, we start by briefly presenting the MIREX audio tempo estimation task, then detail the tempo estimation algorithm and finally discuss the results.

4.6.1 The MIREX audio tempo estimation task

The MIREX audio tempo estimation task[4] aims at providing a reference evaluation framework for audio tempo estimation systems and therefore enabling meaningful comparisons of their performance.

The test set consists of 160 music tracks of various genre, instrumentation, tempo and meter. A subset of 20 tracks is made public for participants to tune their algorithms while the remaining 140 are kept secret so that they are unseen by the participants' algorithms at evaluation time. The challenge organisers specify that the musical excerpts used for testing have the following properties:

- Stable tempo within each excerpt
- A good distribution of tempo across excerpts
- A large variety of instrumentation and beat strengths (with and without percussion)
- A variation of musical styles, including many non-western styles
- The presence of non-binary meters (about 20% have a ternary element and there are a few examples with odd or changing meter)

The dataset is annotated with the "*perceived tempo*", which effectively means that the ground truth data was gathered by asking a group of listeners to tap along each track — thereby associating the *tempo* to the *tactus* rate. In the interest of consistency of nomenclature and conciseness, we will keep on using the term *tempo* in this section. Ambiguity arises as listeners may tap at different rates [21], and consequently the data

[4] http://www.music-ir.org/mirex/wiki/2016:Audio_Tempo_Estimation

forms a probability density distribution across the possible tapping rates. In order to account for the non singularity of tapping behaviour captured, the annotation data for each track is made of two tempo values and their relative strength. Effectively, the two most salient peaks in the probability distribution of tapped rates are chosen as the two tempi, notated T_1 and T_2, and their relative weight is derived from the number of people tapping to each one of them and then normalised to sum to 1.0. The more people tap to a given perceptual tempo, the larger its weight is.

Algorithms receive three scores, in percentages, as evaluation metrics: 'Both tempi correct', 'At least one tempo correct' and 'Tempo P-Score'. While the two first ones are self-explanatory, P-Score calculation is given by :

$$\text{P-score} = ST_1 \cdot TT_1 + (1 - ST_1) \cdot TT_2 \tag{4.10}$$

where TT_1 is the relative perceptual strength (given by groundtruth data, varies from 0 to 1.0) of T_1, TT_1 is the ability of the algorithm to identify T_1 to within 8%, and TT_2 is the ability of the algorithm to identify T_2 to within 8%. Participating algorithms are therefore expected to output two tempo candidates T_1 and T_2. Evaluating a tempo estimation system against two ground truth values instead of one provides a finer granularity in the analysis of the results, and has been implemented in response to the well known *octave error*. Typically, an algorithm producing an octave error output would be less penalised than one producing a totally uncorrelated value.

4.6.2 Proposed algorithm

The algorithm relies on the assumption that the two ground truth tempi correspond to metrical level pulse rates. As a result, the algorithm we propose here aims at selecting the two tempi amongst the metrical level pulse rate estimates produced by the algorithm described in Section 4.3. In particular the metrical levels are filtered on the basis on a set of rules described below. This processing stage is referred to as 'Rule based filtering' block in the flowchart of Figure 4.10, where the dashed line box represents the metrical

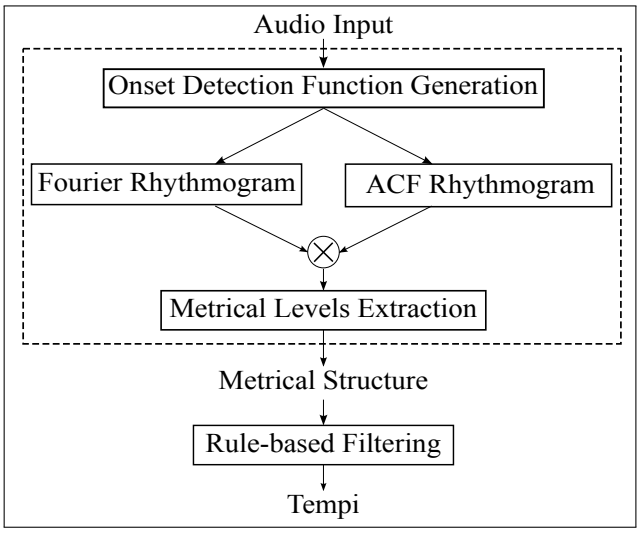


FIGURE 4.10: Audio tempo estimation algorithm flowchart

structure extraction algorithm presented in Section 4.3 and which flowchart was given in Figure 4.5.

The filter uses the parametrised resonance curve proposed to fit the perceived tempo tapped by listeners in [21]:

$$E_{\omega_0,\eta}(\omega) = \frac{1}{\sqrt{\left(\omega_0^2 - \omega^2\right)^2 + \eta \cdot \omega^2}} + \frac{1}{\sqrt{\omega_0^4 + \omega^4}} \tag{4.11}$$

where ω_0 is the resonant tempo, η the damping constant and ω the tempo (i.e. metrical level pulse rate). McKinney and Moelants carried out experiments on two group of listeners: musicians and non-musicians. They then obtained two sets of parameter to fit the results: $\omega_0 = 138$ BPM and $\eta = 5.0$ for musicians and $\omega_0 = 125$ BPM and $\eta = 2.0$ for non-musicians. In our implementation, we use the average values over these two groups for ω_0 and η. The corresponding curve is reproduced in Figure 4.11. It represents the tempo induction likelihood; in other words how likely, or comfortable listeners are tapping at a given rate. The curve tapers at its extremities, which accounts for the unlikeliness of listeners to tap tempo typically below 50 BPM and above 200 BPM. As a consequence, the tempi are limited to the 30-230 BPM range in the present algorithm.

The selection of the two successful candidates among metrical levels extracted at the previous stage is achieved using the following rationale: First of all, the number of

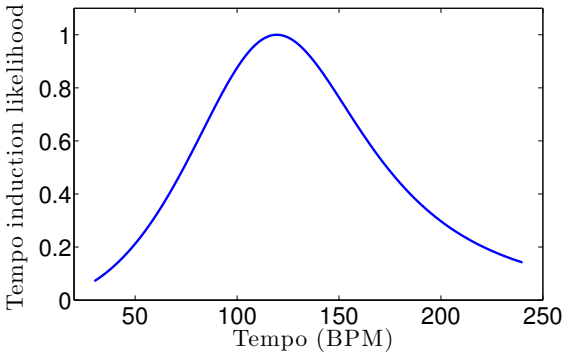


FIGURE 4.11: Tempo induction resonance curve from [21]

metrical levels present in range 30-230 BPM is calculated. Secondly, depending on that number, the appropriate processing is performed, with four possible cases:

- There are exactly two metrical level pulse rates in the 30-230 BPM range: they are chosen as the tempo estimates and their weights W_1 and W_2 are set equal to the corresponding weights in vector U.

- There are more than two metrical level pulse rates in the 30-230 BPM range: The level with the heaviest weight (from weight vector U) and the heaviest first adjacent metrical level (either greater or smaller pulse rate) are chosen as the two estimates[5]. Their weights W_1 and W_2 are set equal to the corresponding weights in vector U.

- There is only one metrical level pulse rate in the 30-230 BPM range: it is set as the first tempo estimate T_1 and W_1 is set equal to its corresponding weight in vector U. Two candidates are then generated from the first estimate. One with double frequency $T_a = 2T_1$ and, one with half frequency $T_b = T_1/2$. Both are weighted using the normalised resonance curve of equation (4.11) , so that $W_a = E_{\omega_0,\eta}(T_a)$ and $W_b = E_{\omega_0,\eta}(T_b)$, and the heaviest one is chosen as the second estimate so that $W_2 = \max(W_a, W_b)$ and T_2 is set equal to the corresponding metrical level pulse rate.

[5]We choose the heaviest first adjacent metrical level rather than the second heaviest level, because we assume that the two annotated tempi correspond to adjacent metrical levels. This assumption is motivated by the annotation procedure of the MIREX dataset: annotators were asked to tap along the music. We therefore assume that it is more likely that annotators would latch on two adjacent metrical levels rather than on two potentially far apart levels.

- There is no metrical level pulse rate present in the 30-230 BPM range: The heaviest metrical level present outside the range is taken as a reference and its frequency is iteratively divided by two until two candidates in the range are found if it is above the higher bound of the range (resp. multiplied if it is below the lower bound). Tempo estimates are then both weighted using the normalised curve of equation (4.11).

The two latter cases are implemented to enable recovery from any possible failure of the previous processing stage (i.e. that would fail at detecting some metrical levels). In that type of scenario, the tempo estimate is an artificial guess that is made at the expense of a major bias: simple duple meter is assumed, hence the factor of two in the rules above. This assumption is made because the music corpus used in the MIREX audio tempo estimation task is known to be predominantly made of music in simple duple meter. Therefore, this assumption will hold in most of the cases on this dataset but does not generalise. Most musical pieces are expected to fall under one of the first two conditions, however.

The weight of the two tempi W_1 and W_2 are then used to compute the relative strength, ST_1, of tempo 1 in comparison to tempo 2:

$$ST_1 = \frac{W_1}{W_1 + W_2} \tag{4.12}$$

4.6.3 Results

The results of the MIREX audio tempo estimation task for the years 2011-2016 are regrouped in Figure 4.12. The algorithm presented here is highlighted in green and labeled QHS1. Its detail score is given in Table 4.5.

The QHS1 algorithm achieves an overall performance that is in the upper half of best performances. This is a good result given the extreme simplicity of this algorithm compared to the other entries in the challenge, which usually involve more complex systems. It produces at least one correct estimate in 92% of the cases, but correctly estimates the

	MIREX audio tempo algorithm
At least one tempo correct	0.92
Both tempi correct	0.56
P-Score	0.80

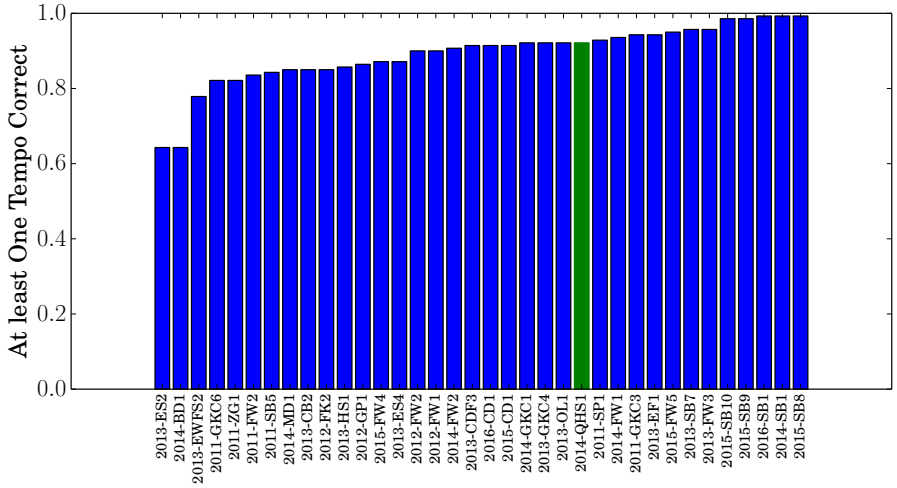
TABLE 4.5: Mirex results

two tempi in only 56%. Unfortunately, because the test set is hidden from participants, it is impossible to investigate in more detail the causes of failure. For instance, the algorithm fails at estimating even one tempo in 8% of the cases and there is no way to investigate the causes of failure further. Similarly it is not clear if the algorithm fails at estimating both tempi because it is picking an incorrect metrical level, or because the metrical level estimation is incorrect all together. However, the results reported in Section 4.5 suggest that the algorithm is robustly picking metrical level pulse rates, and we therefore hypothesise that it is likely that the main cause of failure is picking a metrical level that had not been annotated as the tempo.

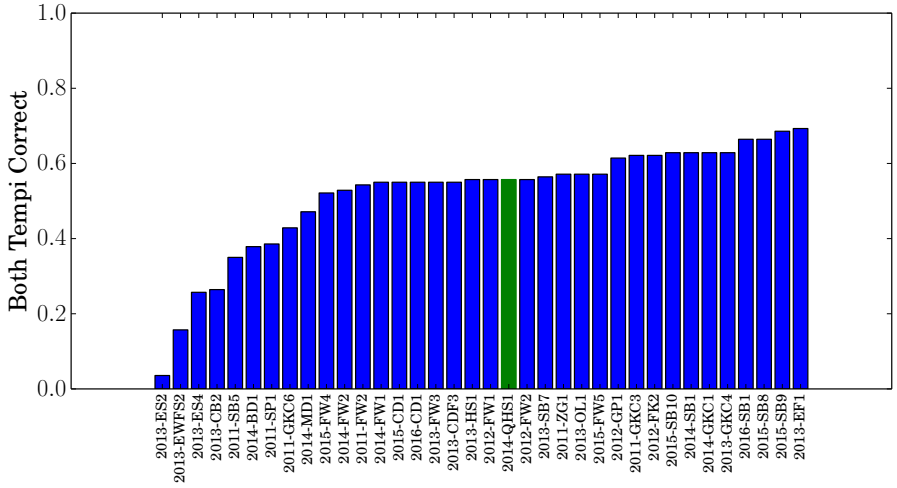
The overall results of the MIREX audio tempo estimation task show that it can be considered that there is little to no room for improvement over the best performing methods with respect to the *at least one tempo correct* metric, as several algorithms scored 0.99 and a large number of entries consistently score over 0.90. In contrast, getting *both tempi correct* proves to be a much more challenging task and therefore still leaves room for improvement. The P-score being a composite derivative of the two other metrics and improving on current state of the art will only be possible by improving on the *both tempi correct* metric.

4.7 Conclusions

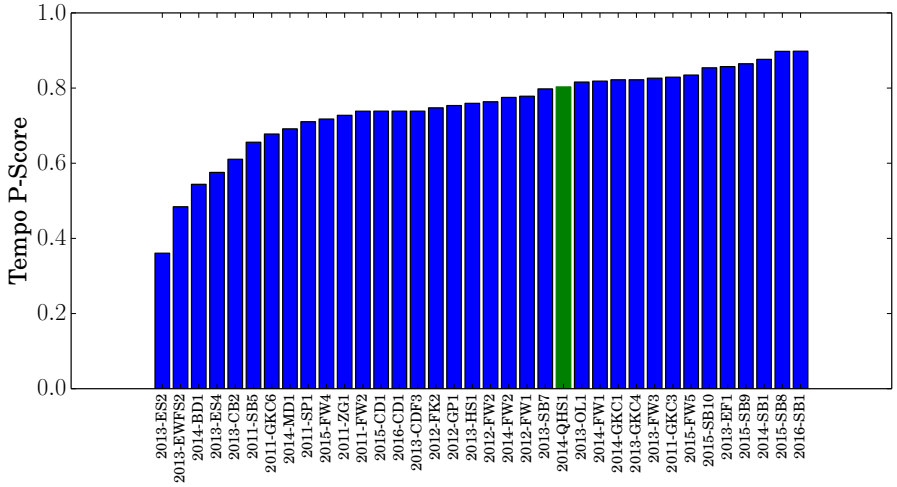
The relationship between peaks in the beat spectrum and metrical level pulse rates has been investigated via the evaluation of algorithms for explicit extraction of all the full metrical structure from the beat spectrum. The conclusions that can be drawn from this evaluation are threefold. Firstly, the results suggest that the hypothesis according to which metrical level pulse rates relate are materialised by peaks in the periodicity spectrum is validated. On the other hand, it clearly appears that not all peaks in



(A) At least one tempo correct



(B) Both tempi correct



(C) Tempo P-score MIREX 2014

FIGURE 4.12: **MIREX 2014 audio tempo results.** The algorithm presented here is labeled QHS1 and highlighted in green in the bar charts

the periodicity spectrum relate to metrical level pulse rates. As result, retrieving the metrical structure from the periodicity spectrum is not straightforward. Secondly, the rhythmogram multiplication strategy proposed by Peeters in order to address this issue appears to be effective to preserve the peaks that relate to metrical level pulse rates while eliminating some of the peaks that do not. However, the elimination is not total, which implies that extra post-processing is required to estimate the metrical structure. Finally, a peak-picking strategy enforcing hierarchical constraints rooted in music theory appeared to significantly improve the accuracy of metrical structure extraction.

Further analysis of the performance is performed by estimating the algorithm performance relative to a measure of the level of agreement between annotators. In this context, we find that the proposed system reaches 93% of maximum achievable accuracy, and largely outperforms the baseline method. Taking advantage of the genre cluster annotations available for the dataset under consideration, the performance was evaluated for the 10 genres in parallel. It appears that for common western popular music genres the metrical structure estimates closely match the expert annotations. However, the analysis of performance on tracks in the Jazz and Classical categories suggest that some musical properties such as syncopation or lack of clear accentuation of strong and weak beats patterns make the estimation of metrical structure more difficult for the machine than it is for human experts. This suggests that there is still room for improving the robustness of algorithms against these properties. On the other hand, by relying on a onset detection function and on the notion of pulse rates, most metrical structure estimation algorithms, including the one proposed in this chapter, implicitly assume the existence of a relatively stable and salient pulse and that musical events result in peaks in the onset detection function. But these fundamental assumptions are contradicted by the musical properties that can be observed in a number of tracks in the Classical class, which explains why neither the algorithm nor human experts are able to provide a satisfactory analysis of these pieces within this framework. As a result it suggests that overcoming such difficulties is not a matter of making algorithms better at detecting localised events and salient pulses but would rather require the use of a different formal model of metrical structure on which to build an automatic estimation system. The definition of such a formal framework is still an open question, however.

Chapter 5

Estimating the Reliability of Rhythmic Features Extraction

5.1 Introduction

The design of systems for automatic feature extraction from audio is a central aspect of the field of Music Information Retrieval. Combining features or using one feature to inform the extraction of another (e.g. beat synchronous chromagram) has appeared to be a fruitful approach [197–199]. In such a setting, it is typically assumed that the base features (e.g. the beat positions) are reliable. However, it is widely known that despite exhibiting good and increasing performance, the automatic extraction of musical features (e.g. tempo or chords) occasionally fails and feature extraction systems seldom provide an indication of reliability. In the absence of a reliability indication, only two postures can be adopted: we either assume the extracted feature is reliable and use it as it is, knowing that it will occasionally fail, or we disregard the feature all together — i.e. it is considered unusable. Blindly relying on extracted features without the provision of additional information implies that the cases of failure cannot be anticipated. Unpredictable failures are a major barrier to building trust in an automated system and

therefore present a significant obstacle to the adoption and/or usefulness of such a system for scientific research as well as industrial applications[1].

The fact that automatic musical feature estimation systems occasionally fail, but, more often than not, do produce a useful estimate leads us to ask: How can we facilitate the development of trust in MIR feature extraction systems? How can we make them usable and useful despite their imperfection? In this chapter we propose a method to estimate the reliability of automatically extracted features, focusing on the case of high-level rhythm features, namely tempo, beat positions and metrical structure. A music recording is given an estimate of the probability that a reliable (or erroneous) estimate would be produced from it. As a result, although the intrinsic performance of the feature extraction scheme is left untouched, the failures are then predictable, so the hypothetical complex system relying on this feature can react accordingly. In this context, the provision of a reliability estimate forms a strategy for handling the imperfection of the MIR feature extraction systems. Our method for predicting reliability shall be seen as a flag providing an extra information about features produced for a given track. In this capacity, it can be used as a way to advice users or complex systems whether or not it is worth or useful to attempt using rhythm related features extracted from a given track. For example, let us considers a hypothetical scenario where a user facing interface which content and operation rely on a rhythm feature estimate. If, for a given track, our method predicts very low reliability, it might be better to skip this track and avoid exposing it to the user. It is important to note, however, that our method only provides an estimate of the reliability and that the decisions to be made from it naturally depend on the application scenario and should be adapted accordingly. The discussion of such strategies, however, is beyond the scope of this work, and we leave it at the discretion of interested parties.

It has been extensively reported in the MIR literature that it is difficult to reliably extract high-level rhythm related features from musical excerpts having properties such as soft onsets, heavy syncopation or use of expressive timing (e.g. rubato playing). Thus far, there has been relatively little effort in quantifying this, however. The extraction of indicators such as 'beat strength' [85] and 'pulse clarity' [200] has been proposed. In these

[1]This problem has been faced on a number of occasions when deploying MIR features at industrial scale and leveraging them to power consumer facing products, via our collaboration with Omnifone Ltd.

studies, the proposed calculation methods were directly evaluated against annotations produced by human ratings. The impact of such an attribute on the extraction of related rhythm features was not investigated, however. Goto [88] proposed a similar approach to estimate the strength of the beat by calculating the difference of the power on the beat, and the power on other positions in order to assess the beat tracking difficulty of a song. The underlying hypothesis is that beat tracking is easiest on a piece with a strong beat. Note that this feature is computed within a beat tracking algorithm in order to adapt its characteristics to the music excerpt under analysis.

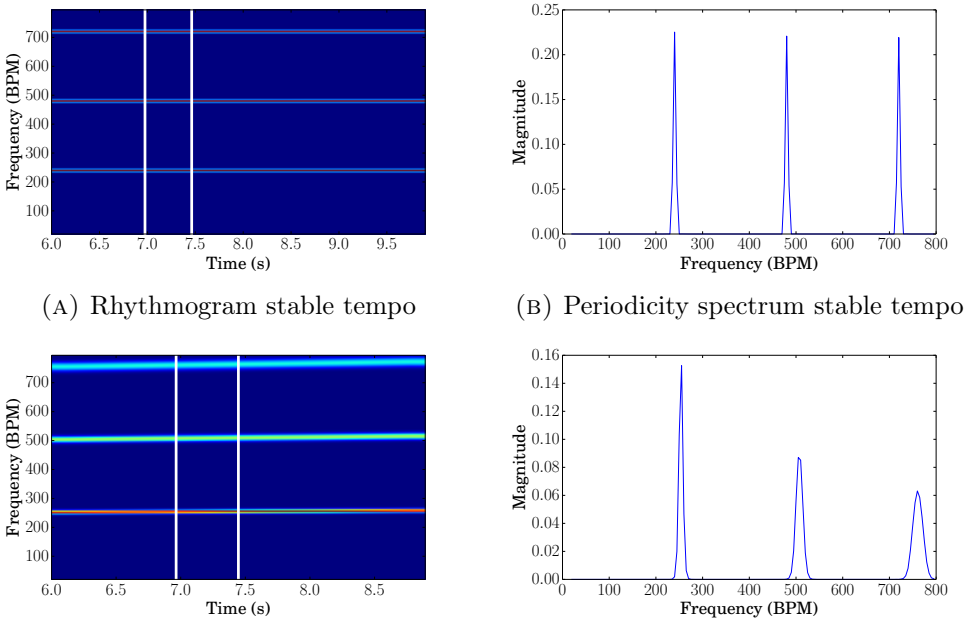
In a case study on Chopin Mazurkas, Grosche tried to identify the musical properties that make a piece difficult for automatic beat tracking [201]. He isolates properties such as trills, arpeggios and grace notes, which manifest as several note onsets very close to each other and therefore make the precise estimation of the beat position harder. Other difficult properties that were reported include soft onsets and tempo changes. Holzapfel proposed a beat tracking difficulty estimation method based on disagreement in a committee of beat trackers, so that a disagreement suggests a difficult case for beat tracking [3]. This method was used to produce the SMC dataset comprising tracks that are difficult for beat tracking which we use in the following sections (cf. Section 3.2). Thanks to tags provided by the annotators, Holzapfel et al. identified recurrent musical properties that make beat trackers fail and summarised them in three categories: i) timing and tempo related (i.e. expressive timing, tempo change, rich ornamentation etc.), ii) lack of clear rhythmic onsets (i.e. lack of transient sounds and quiet accompaniment) and iii) metrical structure related (i.e. compound meter). The existence of category iii) suggests that most beat trackers are geared towards music in simple duple meter. These findings corroborate Grosche's conclusions as the challenging properties he reported fall in categories i) and ii). In Section 5.2 we provide a number of examples showing how the distribution of energy in the rhythmogram is affected by musical properties falling in categories i) and ii), thereby motivating its use as a base feature for the estimation of rhythm feature extraction reliability. It has been shown in Chapter 4 that the rhythmogram captures metrical structure-related information, so this aspect, which corresponds to category iii), will not be addressed again in this chapter.

In recent work, Thoshkahna demonstrated that the entropy of a cyclic rhythmogram (labelled '*tempogram*' in the original publication) [202] can be used as an indicator of the tempo salience of a musical piece [203]. Building on the premise that the distribution of energy in the rhythmogram is affected by properties of the audio signal that make beat tracking and other rhythm features extraction difficult, in Section 5.3 we propose to extend Thoshkahna's work and use the entropy of a rhythmogram to provide an estimate of the reliability of rhythm features extraction. Section 5.4 describes the experiments and the results are presented and discussed in Section 5.5 for the extraction of three rhythm features, namely the tempo, metrical structure and beat positions.

5.2 Rhythmogram and challenging musical properties

In this section, we illustrate how musical attributes known to be challenging for rhythm feature extraction affect the energy distribution in rhythmogram frames. Let us first consider the timing and tempo related properties — i.e. category i) according to [3]. In particular we illustrate how departing from the canonical model of a steady and consistent tempo affects the energy distribution in a rhythmogram. For this matter, two synthetic audio excerpts were produced: the first one consists of a percussive sound repeated at a frequency of 240BPM; realised by creating a MIDI score and triggering the corresponding sound from it. The second example was produced using the same MIDI score and sound but this time linearly increasing the tempo between 240 and 270 BPM over the duration of the excerpt (15s). The corresponding rhythmograms are given in Figure 5.1 (A) and (C). In addition, periodicity spectra, effectively corresponding to a frame of each rhythmogram (delimited by the vertical lines), are given in Figure 5.1 (B) and (D).

For the fixed tempo example, the audio signal consists of sharp percussive events, which translate into sharp and equally spaced peaks in the onset detection function. In other words, the onset detection function is periodic, so that $\Phi(t) = \Phi(t + \tau)$ where Φ is the onset detection function (ODF) and τ is the period. Then, the ODF is closely approximated by a Dirac comb convolved with the envelope shape specific to the percussive



(A) Rhythmogram stable tempo

(B) Periodicity spectrum stable tempo

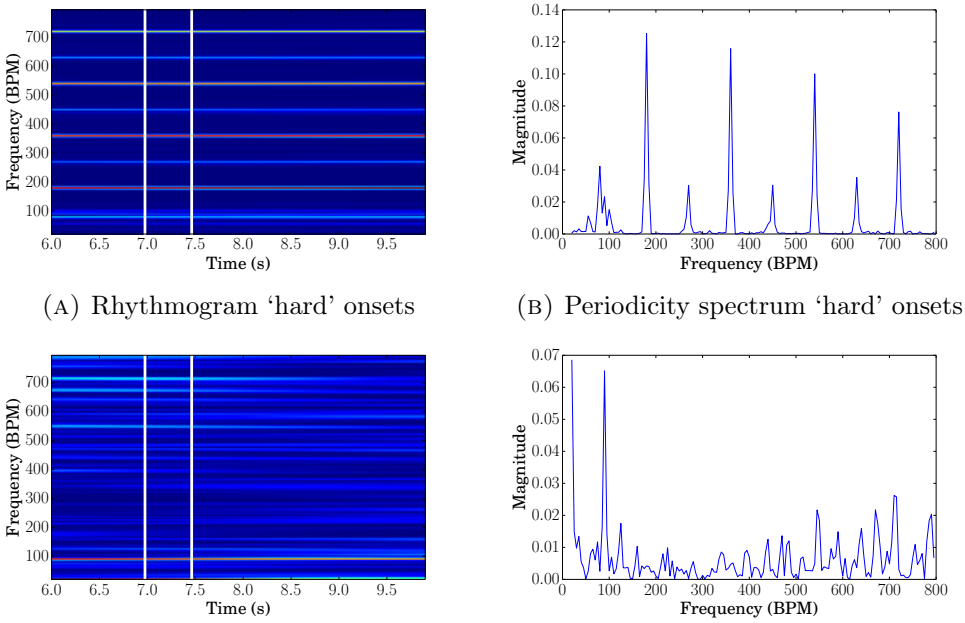
(C) Rhythmogram unstable tempo

(D) Periodicity spectrum unstable tempo

FIGURE 5.1: **Rhythmograms and periodicity spectra for stable and unstable tempo.** The rhythmogram and corresponding periodicity spectra presented here were computed from two audio click tracks. The two excerpts were synthesised using the same instrument (monophonic percussive sound). Each excerpt was synthesised under a specific tempo condition: stable tempo in cases (A) and (B) (240BPM) and unstable tempo, linearly increasing the tempo between 240 and 270 BPM over the duration of the excerpt (15s), in cases (C) and (D). The periodicity spectrum corresponds to the frame delimited by the white vertical lines on the rhythmogram and is normalised to sum to one.

sound. As a result the periodicity spectrum also exhibits sharp peaks arranged in a harmonic structure (cf. Figure 5.1 B).

An unstable tempo implies that the ODF is no longer strictly periodic. This may be modelled by a time-varying quasi-period $\tau(t)$ so that $\Phi(t) = \Phi\left(t + \tau(t)\right)$. In the context of the frame-wise analysis that is considered here, the periodicity estimation is effectively integrated over the window length. Given that the windows used for the periodicity analysis need to be long enough to capture the rhythmic and metrical periodicities of interest (typically several seconds), a change of tempo occurring within a window would result in the integration of its corresponding (time-varying) periodicity. As a result the peaks in the beat spectrum are expected to be widened in presence of tempo variations. This phenomenon can be observed in Figure 5.1. The peaks of the periodicity spectrum depicted in Figure 5.1 (D) are not as sharp as in Figure 5.1 (D), which also translates in thicker horizontal lines in the rhythmogram in Figure 5.1 (C).



(A) Rhythmogram 'hard' onsets

(B) Periodicity spectrum 'hard' onsets

(C) Rhythmogram 'soft' onsets

(D) Periodicity spectrum 'soft' onsets

FIGURE 5.2: **Rhythmograms and periodicity spectra for hard and soft onsets.** The rhythmogram and corresponding periodicity spectra presented here were computed from audio excerpts synthesised from the same MIDI score at constant tempo, using two different instruments: a piano in cases (A) and (B) and a synthesiser producing soft onsets in cases (C) and (D). The periodicity spectrum corresponds to the frame delimited by the white vertical lines on the rhythmogram and is normalised to sum to one.
The entropy of the periodicity spectra is (B) $S = 0.68$ and (D) $S = 0.89$

Let us now consider the category of musical attributes that regroups all forms of lack of clear onsets, i.e. category ii) according to [3]. This could correspond to 'soft' onsets as may be produced by bowed string instruments for instance. The term 'soft' onset is used in contrast with 'hard' onsets, which are typically produced by percussive instruments. Many authors also report 'quiet accompaniment' to describe a similar property [3], i.e. it is not clear where the onset of a musical event (e.g. a note) is located in time. Although we refer the reader to Section 2.2 for a discussion of soft onsets and onset detection functions, we recall that ODF are canonically designed to reveal onsets as peaks and that most rhythm analysis systems rely on an ODF. Since soft onsets typically do not produce clear peaks in the ODF, subsequent rhythmic analyses may be undermined.

In order to illustrate the effect of 'soft onsets' on the rhythmogram and beat spectrum, two musical excerpts were synthesised. The excerpts are both based on a MIDI score of the first two bars of John Lennon's *Imagine*; chosen because it features a simple and

steady rhythm pattern. The first excerpt was produced by synthesising an audio track using a grand piano sound, featuring 'hard' onsets by virtue of the percussive nature of the piano sound production mechanism. The second audio track standing as example of 'soft onsets' was produced using a synthesised sound with very smooth attacks. In both cases the tempo is constant and the note onsets locations are quantised to strict metrical positions on a grid. Figure 5.2 shows the rhythmogram and beat spectra obtained under these two conditions. In the case of 'hard' onset, the canonical behaviour is observed; the rhythmogram features clear horizontal lines and the periodicity spectrum features clear peaks, organised in harmonic series, which account for clear periodicities of the onset detection function. In contrast, in the 'soft' onsets example some peaks are discernable, but they are not as salient as in the hard onset case, and no clear harmonic arrangement is apparent. Moreover, the periodicity spectra are normalised to sum to unity, and thus reveal that most of the energy is located in the peaks in the case of the hard onsets while it is more homogeneously distributed in the case of soft onsets.

5.3 Rhythm salience feature

We hypothesise that musical attributes known to make high level rhythm feature extraction fail, such as expressive timing or soft onsets result in rhythmogram structures in which the energy is more uniformly distributed along the frequency axis. The examples given in Section 5.2 were provided to illustrate this phenomenon and thereby motivate this hypothesis. As a consequence, we hypothesise that the evenness of the distribution of energy in the rhythmogram frames (in other words measuring its peakiness) should be related to the reliability of rhythm feature extraction.

The entropy, commonly known as a measure of certainty when applied to probability distributions, is in fact a peakiness measure. On this premise, Thoshkahna proposed to use the entropy of the frames of a cyclic rhythmogram[2] as a measure of tempo salience [203], but did not investigate the relation between tempo salience and feature extraction reliability. The cyclic variation of the rhythmogram is computed by wrapping it over one octave so that pulses related by a power of two are identified, by analogy with the

[2]labelled as 'tempogram' in the original publication

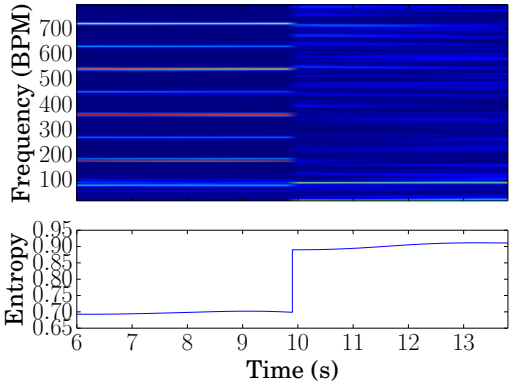


FIGURE 5.3: **Rhythmograms and entropy for examples of hard and soft onsets**. The rhythmogram is the concatenation of rhythmograms obtained for hard and soft onsets examples and shown in Figure 5.2 (A) and (C). The curve in the lower panel show the entropy of the corresponding rhythmogram frames.

chromagram processing for pitch [202]. While in the case of the chromagram the octave wrapping is justified by the octave equivalence in human pitch perception, Grosche motivates the construction of the cyclic rhythmogram solely on the basis of the mathematical analogy. Moreover, the wrapping proposed by Grosche relies on octave ratios (i.e. frequency ratios equal to powers of two), which introduces a strong bias towards simple duple meters. For these reasons, the cyclic rhythmogram is not used here. However, we propose to use the entropy of the rhythmogram frames.

The computation is carried out as follows: An onset detection function is computed from the audio signal using the *Superflux* method [62]. A rhythmogram $\mathbf{R}_F$ is then generated as the Fourier transform based magnitude spectrogram of the onset detection function, using 12s long Hanning windows and 0.2s step. The columns of the rhythmogram are normalised with respect to the L_1 norm, and for a vector $\mathbf{r}_n = (r_{1,n}, \cdots, r_{M,n})$ in $\mathbb{R}^M$ representing the n^{th} rhythmogram frame, the entropy S_n is defined as:

$$S_n = \frac{\sum_{m=1}^{M} -r_{m,n} \log_2 (r_{m,n})}{\log_2(M)} \tag{5.1}$$

with M the number of frequency bins in the rhythmogram. This way, a vector $\mathbf{S} = (S_1, \cdots, S_N)$, where N is the total number of rhythmogram frames, captures the evolution of the entropy over time.

Entropy, commonly used as a measure of disorder, or uncertainty in a probabilistic framework, takes high values for uniform distributions of energy in vector $\mathbf{r}_n$, and small values for highly organised, and therefore uneven, distributions. In other words, the entropy of vector $\mathbf{r}_n$ is small when the energy is mostly concentrated in peaks and conversely. In this context, musical recordings featuring challenging properties for high level rhythm feature extraction are expected to result in high rhythmogram entropy whereas tracks with a clear, steady pulse and hard onsets are expected to result in good feature extraction performance and to yield lower entropy values. Figure 5.3 shows an example of rhythmogram and corresponding entropy for hard and soft onsets.

5.4 Experiments

In this section we lay out experiments to test the hypothesis according to which the entropy of rhythmogram frames is related to the reliability of rhythm feature extraction. First, several features are extracted from audio and the performance of the extraction is evaluated using standard metrics. Secondly, the rhythmogram entropy is computed for every track according to the method specified in Section 5.3. We then investigate how it relates to the evaluated feature extraction performance. It is important to note that the aim of this study is not to investigate nor to improve feature extraction or evaluation methods per se (we refer to relevant literature for this purpose) but to focus on the analysis of the relationship between rhythmogram entropy and feature extraction performance.

Three high level rhythm feature extraction procedures were considered: tempo estimation, metrical structure estimation and beat tracking. Tempo estimation is performed using the Vamp plugin implementation[3] of the algorithm introduced by Davies et al. in [121] (cf. Section 2.5). The metrical structure is extracted using the algorithm we presented in Chapter 4. For beat tracking, the evaluation results are drawn from a prior study on beat tracking evaluation [3]. Two publicly available datasets are used to carry out the estimation of feature extraction reliability. The GTZAN dataset [183] is used in the case of tempo and metrical structure, along with the corresponding annotations for

[3] http://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html#qm-tempotracker

tempo[4] and metrical structure (cf. Section 3.5). The average tempo and the metrical structure estimate for the average periodicity spectrum (identically to the processing of Section 4.3.1) are used for each track since the GTZAN dataset comprises 30 seconds long excerpts of overall reasonable consistency (i.e. they do not contain a lot of musical changes).

For each track, the estimated tempo is compared to the annotated tempo and considered correct if they are equal within a tolerance window of 8% of the annotated value. This tolerance window is chosen consistently with the standard adopted in the MIREX audio tempo evaluation task[5], which is a commonly used standard in MIR literature, so that the results presented here are comparable with existing work. We refer the reader to Chapter 4 for a detailed description of the metrical hierarchy feature extraction evaluation metrics. However, we recall that, the extracted feature consists of an estimate of the pulse rate of all the metrical levels present in the music. These are compared with the corresponding annotations and the result is summarised by an F-measure. In both the tempo and metrical structure cases, the feature extraction procedure is considered reliable if it consistently matches the human annotations. In the case of beat tracking, we rely on the difficulty assessment by disagreement in a committee of beat trackers [3]. Holzapfel et al. then used this method to compose the SMC dataset, made up 'hard' and 'easy' musical excerpts (cf. Section 3.2). The hard tracks were chosen for their propensity to generate disagreement in the committee, that is to say unreliable beat estimates. Conversely, 'easy' tracks were chosen for their propensity to generate consistent agreement in the committee. It is therefore considered that the estimates produced for 'easy' tracks are reliable. For each musical excerpt considered in this chapter, we computed the entropy of all rhythmogram frames according to equation 5.1, and an average entropy value $\hat{S}$ obtained as:

$$\hat{S} = \frac{\sum_{n=1}^{N} S_n}{N} \tag{5.2}$$

where N denotes the total number of rhythmogram frames.

[4]http://www.marsyas.info/tempo/genres_tempos.mf
[5]http://www.music-ir.org/mirex/wiki/2015:Audio_Tempo_Estimation

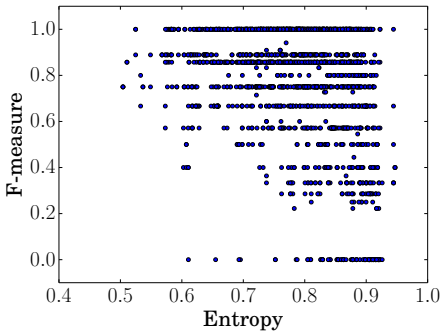


FIGURE 5.4: Metrical structure feature extraction performance, given by the F-measure, against track mean entropy $\hat{S}$. Each dot on the graph represents the results of the evaluation for a track of the GTZAN dataset.

5.5 Results

In this section we analyse the relationship between the rhythmogram entropy and the performance of rhythm feature extraction algorithms, evaluated according to the methods described in Section 5.4. Since the evaluation procedures are feature specific, the results are presented on a per-feature basis.

5.5.1 Metrical structure

We first investigate the existence of a linear correlation between the entropy and the performance F-measure for all the songs. The Pearson coefficient was computed and is presented in Table 5.1. Pearson's coefficient is a measure of linear correlation and results in values which magnitude ranges from 0 (no correlation) to 1 (maximum correlation), with the sign indicating the direction of the correlation. For the metrical structure, the Pearson coefficient reveals a significant but weak correlation between entropy and the algorithm performance. This evidence is graphically corroborated by the scatter plot of Figure 5.4 in which no clear linear trend is apparent. However, it is apparent on this plot that the bottom left area contains virtually no data points, which seem to indicate a tendency in the distribution: tracks with low entropy tend to consistently lead to good performance while tracks with high entropy result in inconsistent performance.

In order to gain more statistical insight, we now group data points by entropy classes. Each class regroups tracks for which the average entropy $\hat{S}$ is within a given interval.

TABLE 5.1: Correlation coefficients between entropy and both the mean tempo accuracy and the metrical structure F-measure, alongside with the corresponding p-value.

	Tempo		Metrical Structure	
	Coefficient	p-value	Coefficient	p-value
Pearson	-0.947	0.0001	-0.282	<0.0001

Figure 5.5 shows a boxplot of the distribution of the feature extraction performance F-measure for each entropy class. Although the [0.6,0.65] class appears as a relative outlier, it suggests a tendency for the performance characterised by the F-measure to be relatively consistent up until the entropy reaches values around 0.8, and a clear decrease of both mean performance and performance consistency (characterised by the spread of the distribution) is observed. In order to assess the statistical significance of the drop in mean performance, a Mann-Whitney U-test was run on F-measures distributions belonging to adjacent entropy classes. The results are shown in Table 5.2 and confirm that the decrease of mean performance observed for entropy values higher than 0.8 is statistically significant at the 0.001 level. The distributions for the two smaller entropy classes also exhibit apparently significant differences in their means ($p < 0.01$). The number of observations in theses classes is small (<10 in the lowest entropy class) and the overall mean F-measure remains very high as well as the spread of the distribution remains small. As a consequence, although the means of these two classes are different, the data still suggests both high performance and high performance consistency, with high mean and narrow distribution. The width of the distribution in the [0.6,0.65] entropy class is probably affected by a number of relatively mediocre performance outliers, as suggested by the scatter plot of Figure 5.4. Nevertheless, its mean appears not to be significantly different from the mean of the [0.65,0.7] class.

In conclusion, it appears that for entropy values higher than 0.8 (approximately), the mean performance significantly decreases and the consistency of performance also decreases, as suggested by the widening of the performance scores distribution. In other words, the reliability of the feature extraction drops significantly for high entropy values, while it remains relatively stable on the lower range.

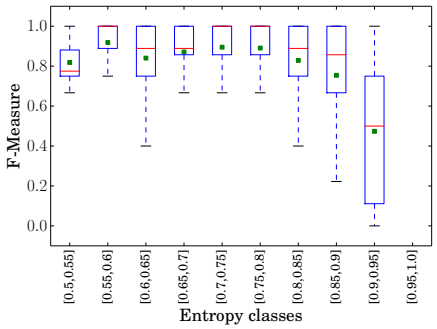


FIGURE 5.5: F-measure distribution for each entropy class. The box extends from the lower to upper quartile values of the data, with a red line at the median while the green dot is the mean. The whiskers extend from the box to show the range of the data, excluding the outliers.

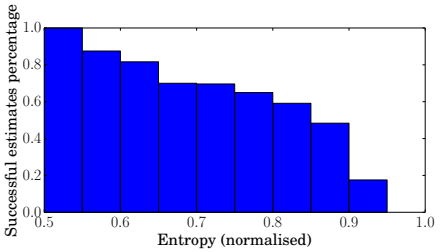
TABLE 5.2: Mann-Whitney u-test p-values for the mean of F-measure of metrical hierarchy evaluation. Values rejecting the null hypothesis of equal means at the 0.01 level are in bold font.

Entropy classes	Means	Sample Sizes	p-value
$[0.50, 0.55]$ - $[0.55, 0.60]$	0.82 - 0.92	14 - 45	$<$ **0.001**
$[0.55, 0.60]$ - $[0.60, 0.65]$	0.92 - 0.84	45 - 125	**0.002**
$[0.60, 0.65]$ - $[0.65, 0.70]$	0.84 - 0.87	125 - 207	0.062
$[0.65, 0.70]$ - $[0.70, 0.75]$	0.87 - 0.90	207 - 430	0.094
$[0.70, 0.75]$ - $[0.75, 0.80]$	0.90 - 0.89	430 - 562	0.088
$[0.75, 0.80]$ - $[0.80, 0.85]$	0.89 - 0.83	562 - 555	$<$ **0.001**
$[0.80, 0.85]$ - $[0.85, 0.90]$	0.83 - 0.75	555 - 538	$<$ **0.001**
$[0.85, 0.90]$ - $[0.90, 0.95]$	0.75 - 0.47	538 - 147	$<$ **0.001**

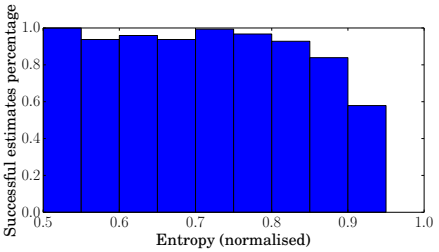
5.5.2 Tempo

The evaluation of tempo extraction provides a dichotomy between correct and incorrect estimations. The resulting data is grouped in entropy classes so that some statistical information can be derived. The percentage of successful tempo estimation for each entropy class is given in Figure 5.6 (A). The apparent trend in this data suggests that the tempo extraction accuracy decreases as the rhythmic entropy increases. The Pearson coefficient computed for the middle of the entropy class and the mean tempo accuracy for each class is given in Table 5.1. It reveals a strong and significant negative linear relationship between entropy and tempo estimation accuracy.

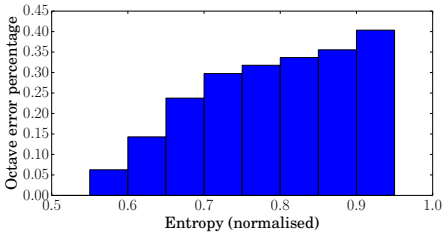
Tempo usually represents the rate of a metrical level, and an 'octave error' occurs when the algorithm produces a tempo estimate that is typically half or twice of the annotated



(A) Tempo estimation accuracy



(B) Tempo estimation accuracy, octave error tolerated



(C) Octave error rate

FIGURE 5.6: **Tempo estimation performance against rhythmogram entropy.** (A) Mean tempo extraction accuracy (proportion of correct estimations) per entropy class. (B) Mean tempo extraction accuracy (proportion of correct estimations) per entropy class, also considering an octave error by a factor 1/3, 1/2, 2 or 3 as correct tempo estimate. (C) Mean octave error rate per entropy class.

tempo in the case of duple meter (a third or three times in the case of triple and compound meter). As such, the 'octave error' estimate effectively corresponds to a different metrical level than the one which the annotated tempo is associated with. Therefore, incorporating tolerance to octave error in the evaluation procedure implicitly relates to the estimation of a part of the metrical structure. Interestingly, if the evaluation metric that we use so that it counts an 'octave error' (by ratios of either 1/3, 1/2, 2 or 3) as a correct tempo estimate, the distribution of percentage of 'correct' estimates exhibits a shape very similar to the distribution of mean F-measure in the case of metrical structure extraction, as shown by the comparison of Figure 5.6 (B) and Figure 5.5. Here again, the performance appears to be relatively stable from lowest entropy class up to a critical value (around 0.8) after which the performance drops. Note that the difference between Figure 5.6 (A) and (B) is the octave error rate, which increases as the entropy increases as shown in Figure 5.6 (C)

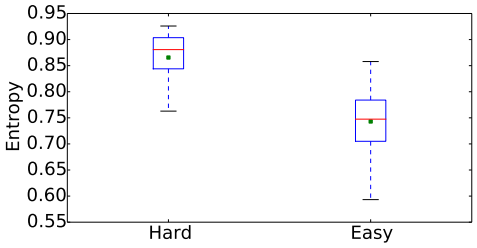


FIGURE 5.7: Entropy distribution for the dataset published by Holzapfel *et al.* [3]. The box extends from the lower to upper quartile values of the data, with a red line at the median while the green dot is the mean. The whiskers extend from the box to show the range of the data, excluding the outliers.

5.5.3 Beat tracking

In the case of beat tracking, we use the SMC dataset for evaluation purposes. In particular, we investigate if a significant difference between the distributions of entropy of 'hard' and 'easy' tracks can be observed. The entropy distribution for 'hard' and 'easy' categories are graphically set apart in Figure 5.7. In addition, a Mann-Whitney U-test strongly rejected the null hypothesis of equal means of the two distributions at the 0.001 level, which means that 'easy' tracks tend to have a significantly smaller entropy than 'hard' tracks. This suggests that the entropy measurement is correlated with the results obtained by Holzapfel *et al.* [3]. In other words, the beat tracking difficulty (and thus the reliability of the beat estimates) that had been estimated using beat tracker disagreement, is also related on average to measurement of the rhythmogram entropy.

5.6 Conclusions

Despite good and increasing performance of contemporary algorithms, the automatic extraction of musical features occasionally fails. In addition, feature extraction systems often do not provide an indication of the reliability of the corresponding feature, which makes the inevitable failures unpredictable. Given this unpredictability, such systems are doomed to be (at least to some extent) unreliable, and therefore unusable in a variety of research and application scenarios. As a consequence, providing a reliability or confidence value alongside an extracted feature significantly increases its usability in complex and/or composite systems and studies.

In response to this observation, the relationship between the entropy of a rhythmogram derived from the audio, and the reliability of the extraction of several high level rhythm features was investigated in this chapter. In particular, tempo and metrical structure estimation algorithm were considered along with beat tracking. A feature extraction system is considered reliable if it performs consistently well. The results show that the entropy of rhythmogram frames is statistically related to the reliability of the extraction of multiple high-level rhythm features, a higher entropy being related to lower feature extraction reliability and vice versa. Given that the rhythmogram entropy is computed directly from the audio, it is a valuable asset for the production of an estimate of the reliability of high level rhythm feature extraction, independent of the feature extraction itself. As a result, this may enhance the usability of the corresponding features for applications such as exploring large music collections or powering user-facing systems to name but a few [204]. It may therefore prove useful for a number of cases including assessing if it is worth attempting the extraction of rhythm features and attaching a confidence value to a feature for which it was not originally provided. Furthermore, in a machine learning setting, the entropy could be used as a weight for feature or model selection. Naturally, many more applications can be considered, beyond the handful of use case examples that were cited here.

Chapter 6

On Metric Modulations Taxonomy

6.1 Introduction

A metric modulation in a piece of music is understood in this thesis as a change in the metrical structure over time. The nature of the modulation, as well as the effect it produces on the listener, naturally depends on what attributes of the metrical structure are modified and how. Modulations could for instance be realised by the alteration of the number of beats in a bar, the beat subdivision, the beat rate, or possibly several of these occurring simultaneously. It follows that metric modulations come in a variety of types, and we will revisit this point later. We recall that our usage of the term *metric modulation* in this thesis includes any type of changes of meter. As a consequence, the terminology used by other authors may or may not be in line with ours. In most instances, authors have focused on a subset of all possible metric modulations and have named them accordingly. However, unless stated otherwise, we will not adopt the terminology proposed by other authors in the remainder of this thesis.

Fétis, who wrote prolifically on the topic in the 19^{th} century (see for instance [205, 206]), stated that rhythm had been left in a secondary position [throughout history] and advocated for a more prominent role to be given to rhythm as a compositional device [207]. His stance has later found an echo in a movement of 20^{th} century avant-garde music to which complex manipulations of rhythm and meter were central pillars of

compositions. Elliott Carter, who gives primary importance to rhythm in his music [208] is seen as a pioneer in the art of using tempo and metric modulations as a fundamental compositional technique, and is widely considered as a figurehead of this movement. As a consequence, his work is of particular interest for musicologists interested in rhythm studies [209–212]. Although not used so extensively or intensely outside of 20^{th} century avant-garde music, metric modulations are not a rarity in the more common musical repertoire. For instance, they are frequently encountered in Stravinsky's body of work and in the progressive rock genre as well as occasionally in popular music. Madonna's 'Dear Jessie' contains a metric modulation, as does the music of the Beatles, to only name a few.

Study, analysis, characterisation and classification of harmonic modulations are extensively covered in the musicology literature. In contrast, there are only very few studies focusing on metric modulations. Nevertheless, as Fétis mentioned in the 19^{th} century, there exist numerous parallels between metric and harmonic modulations [207] and this analogy offers an opportunity to transfer many of the benefits of work carried out on harmonic modulations to its metric counterpart. Bouchard [213], inspired by the concept of this analogy, proposed a transposition of the theoretical framework for tonal modulation analysis into the rhythm domain. From this theoretical framework, he derived a relatively exhaustive taxonomy of metric modulations.

With the aim of enabling musically meaningful analyses, this chapter is concerned with the question of outlining a musicologically-grounded metric modulations taxonomy suitable for automatic classification of metric modulations from audio recordings. We first provide in Section 6.2 a description, in English, of the metric modulations taxonomy proposed (in French) by Bouchard in [213]. While we intend here to perform audio-based analysis, Bouchard proposed a score-based taxonomy of metric modulations. Musical scores are not always available and recreating a score from an audio recording, which is known as automatic transcription, is by no means a solved problem in the current state of the art [101, 133]. Consequently, it is not possible to directly deploy Bouchard's analytical framework to audio recordings. Benadon suggested the use of another musical representation, namely the metrical level pulse rates, as a feature to characterise what

he calls "tempo modulations"; which in fact correspond to a category of metric modulations in Bouchard's taxonomy [214]. We propose a generalisation of Benadon's concept and tie it with the framework introduced by Bouchard in order to produce a taxonomy applicable in an automatic analysis framework. In particular, in Section 6.3 we present a reformulation of Bouchard's taxonomy in terms of metrical level pulse rates, which can be extracted automatically from audio recordings (cf. Chapter 4) rather than score notation.

In Section 6.4 we employ the newly introduced taxonomy for automatic metric modulations classification. In particular, we introduce an automatic classifier based on the taxonomy and then carry out two classification experiments on the dataset introduced in Section 3.6. First we employ the metric modulations classifier to automatically label the modulations from the reference annotations. In a second experiment, we automatically label modulations from automatically extracted metrical structure. We finally draw conclusions regarding the use of such a taxonomy in an automatic classification scenario and propose directions for future work in Section 6.5.

6.2 Bouchard's Taxonomy

In this section we summarise the taxonomy of metric modulations derived by Bouchard in [213]. His theoretical framework is built on the premise of an analogy between tonal and metric modulations. This allowed him to transpose a series of theoretical concepts developed for the analysis of tonal modulations to the case of metric modulations. The notion of *pivot*, which is the unit common to the two structures around which the modulation is articulated, is an example of such a concept. In the case of tonal modulation, the pivot may be a chord or a note, whereas in the case of metric modulation it would typically be a metrical level (e.g. the eighth note). As such, the pivot articulates the modulation while maintaining a link that unites the parts together. Using a pivot is a way for composers to produce modulations that bring a metrical change between parts without sounding disjoint, because the pivotal unit maintains a certain continuity. Choosing the pivot and the mechanism by which the modulation articulates around it allows the

composer to control the musical effect produced. In the following, we briefly describe the varieties of modulation types defined by Bouchard.

Bouchard sorts the metric modulation types he defined into two groups: *Metric Modulations*, which imply a change of metrical structure that leaves the beat rate untouched and *Combined Modulations* which are defined as joint tempo and metric modulations. The following definitions summarise those provided by Bouchard in [213]. They are formulated using his choices of terms in order to maximise the fidelity of the translation.

Tempo Modulation

Tempo modulations regroup the variations of speed of execution in a music piece. Doubling or halving tempo are examples of tempo modulation. Naturally, other speed ratios are possible. This category does not include subjective or expressive speed variations such as *rubato*, *ritardando*, *accelerando* etc. In its simplest form, a tempo modulation does not imply a metric modulation.

Metric Modulation Type I

Bouchard's Type I modulation is defined as a combined alteration of the numerator and the denominator of the time signature, such that the number of beats per bar remains unchanged. The beat rate also remains unchanged (and can therefore be regarded as the pivot). In other words, this modulation is characterised by a change of beat subdivisions. Figure 6.1 shows an example of Metric Modulation Type I.

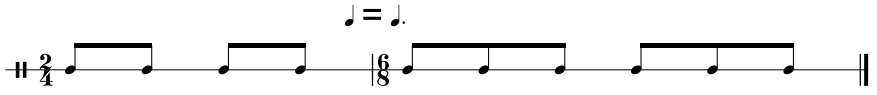


FIGURE 6.1: Example of Metric modulation Type I

Metric Modulation Type II

This type of modulation is characterised by Bouchard as a change of numerator in the time signature i.e. the number of beats in the bar, while the tempo remains unchanged. The beat rate is used as the pivot. Figure 6.2 shows an example of Metric Modulation Type II.

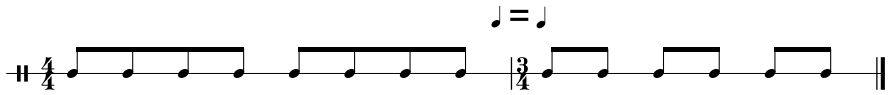


FIGURE 6.2: Example of Metric modulation Type II

Metric Modulation Type I Hybrid

This modulation is characterised by a change of numerator and denominator of the time signature, as well as a change of number of beats per bar, while the beat rate remains unchanged. As such it has got the characteristics of both Type I and Type II modulations, hence the "hybrid" label. Figure 6.3 shows an example of Metric Modulation Type I Hybrid.

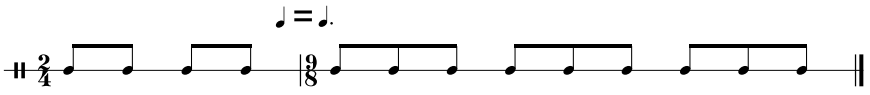


FIGURE 6.3: Example of Metric modulation Type I Hybrid

Metric Modulation Type III

It is characterised as a change of denominator in the time signature while the pivot keeps the same pulse value before and after the modulation. All other metrical aspects are kept identical, including number of beats per bar, strong and weak beats patterns, subdivisions etc. As a result, this type of modulation only exists as a written (i.e. score) artefact, and does not imply any audible change. Bouchard states he added this modulation type in order to provide symmetry of the modulation parameters. Although potentially relevant for score-based musicology and certainly impactful on musicians performance, this type of modulation is not practically useful for an audio-based analysis of music recordings as it is not related to any sonic change. Figure 6.4 shows an example of Metric Modulation Type III.

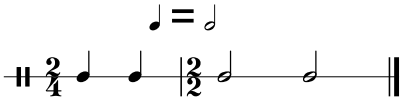


FIGURE 6.4: Example of Metric modulation Type III

Combined Modulation Type I

A Type I combined modulation is a Type I metric modulation — that is to say the numerator and denominator of the time signature are altered — paired with a change of tempo. Two sub-categories of Type I combined modulation can be specified based on the role played by the pivot before and after the modulation:

a. Identical pivot

The numerator and denominator are both modified, and the pivot chosen corresponds to the same note value for each metrical structure. In the example of Figure 6.5, the eighth note is the pivot, which results in a modified beat rate. Figure 6.5 shows an example of Combined Modulation Type I a.

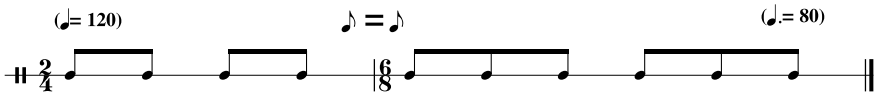


FIGURE 6.5: Example of Combined modulation Type I a

b. Different pivot

The numerator, denominator and beat rate are modified, but the pivot corresponds to different note values for each metrical structure. The comparison of cases a. and b., as well as the corresponding examples of Figure 6.5 and Figure 6.6 illustrates the fact that the choice of the pivot has a direct influence on the resulting tempo modulation, all other elements left untouched.

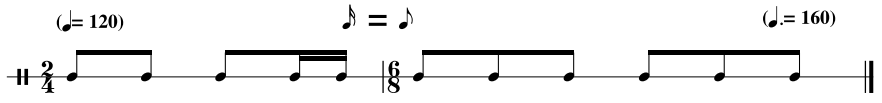


FIGURE 6.6: Example of Combined modulation Type I b

Combined Modulation Type II

A Type II combined modulation is a Type II metric modulation paired with a change of tempo. In this case only the numerator of the time signature changes (i.e. the number of beats per bar). As a consequence, in order to guarantee the change of tempo, the pivot must correspond to a different note value on each side

of the modulation. Figure 6.7 shows an example of Combined Modulation Type II.

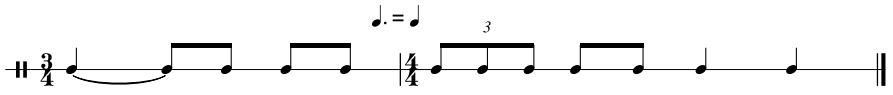


FIGURE 6.7: Example of Combined modulation Type II

Combined Modulation Type I Hybrid

As with Type I and Type II metric modulation, a Combined Type I hybrid modulation is introduced to describe a modulation that exhibits attributes of both a Combined Type I and a Combined Type II modulation. In this case, the tempo, the numerator and denominator of the time signature, as well as the number of beats in a bar are altered. Inheriting from Combined Type I modulations, the pivot may correspond to identical or different note values on each side of the modulation. Figure 6.8 provides an example for a pivot corresponding to identical note values. Note that pivots corresponding to different note values may be used.

FIGURE 6.8: Example of Combined modulation Type I Hybrid

Combined Modulation Type III

A Type III combined modulation is a Type III metric modulation paired with a change of tempo. Here again, two sub-categories can be specified based on the value of the pivot before and after the modulation:

a. Identical pivot

Changing the denominator of the time signature with a pivot having identical value on both side of the modulation necessarily implies a change of beat rate. In the example of Figure 6.9 the beat rate is doubled.

b. Different pivot

In this case, values used as the pivot must not correspond to the denominator

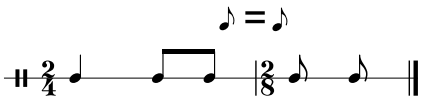


FIGURE 6.9: Example of Combined modulation Type III a

of each time signature in order to guarantee the change of beat rate. They can be chosen freely outside of this limitation. Figure 6.10 shows an example of Combined Modulation Type III b.

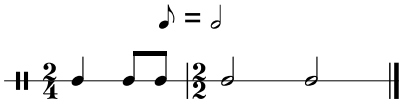


FIGURE 6.10: Example of Combined modulation Type III b

This type of modulation entails a change of beat rate, but does not explicitly relate to a change of hierarchical organisation of the metrical structure. The difference between a Combined Type III modulation and a Tempo Modulation resides in the score notation. As a consequence, these two modulations may be indistinguishable without the score. Although these characterisations are redundant from a sonic perspective, they result in a significant difference for the performing musician reading the score. Figure 6.9 and Figure 6.11 represent the same musical artifact with two different notations implying a Combined Type III modulation and a Tempo Modulation respectively. The main difference between these two notations is that the

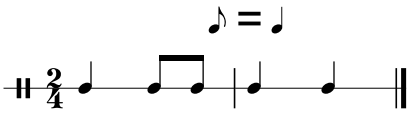


FIGURE 6.11: Example of Tempo Modulation equivalent to a Combined Type III modulation

note values used as the pivot in the case of the tempo modulation are different on each side of the modulation, whereas they are identical in the case of the Combined Type III Modulation. Using a pivot with identical values may prove easier to read for the performer because having different values would require one extra operation

in order to adapt to the tempo change. Scoring a change of tempo as a Tempo Modulation or as a Combined Modulation Type III is then a composer's decision.

6.3 Adapted Taxonomy

In this section we propose a transposition of Bouchard's taxonomy in terms that make it compatible with features that may be extracted automatically from audio recordings. Benadon reported a preliminary study on characterising what he called "tempo modulations" in [214]. His contribution consisted in proposing the use of metrical level pulse rates to characterise "tempo modulations". He did not explicitly tie this concept with any theory or model of metric modulations even though some elements were already present in his work. Notably, he mentioned one or more pulse rates common to the two metrical structures, which effectively correspond to the notion of a *pivot* presented above. He did not incorporate this idea in any automated system either. Our contribution in this section consists is the reformulation of Bouchard's taxonomy using metrical level pulse rates as a feature, in a similar fashion as Benadon. We choose a nomenclature that remains close to Bouchard's taxonomy, so that the proximity remains explicit when it is relevant.

In some cases, the transposition of the definition of metric modulations provided by Bouchard to metrical level rates based framework leaves some ambiguities. For instance, he defines a Type I Combined Modulation as the alteration of the numerator and denominator of the time signature along with a beat rate change. It specifies explicitly the existence of a pivot that is not the beat rate, which effectively means that the pulse rate of the corresponding metrical levels are equal, but does not specify any constraints for the other beat subdivisions. In such a case, we propose two possible type of modulations (cf. below C1.1 and C1.2) to cover different scenarios. Conversely some definitions provided by Bouchard offer nuances for analysis of score notation, but are sonically indistinguishable. For instance, the Type III Metric Modulation does not relate to an audible change and the Type III Combined Modulation is indistinguishable from a Tempo Modulation.

Such definitions (e.g. the Type III modulations) are thus discarded in the transposed taxonomy.

A few categories are added on top of Bouchard's taxonomy in order to augment the descriptive power as well as the real life applicability of the taxonomy. Examples of such categories are for instance the 'No Meter Modulation' that captures transitions from a region without a clear sense of meter to a region with a clear meter and vice versa or the 'Other changes' category to flag modulations that do not correspond to a modulation with characteristic features such as the use of a pivot. A detailed description of the transposed taxonomy is given in the following.

All modulation types are illustrated by a diagram in Figure 6.12 to Figure 6.25. In these diagrams, the modulation (i.e. the segment boundary) is represented by the vertical axis. Horizontal lines on the left hand side of the vertical axis represent the pulse rates (in BPM) of metrical levels corresponding to the metrical structure before the modulation. Similarly, horizontal lines on the right hand side of the vertical axis represent pulse rates of the metrical levels after the modulation. Given that metrical level pulse rates relate to horizontal lines in a metergram, the diagrams used in this section may be interpreted as idealised metegrams for each modulation type. The characteristics of all modulations are then summarised in Table 6.1.

Tempo Modulation (TM)

A tempo modulation is characterised by a different tempo on each side of the modulation, while the relative metrical structure is preserved. Figure 6.12 shows an example metrical level pulse rates diagram for a tempo modulation.

Modulation Type 1 (T1)

In a Type 1 modulation, the beat rate and the metrical level rates lower the beat rate (e.g. the number of beats per bar) remain unchanged. The metrical levels with pulse rates greater than the beat rate (i.e. the beat subdivisions) change from one segment to the next. Figure 6.13 shows an example metrical level pulse rates diagram for a Type 1 Modulation.

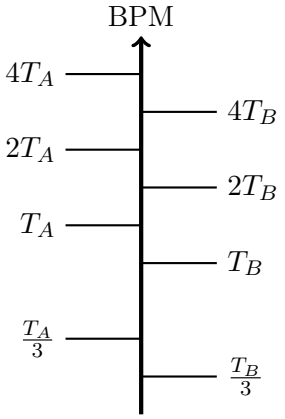


FIGURE 6.12: Example of Tempo Modulation where T_A and T_B are the beat rates before and after the modulation respectively

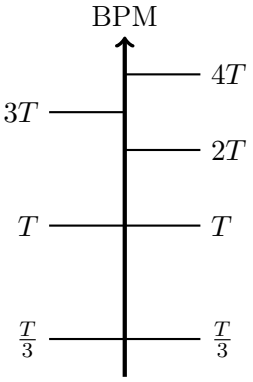


FIGURE 6.13: Example of Type 1 Modulation where T is the beat rate

Modulation Type 2 (T2)

The beat rate and the metrical level pulse rates greater than the beat rate remain unchanged. Only the metrical levels with rate lower beat rate (i.e. longer periodicities, such as bar length) are altered. Figure 6.14 shows an example metrical level pulse rates diagram for a Type 2 Modulation.

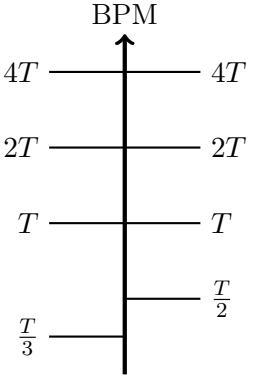


FIGURE 6.14: Example of Type 2 Modulation where T is the beat rate

Modulation Type 1 Hybrid (T1H)

The Type 1 Hybrid modulation combines characteristics of the Type 1 and Type 2 modulations. The tempo remains the same, but the number of beats in the bar as well as the subdivisions of the beat differ. Therefore the Type 1 Hybrid modulation is characterised by having only one metrical level in common between the two segments, and this level is the beat rate T. Figure 6.15 shows an example metrical level pulse rates diagram for a Type 1 Hybrid Modulation.

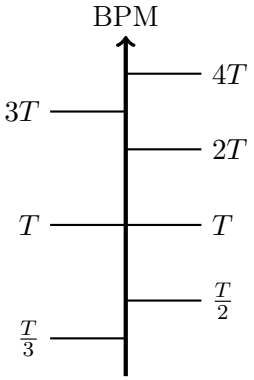


FIGURE 6.15: Example of Type 1 Hybrid Modulation where T is the beat rate

Combined Modulation Type 1.1 (C1.1)

The Combined Modulation Type 1.1 is encountered if the two segments have only one metrical level in common and that level has a pulse rate strictly greater than the tempo (i.e. be a beat subdivision). The beat rate of the two segments is therefore different. Figure 6.16 shows an example metrical level pulse rates diagram for a Combined Modulation Type 1.1.

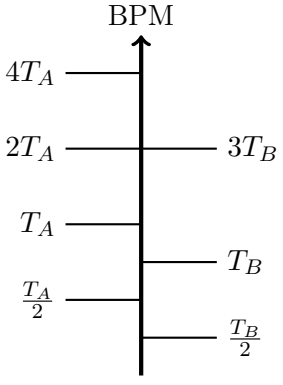


FIGURE 6.16: Example of Combined Type 1.1 Modulation where T_A and T_B are the beat rates before and after the modulation respectively

Combined Modulation Type 1.2 (C1.2)

In a Combined Modulation Type 1.2, the beat rates of the two segments are different but all the metrical levels with a pulse rate greater than the beat rate are identical. Figure 6.17 shows an example metrical level pulse rates diagram for a Combined Modulation Type 1.2.

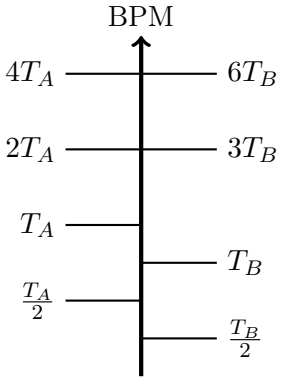


FIGURE 6.17: Example of Combined Type 1.2 Modulation where T_A and T_B are the beat rates before and after the modulation respectively

Combined Modulation Type 2.1 (C2.1)

In a Combined Modulation Type 2.1, the beat rates of the two segments are different and the two segments have only one metrical level rate in common. This common level has a pulse rate lower than the beat rate (e.g. the bar level). Figure 6.18 shows an example metrical level pulse rates diagram for a Combined Modulation Type 2.1.

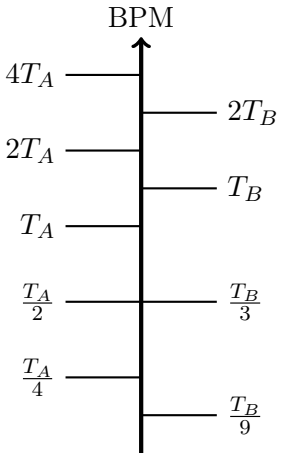


FIGURE 6.18: Example of Combined Type 2.1 Modulation where T_A and T_B are the beat rates before and after the modulation respectively

Combined Modulation Type 2.2 (C2.2)

A Combined Modulation Type 2.2 is characterised by the beat rates of the two segments being different and all the metrical levels with a pulse rate lower than the beat rate (e.g. the bar level) remaining identical. Figure 6.19 shows an example metrical level pulse rates diagram for a Combined Modulation Type 2.2.

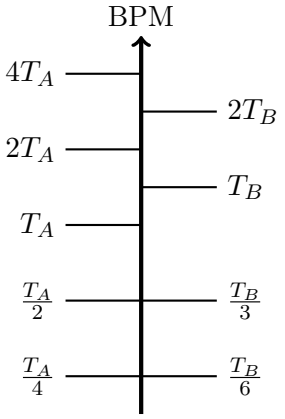


FIGURE 6.19: Example of Combined Type 2.2 Modulation where T_A and T_B are the beat rates before and after the modulation respectively

Subdivision Addition (SA)

A subdivision addition is characterised by the addition of an extra level of beat subdivision (therefore having the highest pulse rate) to the metrical structure. Figure 6.20 shows an example metrical level pulse rates diagram for a Subdivision Addition.

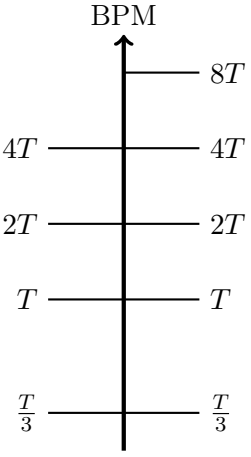


FIGURE 6.20: Example of Subdivision Addition where T is the beat rate

No Meter Modulation (NoMe)

This modulation characterises the transition from a segment with no meter to a segment with a clear meter and vice versa. Figure 6.21 shows an example metrical level pulse rates diagram for a No Meter Modulation.

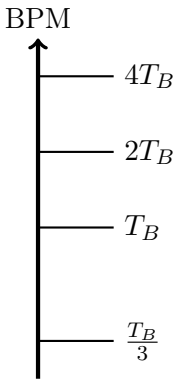


FIGURE 6.21: Example of Modulation from a segment with no clear meter to a segment with clear meter where T_B is the beat rate after the modulation.

Indeterminate Modulation (IM)

When a change of metrical structure that does not fit any modulation category, but features some characteristics of the Type 1, Type 1H or Type 2 modulations it is classified as an Indeterminate Modulation. In particular, at least one metrical level pulse rate is preserved (i.e. there is a pivot), and the beat rate is not modified. Figure 6.22 shows an example metrical level pulse rates diagram for an Indeterminate Modulation.

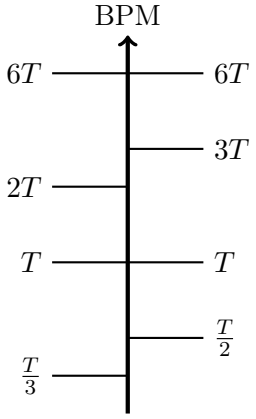


FIGURE 6.22: Example of Indeterminate Modulation where T is the beat rate

Indeterminate Combined Modulation (IC)

When a change of metrical structure that does not fit the into any modulation category, but features some characteristics of the combined modulations it is classified as an Indeterminate Combined Modulation. In particular, at least one metrical level pulse rate is preserved (i.e. there is a pivot), and the beat rate is altered. Figure 6.23 shows an example metrical level pulse rates diagram for an Indeterminate Combined Modulation.

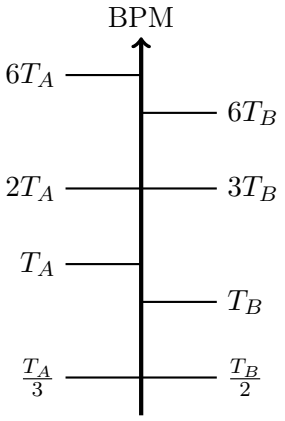


FIGURE 6.23: Example of Indeterminate Combined Modulation where T_A and T_B are the beat rates before and after the modulation respectively

Other Change (OC)

This category regroups all metrical structure changes by which no metrical level pulse rate is preserved and that does not fit in any other category. It implies that there is no pivot providing a sense of metrical structure continuity. The modulations falling in this category are therefore expected to sound more disruptive to a listener. As an example, a joint change of tempo and metrical structure in ratios that do not allow any metrical reinterpretation (i.e. prevents the existence of a pivot) would fall in this category. Figure 6.24 shows an example metrical level pulse rates diagram for an Other Change modulation.

No Modulation (NoMo)

In the eventuality that none of the modulations described above has been detected and therefore that no change of metrical structure is happening, the transition between the two segments is flagged as featuring 'No Modulation'. This class

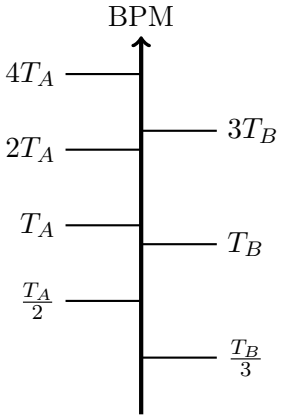


FIGURE 6.24: Example of Other Change Modulation where T_A and T_B are the beat rates before and after the modulation respectively

is added to enable recovery from spurious segmentation. Figure 6.25 shows an example metrical level pulse rates diagram for No Modulation.

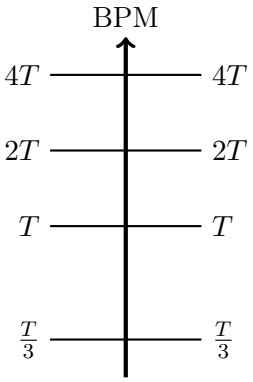


FIGURE 6.25: Example of No Modulation where T is the beat rate.

6.4 Metric Modulations Classification

In this section we employ the adapted metric modulations taxonomy presented in Section 6.3 in a metric modulation classification scenario. We assume that the segmentation of the music and the metrical levels pulse rates characterising the metrical structure of each segment are known. A hard classifier is derived by the direct implementation of the description of each modulation defined in the adapted taxonomy: each metric modulation can be classified with respect to this taxonomy, given the metrical pulse rates before and after the modulation.

TABLE 6.1: **Summary of adapted metric modulation taxonomy.** Each modulation is characterised by preserving or not some metrical levels. A modulation can preserve One, All, None of the metrical levels in question or be undetermined regarding them (represented by a '-' in the table). For SA, All* stands for 'All but one'. See Section 6.3 for a detailed description of all modulations

	Metrical levels preserved		
Modulation Type	Beat rate	Above beat rate	Below beat rate
TM	No	None	None
T1	Yes	None	-
T2	Yes	All	-
T1H	Yes	None	None
C1.1	No	One	-
C1.2	No	All	-
C2.1	No	-	One
C2.1	No	-	All
SA	Yes	All*	All
NoMe	No	None	None
IM	Yes	-	-
IC	No	-	-
OC	No	None	None
NoMo	Yes	All	All

In the following we consider two approaches: First we use the reference segmentation and metrical structure annotations (cf. Figure 6.26a) in order to compute a reference distribution of the metric modulation types across the dataset. Secondly, we use the segmentation from the reference annotation and the metrical structure automatically extracted to perform the classification (cf. Figure 6.26b). This result is then compared to the result obtained from the pure annotations, which enables the evaluation of the metric modulation classification performance when using automatically estimated metrical structure. We save the description of the automatic detection of the metric modulation boundaries (cf. Figure 6.26c) for Chapter 7.

6.4.1 Metric modulations classifier

We outline here a simple system to classify metric modulations based on the taxonomy presented in Section 6.3. In particular, the intent is to construct a classifier that directly implements the taxonomy. A flowchart representing the classifier is shown in Figure 6.27. The input consists of the metrical level pulse rates as well as the beat rate before and after the modulation. For each metric modulation, a boolean detector was created from its description formulated in the taxonomy, and is represented as a diamond in Figure 6.27.

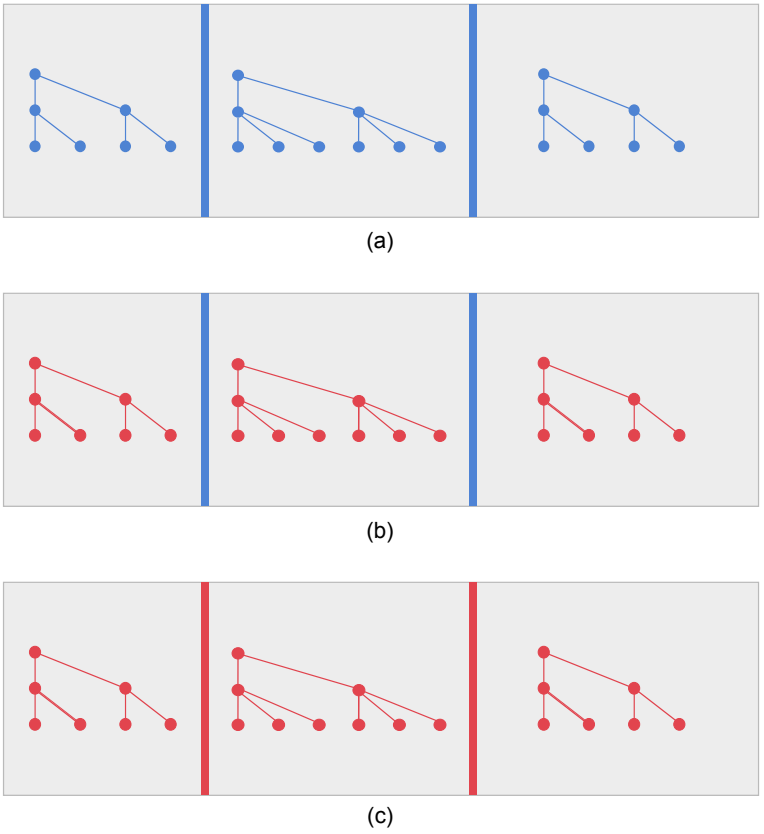


FIGURE 6.26: **Formal representation of the metric modulation tracking system and its evaluation.** The human annotations are represented in blue and features automatically extracted are represented in red. (a) The segment boundaries (vertical lines) and metrical structure for each segment are provided by human annotations. A reference metric modulation classification can be derived from these. (b) The tracks are segmented using human annotations, but the modulation classification is performed on the metrical structure computed automatically for each segment. (c) A representation of the ideal automatic system that produces both segment boundaries and metrical structures (and therefore metric modulation) that match the human annotations

Each modulation detector outputs a *True* value when the detection of a given modulation type is positive and *False* otherwise.

The C2.1 modulation detector is described in pseudo-code in algorithm 2. Since it is derived from the definition of the C2.1 modulation, we refer the reader to Section 6.3. The first step (line 1) consists in computing the metrical level pulse rates matches. This step aims at detecting which metrical level pulse rates are altered by the modulation and which ones are not, so that each level is attached with *match* or *non-match* Boolean value. This operation is performed by comparing the two metrical structures using the metrics described in Section 4.4.1. The true positives reveal a *match* (pulse rate non-altered), while the false positives and false negatives reveal an *non-match* (pulse rate

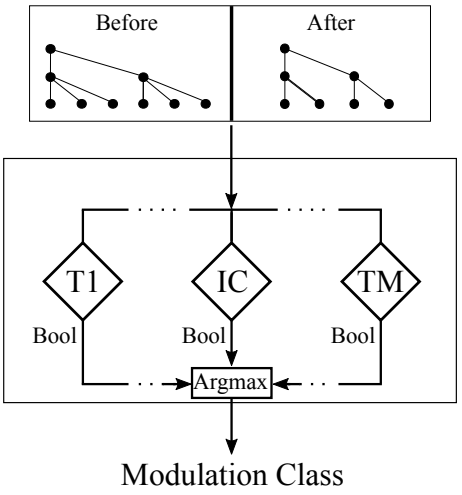


FIGURE 6.27: **Metric modulations classifier.** The input consists of the metrical level pulse rates before and after the modulation boundary. Each diamond represents Boolean metric modulation type detector. The final classification is derived from the overall result of individual classifications.

altered). Lines 2 and 3 guarantee that there are metrical level pulse rates detected before and after the modulation because their existence is implicitly assumed by the C2.1 definition. The definition of the C2.1 modulation also stipulates that the beat rate before and after the modulation must be different. This condition is tested in line 4 and 5: a metrical structure change cannot be a C2.1 modulation if the beat rate is not altered. Finally lines 6 to 12 first evaluate whether or not there is only one match (noted Ξ) and secondly if this match has a lower pulse rate than the beat rate before and after the modulation.

Algorithm 2 C2.1 Modulation Detector

Require: Metrical level pulse rates and beat rate before (T_B) and after (T_A) the modulation boundary

1: Compute metrical level pulse rates matches
2: **if** no pulse rate before or after modulation **then**
3: **return** False
4: **if not** $T_B \neq T_A$ **then**
5: **return** False
6: **if** there is only one match (Ξ) **then**
7: **if** $\Xi \leq T_B$ and $\Xi \leq T_A$ **then**
8: **return** True
9: **else**
10: **return** False
11: **else**
12: **return** False

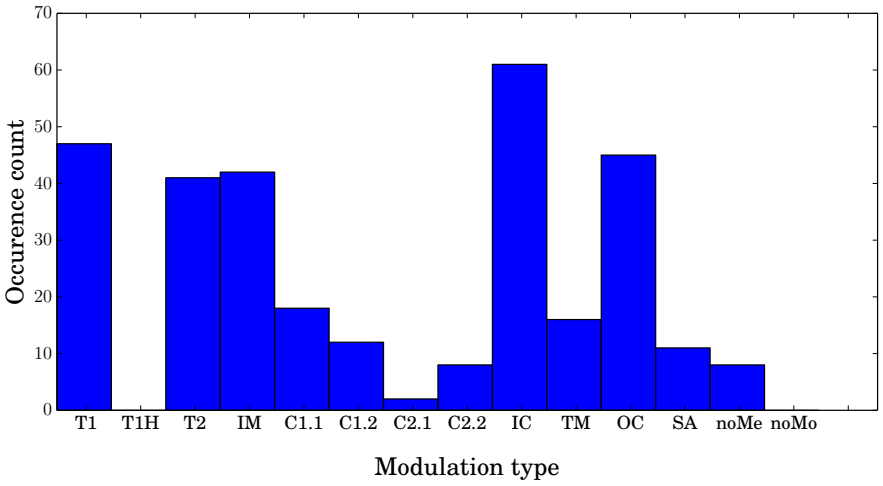


FIGURE 6.28: **Occurences count per metric modulation type in the reference annotations.** The metric modulation labels were obtained using the classifier described in Section 6.4.1 directly on the annotations in the dataset.

The algorithms for detection of all other modulations are obtained with a similar approach. Each metric modulation is tested against all taxonomy class detectors. Note that the taxonomy is defined so that metric modulation classes are orthogonal, i.e. there can be a *True* value only for one class. The modulation label is then obtained by finding which modulation detector output a *True* value, hence the argmax block in Figure 6.27.

6.4.2 Reference Annotations Content

The metric modulation dataset we use here includes annotations of the segment boundaries and the metrical structure of each segment. It therefore contains information about the nature of the metric modulations involved although it does not contain explicit modulation labels. In this section we intend to uncover what type of metric modulations are present in the dataset and in which proportions. In order to do so, all modulations are labelled using the classifier described in Section 6.4.1. The input to the classifier consists of the annotated metrical level pulse rates and the beat rate, calculated as the median of the inter-beat interval for each segment. In Figure 6.28 we present the resulting distribution of metric modulation types in the reference annotations.

It appears that the dataset does not seem to contain any examples of Type 1 Hybrid modulation. The absence of 'no Modulation' indicates good quality annotations: all the

annotated segment boundaries effectively correspond to a metrical structure change. In addition, the modulation classes added on top of Bouchard's taxonomy (i.e. 'Subdivision Addition', 'no Meter') are observed in the dataset, which validates their inclusion. It is also apparent that not all metric modulation types are equally represented in the dataset. For instance, the combined type 2.1 modulation is only encountered twice while there are 47 instances of Type 1 modulation.

More interestingly, we note that the three classes capturing relatively undefined modulations (i.e. 'Indeterminate Modulation', 'Indeterminate Combined Modulation' and 'Other Changes') represent nearly half (48%) of the modulations present in the dataset. Furthermore, the proportion of indeterminate to clearly determinate modulations is significantly larger in the case of combined modulations than in the case of tempo-preserving modulations (i.e. T1, T1H, and T2). This observation is not surprising, however. The combined modulations imply a change of tempo and of some of the metrical structure while preserving a small part of the metrical structure (as little as one metrical level pulse rate), which allows a much larger number of possible combinations than in the case of tempo-preserving modulations. Despite being labeled as 'indeterminate', these classes are nonetheless informative as they capture characteristic modulation features. For instance, the Indeterminate Combined Modulation class regroups modulations generating a change of beat rate and using a pivot (i.e. maintaining some metrical continuity) while the Indeterminate Modulation class regroups modulations that preserve the tempo and alter some metrical levels. As a result, from a musical standpoint, it may be expected that an Indeterminate Combined Modulation generates a stronger perceptual effect than an Indeterminate Modulation.

The sizeable proportion of modulations assigned to indeterminate classes suggests that the modulation types defined in the taxonomy do not exhaustively capture the variety of modulations present in observable data. As a result, this suggest that there is room to develop the taxonomy further. A greater exhaustivity could for instance be achieved by including more sub-classes. However, what should the new categories be? On which basis should they be specified? These are open research questions. In order to make such an extension musically meaningful it is desirable that it is rooted in music theory, and

perhaps music psychology. Consequently, it is probably desirable that these questions are addressed in collaboration with musicologists in future work.

6.4.3 Classification from extracted features of known segments

In this section we aim to evaluate the use of automatically extracted metrical structure for metric modulations classification. To serve this purpose, we use the segment boundaries from the reference annotations along with the metrical structure extracted automatically, which is schematically illustrated by Figure 6.26b. This is then used as an input to the metric modulation classifier described in 6.4.1. Using the annotated segment boundaries guarantees that the segments under consideration in this experiment are identical to those used in Section 6.4.2. As a result, comparing the classification results obtained in this condition with the classification obtained in Section 6.4.2 (we recall it was obtained using segment boundaries and metrical structure from the reference annotations) enables the evaluation of the impact of automatic extraction of metrical structure.

Since we focus here on abrupt metrical structure changes, which locations are indicated by the segment boundaries, the metrical structure is assumed to be consistent within every segment. We first extract the average metrical level pulse rates using the method described in Chapter 4. In addition, the beat rate of each segment is estimated using the Vamp plugin implementation[1] of the algorithm proposed by Davies et al. in [121]. The combination of beat rate and metrical level pulse rates constitute the information necessary to classify the metric modulation types using the classifier presented in Section 6.4.1. As a result, each modulation is classified with respect to the taxonomy presented in Section 6.3. For each segment boundary (i.e. each metric modulation), we then compare the classification produced from the automatically extracted metrical structure with the classification produced using reference annotated metrical structure. The results of this comparison are presented as a confusion matrix in Figure 6.29.

Entries on the diagonal indicate the number of instances of agreement between classification generated from reference annotations and from automatically extracted features. In other words, the same modulation class was affected to a given segment using the

[1] http://www.vamp-plugins.org/download.html

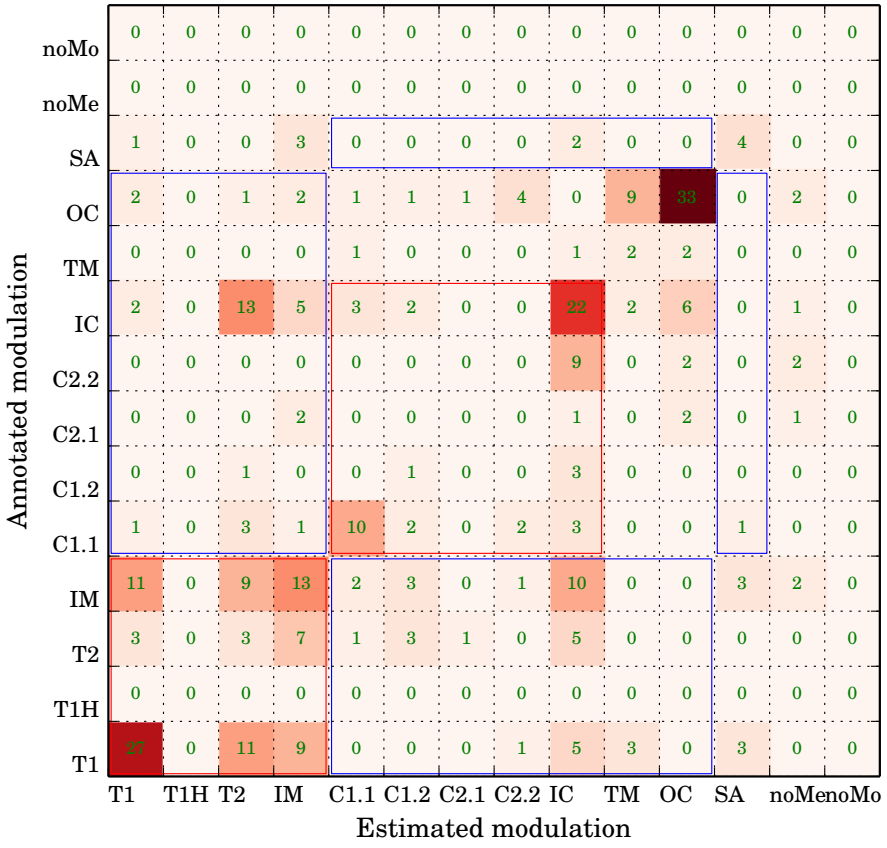


FIGURE 6.29: **Metric Modulations Confusion Matrix**. For each segment boundary, this matrix represents the comparison between the modulation classification obtained using reference annotations and the modulation classification obtained using automatically estimated metrical structure. Diagonal entries reveal 'correct' classifications, while off-diagonal entries reveal confusions. The areas marked by blue boxes imply an incorrect beat rate estimation. Areas delimited by red boxes include correct classification and confusions within a family of metric modulations.

TABLE 6.2: Metric Modulation confusion statistics

	Occurences	%
Correct	115	37.0
Intra-family confusion	75	24.1
Necessary beat rate error	71	22.8
Other errors	50	16.1

annotated and the automatically estimated metrical structure. These may then be referred to as 'correct' classifications. Conversely, off-diagonal entries reveal the number of segment boundaries for which the classification obtained using annotated metrical structure differs from the classification obtained using the automatically extracted metrical structure. These may then be referred to as 'confusions' (or 'incorrect' classifications).

The confusion matrix reveals that 37% of the classifications are correct. No confusion arises with the 'no Modulation' class. Given that the reference annotations for segment boundaries always correspond to a metrical structure change (cf. Section 6.4.2), this result suggests that the automatic feature extraction reliably captures metrical structure changes. In other words, even if the nature of the metric modulation may be incorrectly classified, the system captures the presence of a change.

Some confusions may be grouped into clusters on the basis of their similar nature. The corresponding areas of the confusion matrix are delimited by rectanglular boxes with colour corresponding to a particular confusion group. Some statistics based on these confusion clusters are given in Table 6.2. The original taxonomy proposed by Bouchard groups the metric modulations in two families: the 'Metric Modulations' (i.e. T1, T1H, T2, and by our extension IM), and the 'Combined Modulations' (i.e. C1.1, C1.2, C2.1, C2.2 and IC). The red boxes in Figure 6.29 mark areas corresponding to confusion between metric modulations belonging to the same family, which are referred to as 'intra-family confusion' in Table 6.2. Within each family, a sizeable proportion of the confusions occur between the indeterminate class (IM and IC respectively) and the other classes. This observation is easily explained by the hard selectivity of the metric modulation classes characteristics. As a matter of fact, if as little as one metrical level pulse rate is incorrectly detected (irrespectively of being a false negative or false positive) the classification decision may go from one class to another. The number of correct classifications and the number intra-family confusions add up to 61.1% of the total number of classification decisions and effectively represents the cases in which at least the correct family is chosen. This result suggests that the classification is more robust for coarser granularity, i.e. when classifying modulation in families rather than against a more fine grained taxonomy.

A direct consequence of the architecture of the metric modulation taxonomy is that the confusion between certain classes necessarily implies that the beat rate estimation is incorrect (i.e. does not match the annotated reference) in at least one of the two segments. In particular, a range of modulation types are characterised by the preservation of the beat rate (e.g. T1, T2, SA etc.), while some others are explicitly characterised by a change of beat rate (e.g. all combined modulations, OC, etc.). Then, the confusion of a modulation type preserving the beat rate with a modulation type characterised a beat rate change necessarily implies that the beat rate has been incorrectly estimated on at least one side of the modulation. The areas of the confusion matrix corresponding to such confusions are marked by the blue rectangles, and referred to as 'necessary beat rate error' in Table 6.2. It is to be noted, however, that incorrect beat rate estimation may occur in other zones of the confusion matrix, although it is not a necessary condition. The 'Necessary beat rate errors' account for a significant proportion of all confusions (22.8%). The beat rate estimation algorithm may be incriminated for this result and a first order conclusion to be drawn from this observation may be that improving the beat rate estimation would have potential to very significantly improve the classification results.

However, as discussed in Chapter 3, one of the limits of notions such as tempo and beat is that even human experts tend to disagree when producing annotations. As a consequence, it is hard, if possible at all, to provide an annotation of the 'true' or 'correct' beat rate. The taxonomy considered here is also affected by this limitation because it relies on the provision of the beat rate. Moreover, the current taxonomy is a relatively direct transposition of a taxonomy originally designed for score-based analysis, which implements hard classification constraints (boolean tests, essentially). Typically, symbolic data is free of noise and of relatively clearly established semantics, so that the application of hard constraints is effective in this context. In contrast, audio data is typically noisy and extraction of semantically meaningful information from it is difficult, therefore resulting in an error prone process. A possible avenue for future work could be to investigate the design of a taxonomy based on softer classification constraints. This would, however, require a reformulation of the underlying musicological concepts in terms that are compatible with the aforementioned softer constraints, which is not

trivial. Again, it is probably desirable that such a research question is addressed in collaboration with musicologists in order to maximise its musical meaningfulness.

6.5 Conclusions

In this chapter we proposed the use of a musicologically-grounded taxonomy as a strategy for classifying metric modulations in a musically meaningful way. We first summarised in English a taxonomy inspired by the analogy between theory of harmony and metrical structure originally formulated in French by Bouchard. Since this taxonomy was designed for score-based analyses it is not directly applicable in an audio-based scenario in which the score is not available. On the other hand, in a preliminary study, Benadon proposed to use the metrical level pulse rates to characterise metric modulations. He did not formulate a metric modulation taxonomy, neither did he attempt automatic classification, however. As a consequence we proposed a new metric modulations taxonomy obtained by transposing Bouchard's taxonomy in a framework relying on metrical level pulse rates, as suggested by Benadon. A metric modulation classifier was then proposed on the basis of the newly introduced taxonomy. The metric modulation classification obtained with this classifier using either the reference annotations or automatically extracted features allowed us to draw several conclusions: First of all, the relatively direct transposition of Bouchard's taxonomy performed here seems to be insufficient to capture all the variety of modulations present in the dataset. This suggests that in order to achieve an exhaustive description of metric modulations from audio recordings, the taxonomy would need to be extended (to include more classes). Nevertheless, it has also been shown that the classification performance is significantly higher when only considering metric modulation families (i.e. a less refined taxonomy); thus providing an informative classification. Moreover, the classification considered here relies on hard classification constraints (i.e. Boolean decisions), which is an error prone process. We therefore note that these promising results might be improved by formulating a taxonomy of metric modulations based on softer constraints. However, the construction of such a taxonomy require a redefinition of the underlying musical concepts in terms compatible with softer constraints, which is

an open research question that ought to be addressed in collaboration with musicologists in future work.

Chapter 7

Towards the Automatic Detection of Metric Modulations

7.1 Introduction

The presence of a metric modulation implies a transition from one metrical structure to another over time. Formally, the detection of metric modulations may be broken down in two sub-tasks: detecting a change and identifying the nature of the change - e.g. with respect to a taxonomy. Latent state space models, such as the bar pointer model originally introduced in [27], have become popular in the recent years to infer rhythmic structure of music and are theoretically capable of tracking its changes. This model includes three hidden variables: the *bar position*, defined as the position of the current audio frame with respect to the bar cycle (therefore independent of tempo), the *tempo*, defined as the time derivative of the bar position, and the *rhythmic pattern* representing the likelihood of onset with respect to the position in the bar cycle [28]. A range of variations have been proposed to extend the scalability of the system by amending the state space [96] or using particle filters [28, 111]. The bar pointer model represents changes in metrical structure by transitions between states representing different bar length (i.e. number of beats in a bar) or rhythmic pattern [27]. Other latent state space models for rhythm tracking typically also represent changes in metrical structure by transitions between states [215]. The model parameters, such as bar length and

rhythmic patterns, are not known a priori and are therefore typically set manually based on expert knowledge, musical structure hypothesis, or by supervised learning from data. Similarly, the structure of the model (e.g. the number of rhythmic patterns to be learnt) has to be manually set. As a result, this type of approach is suitable for classification tasks in which the number of classes is known or for which a robust hypothesis can be made; and for which a large amount of annotated reference data is available, but impractical otherwise. Authors have typically used this type of method for beat and downbeat tracking on corpora having a known range of metrical structures [28, 35].

Even if the computational models in use have this capability, changes of metrical structure over time are rarely considered and when they are, tracking of metric modulations is not typically considered as a task in itself. We propose here to focus on the detection and classification of metric modulations. In particular, we consider an application scenario in which neither the number nor the nature of the metrical structures to be encountered in a piece are known a priori. In contrast with the supervised learning systems such as the beat pointer model mentioned above, we propose an unsupervised method for automatic metric modulations detection.

We formulate the detection of metric modulations as a segmentation problem. Indeed, the transition between two sections of different metrical structure outlines a form segmentation of the musical piece, as shown on Figure 7.1. For ease of reading, we refer to *metric modulation-based segmentation* simply as *segmentation* for the remainder of this chapter. We restrict the scope of the current study to metric modulations that feature a stable metrical structure before and after the modulation as well as a relatively abrupt transition between the two. In other words, we do not consider slow and/or progressive alterations of the metrical structure (e.g. progressive tempo change or gradual metrical structure change via long transitory sections), so that transitions between two segments may be modelled as discrete boundaries. As such, this problem formulation is analogous to the structural segmentation retrieval task, well documented in the MIR literature (see for example [7]). Musicians and musicologists commonly use the terms 'structure' or 'form' to refer to the high-level layout that divides a piece into sections. In the context of popular music, these sections are typically labelled using terms such

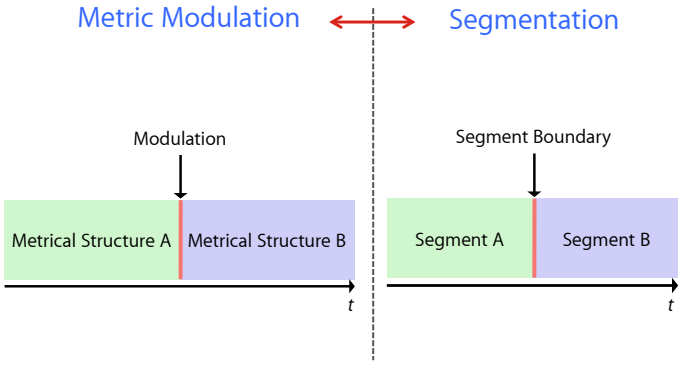


FIGURE 7.1: **Problem definition: metric modulation detection as a segmentation task.**

as 'verse', 'chorus' and 'bridge'. It is usually considered that the sections constituting the musical form exhibit some self-consistency with respect to some musical features, such as instrumentation, harmony, melody or rhythmic patterns to name but a few. The inherent open-ended nature of such a description jointly accounts for the wide degree of liberty at the composer's disposal to realise structural segmentation and the resulting difficulty to automatically retrieve such a structure. Automatic structural segmentation algorithms are therefore inevitably built on the assumption that the musical form is made apparent by a given musical feature or a set thereof. For instance, assuming that the structural segmentation is underpinned by harmonic progressions, using a feature capturing the evolution of harmonic content over time (e.g. a chromagram) is one option [172, 216]. Other features will naturally capture different musical properties, such as timbre and may also be useful for structural segmentation, e.g. mel spectrogram [165] or spectral envelope [155]. Since the musical attributes used by the composer to realise the structural segmentation are usually not know a priori, several authors have proposed methods based on the combination of a number of audio features [169, 173].

For the problem we are interested in here, prior information regarding the strategy used by the composer is not available either. However, the definition of the problem provides a strong constraint that is a metric modulation implies a change in the metrical structure. In this case, audio features capturing the evolution of the metrical structure over time may be relevant for this task. In particular, we introduce the metergram as a feature from which we might recover the segmentation. It consists of the combination of rhythmogram which has been shown to capture metrical structure information in Chapter 4, and its

computation is described in Section 7.2.1. In order to retrieve the segmentation from the metergram in an unsupervised fashion we propose the use of Non-negative Matrix Factorisation (NMF) as a frame clustering technique. Several varations of this approach are explored in Section 7.3 and we refer the reader to Section 2.6 for a description of the bascics of NMF. We also investigate the use of a novelty-based approach that is commonly used for structural segmentation in Section 7.4. By construction, all methods presented in this chapter are off-line an non-causal, therefore not suitable for real-time processing in their current form.

The segmentation and the classification of metric modulations are evaluated separately. All the experiments reported here have been carried out on the dataset introduced in Section 3.6. Firstly, a reference classification of modulation is produced using the human annotations and the adapted taxonomy (Figure 6.26 a). Then, given the annotated segmentation, we evaluate the automatic classification of the metric modulations using the adapted taxonomy and the metrical levels pulse rates extracted with the method presented in Chapter 4 (Figure 6.26 b) in Section 6.4. In a separate experiment, we propose and evaluate a range of methods to automatically retrieve the metrical structure-based segmentation in sections 7.4 and 7.3. An automatic metric modulation tracking system is considered ideal if it accurately reproduces both the segmentation and the classification of metric modulations (Figure 6.26 c).

In the remainder of this chapter, we first present the different signal processing building blocks and segmentation algorithms under scrutiny and illustrate their respective properties on a single example in Section 7.2 to 7.4. The example track we use for illustration purpose is “Geno (Tribute to Dexys Midnight Runners)” by Union of Sound and its metergram with overlaid annotated metric modulation boundaries is given in Figure 7.2. The track starts with a short introduction and then develops as the alternation between two parts of distinct metrical structure. It is therefore made of three metrically distinct parts. The two alternating parts are respectively in compound and simple meter. We then compare all algorithms on the entire dataset using a range of metrics, while parameters for secondary steps not expected to produce much difference across the different

algorithms (e.g. metergram window size, and P_d, cf. below) were set in preliminary studies.

7.2 Feature pre-processing for automatic metrical structure change detection

7.2.1 Metergram

Initially suggested by Peeters [93], the multiplication of the Fast Fourier Transform (FFT) and Autocorrelation Function (ACF) based periodicity spectra has been shown in Chapter 4 to be effective to filter out harmonics in the resulting spectrum so that its peaks more closely relate to the metrical structure of the music. Such composite beat spectra have so far only been used as a summary feature (i.e. averaged over time), however. Here, we propose to extend this approach to the use of a *metergram*, in which every frame is the product of FFT and ACF beat spectra proposed by Peeters. Since the peaks in the periodicity spectra (i.e. the metergram frames) relate to metrical level pulse rates (cf. Chapter 4), the metrical structure relates to a structure of horizontal lines in the metergram, each line typically corresponding to a metrical level pulse rate. As a result, changes in metrical structure over time are expected to manifest as apparent changes of structure in the metergram.

The computation of the metergram is identical to the calculation detailed in Chapter 4, with the addition of a logarithmic rescaling of the frequency axis and frame normalisation. We briefly summarise this calculation in the following. We first compute a spectrogram of the audio signal sampled at 44.1kHz using a Hann window of 512 samples and a step size of 256 samples. An onset detection function is derived using the *superflux* method with the parameter values recommended by the authors [62]. We then compute two rhythmograms $\mathbf{R}_F$ and $\mathbf{R}_A$, based on the FFT and ACF of the windowed onset detection function respectively, using 12s Hann windows with 0.24s overlap. The ACF rhythmogram $\mathbf{R}_A(l, n)$ is mapped to the frequency domain $\mathbf{R}_A(m, n)$, with $l = f_s/\omega$ where l is the lag in the ACF, and ω is the corresponding rate associated to frequency bin

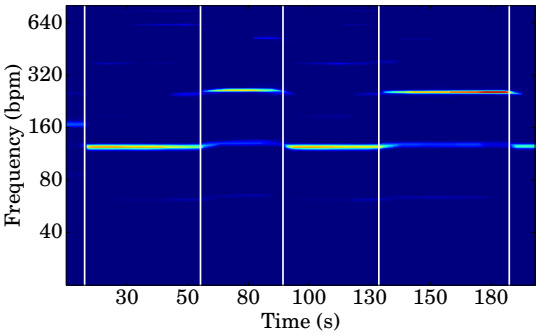


FIGURE 7.2: Metergram with annotated segment boundaries overlaid for the track "Geno (Tribute to Dexys Midnight Runners)" by Union of Sound.

m and f_s is the onset detection function sampling frequency, as initially proposed in [93]. The metergram is then computed as the element-wise product of the two rhythmograms:

$$\mathbf{R}(m,n) = \mathbf{R}_F(m,n) \odot \mathbf{R}_A(m,n) \tag{7.1}$$

The relative structure of horizontal lines in the metergram (i.e. the relationships between metrical level pulse rates) characterises the metrical structure of music. In other words, a given metrical structure results in a pattern of metrical level pulse rates, just like a harmonic sound results in a pattern of partials in a spectrogram. On a logarithmic frequency scale, the shape of the corresponding horizontal lines pattern is independent of the fundamental frequency in a spectrogram, which is a desirable property for further analysis of these patterns [217]. Similarly, using a logarithmic pulse rate scale in a metergram implements invariance of the shape of the pattern of metrical level pulse rates against speed of execution of a piece. The bins of the metergram are then re-assigned to a logarithmic rate scale so that the rate corresponding to the m^{th} bin is:

$$\omega_m = \omega_0 \times 2^{\left(\frac{m}{\rho}\right)} \tag{7.2}$$

where $\omega_0 = 20$ BPM and $\rho = 100$ bins/octave. Additionally, each frame of the metergram is normalised by its L_1 norm:

$$\hat{\mathbf{r}}_n = \frac{\mathbf{r}_n}{||\mathbf{r}_n||_1} \tag{7.3}$$

where $\mathbf{r}_n$ is the n^{th} metergram frame.

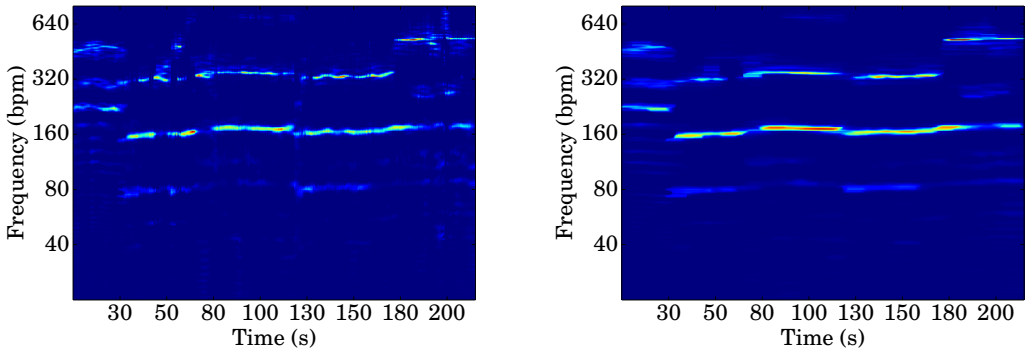
Metric modulations are expected to be related to the alteration of metrical level pulse rates over time (cf. Chapter 6), which are expected to correspond to discontinuities in the horizontal lines structure of the metergram. Figure 7.2 illustrates this on an example taken from the metric modulations dataset introduced in Section 3.6, where the annotated modulation boundaries are represented by the white vertical lines. In this chapter we seek to automatically recover these changes of structure in the metergram.

7.2.2 Horizontal median filtering

The energy distribution characteristic of a clear metrical structure that is expected in a metergram consists of horizontal lines and therefore resembles a harmonic structure in an audio spectrogram. This chapter is concerned with the automatic detection of metric modulations, which we formulate here as the detection of changes in the structure of horizontal lines in the metergram over time. The diagrams given in Section 6.3 represent idealised metergram structures for a variety of metric modulations: an abrupt change from one structure of horizontal lines to another. Naturally, metergrams computed from musical recordings are more noisy than their idealised counterparts. Broad band energy distributions and noisy components that form vertical (or at least non-horizontal) structures in the metergram, analogous to percussive events in an audio spectrogram, are not informative about the metrical structure and may therefore be removed because they may interfere with the segmentation algorithms. Fitzgerald proposed a method based on median filtering to perform separation of harmonic and percussive components of an audio signal that effectively consists in filtering horizontal and vertical lines, and which is briefly described in Section 2.8. Applying this method to the metergram instead of the magnitude spectrogram enables the enhancement of the structure of horizontal lines in the metergram. Replacing the audio spectrogram $\mathbf{X}$ by the metergram $\mathbf{R}$ in equation (2.35), a new metergram in which the horizontal lines ("harmonic") structures of interest are enhanced is created:

$$\bar{\mathbf{r}}_m = \mathcal{M}\left\{\hat{\mathbf{r}}_m, \ell\right\} \tag{7.4}$$

where $\hat{\mathbf{r}}_m$ is the m^{th} normalised metergram frequency slice and ℓ is the length of the median filter. The length of the median filter controls the temporal extent over which



(A) Metergram before median filtering (B) Metergram after median filtering

FIGURE 7.3: Metergram before and after horizontal lines enhancement by median filtering for the track 'One Rainy Wish' by The Jimi Hendrix Experience

the filtering occurs. As a result, horizontal lines whose extent is equal to or greater than the median filter length are preserved while energy distributions that do not form a horizontal line or form a horizontal line for a duration noticeably shorter than the median filter length are eliminated. Given that forming a meter perception takes a few seconds [19], we assume sections of consistent metrical structure to be at least around 10s long. Then, successive metric modulations are assumed to be separated by 10s or more. On this account, the median filter length must be set in the order of 10s so that changes of energy distribution stemming from metric modulations, i.e. alteration of horizontal lines that were relatively stable for a period of 10s or more, while other alterations are removed. Preliminary experiments have shown that modifying the median filter length by a few seconds did not significantly affect the resulting metergram. The filter length is then set to $\ell = 15$s. Note that as opposed to the audio spectrogram, the metergram is not invertible because it is computed as the product of two rhythmograms.

Figure 7.3 illustrates the effect of the application of the median filter on the metergram. As expected, the vertical energy distributions and noisy components are removed. See for example some vertical energy distributions around the 200s mark (especially visible around 640BPM) present in Figure 7.3 (A) and removed in Figure 7.3 (B). Small and local energy fluctuations along the frequency axis are also stabilised by the application of the median filter. These correspond to small and local changes in metrical level pulse rates and therefore represent local timing instabilities or expressive timing. See for example the horizontal line around 320BPM and between timestamps 130s and 180s in Figure 7.3. The

rate of the corresponding metrical level appears to slightly fluctuate before filtering and is stabilised by application of the filter. Note that alterations of energy distribution that occur on a longer time scale, typically corresponding to sections of consistent metrical structure, are preserved by the filter. In other words, the manifestation of the metric modulations are preserved. In the remainder of this chapter and unless stated otherwise, we exclusively use the median filtered metergram, and refer to it simply as the metergram for conciseness.

The use of the vertically enhanced metergram, which can be expected to specifically capture the information that is removed from the horizontally-enhanced metergram, is not considered in this study. Although the local fluctuations are interesting in their own right, their study is beyond the scope of this work. We note that this may be an interesting avenue for future work, however. The vertical structures in the metergram reveal the absence of a clear periodicity in the novelty curve for the window under analysis. Thus it can be hypothesised that this phenomenon may have several origins such as soft onsets, a lack of clear pulse, micro-timing, local inconsistencies and fluctuations or a somewhat chaotic rhythm to name only a few.

7.3 Novelty-based automatic metrical structure change detection

Metric modulations may be characterised by a change of energy distribution in the metergram frames. Figure 7.2 illustrates such changes. One approach to segmentation consists of computing similarity between metergram frames and characterising segment boundaries as significant dissimilarities over time. This type of approach is commonly referred to as *novelty*-based segmentation [7, 167]. Foote introduced a novelty-based method that has since become a standard for automatic structural segmentation [63]. Here, we adapt it to the detection of metric modulations by applying it to the metergram. This method is then regarded as a baseline for novelty-based methods in our experiments.

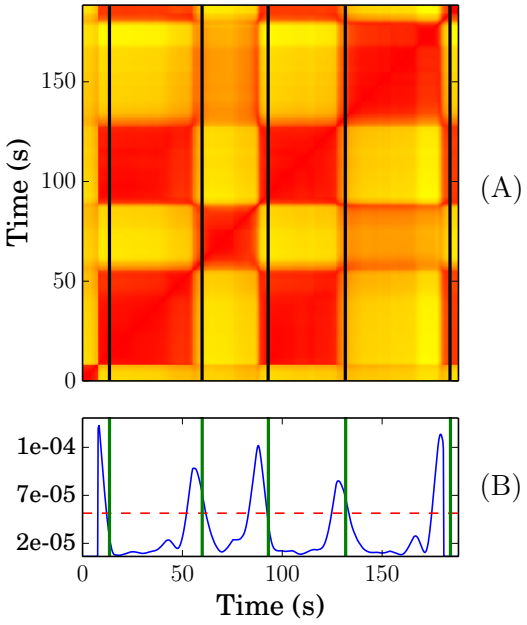


FIGURE 7.4: **Example of Self-Similarity Matrix (A) and resulting Foote novelty curve (B) with reference segment boundaries annotations overlaid as vertical lines.** The SSM (A) and novelty curve (B) are computed for the track "Geno (Tribute to Dexys Midnight Runners)" by Union of Sound. The horizontal dotted line in (B) represents an example of possible hard threshold for retrieving segment boundaries by peak-picking the novelty curve

First, we compute a Self-Similarity Matrix (SSM) of the metergram, using the Euclidian distance as a similarity measure between frames, so that:

$$b_{i,j} = ||\mathbf{r}_j - \mathbf{r}_i||_2 \tag{7.5}$$

where $\mathbf{r}_j$ and $\mathbf{r}_i$ are the j^{th} and i^{th} metergram frames respectively, $b_{i,j}$ is the corresponding entry in the self-similarity matrix $\mathbf{B}$ and $||\cdot||_2$ denotes the Euclidian distance operator. Although presented here using the euclidian distance, the self-similarity matrix may be computed using other distance measures. Foote suggested the cosine distance as a possible alternative that is invariant to the norm of the frame vectors considered ($\mathbf{r}_j$ and $\mathbf{r}_i$ in our case). In order to maximise comparability of our results with existing work, which predominantly employs euclidian distance, we use the Euclidian distance to compute the SSM. We note that informal experiments have suggested that the cosine and euclidian distance produce comparable results, owing to the L_1 normalisation applied to the metergram frames.

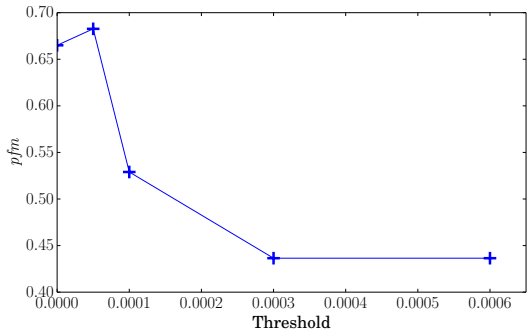


FIGURE 7.5: **Average segmentation performance as a function of the peal-picking threshold.** The *pfm* segmentation score is computed for every track of the dataset and the average value is presented here.

We recall from Section 2.10 that the size of the checkerboard kernel determines the timescale at which the novelty may be captured. As a consequence, small checkerboard allow computation of novelty at a short timescale, for instance suitable for onset detection, while larger (typically by a few orders of magnitude) checkerboards are suitable for structural segmentation. In addition, the fine adjustment of the checkerboard size controls the smoothness of the novelty curve produced, longer kernels yielding smoother curves. In this work we assume that sections of consistent metrical structure will be at least around 10s long and segment boundaries are expected to manifest as peaks in the novelty curve. Because the boundary positions are to be retrieved by peak-picking, it is desirable to produce a smooth novelty curve. As a result, the checkerboard kernel length was set to 15s.

In our experiments, a simple peak-picking algorithm[1] followed by a hard thresholding stage are used, in order to only keep peaks which magnitude is greater than the threshold (represented by the horizontal dashed line in Figure 7.4 B). Boundary locations are then retrieved as the timestamp of the peaks. Because the optimal threshold value is not known a priori, the segmentation was computed for a range of threshold values. Figure 7.5 shows the average *pfm* score obtained across all tracks of the dataset for a range of threshold values. For clarity, in the following we only report the results obtained with the threshold value ($=5.10^{-5}$) resulting in the highest average performance across the dataset.

[1] A sample is considered as a peak if its magnitude is greater that both the previous and next (along the time dimension) sample. We use the implementation from the SciPy library v.0.15.1 via the function scipy.signal.argrelmax()

Figure 7.4 shows an example SSM and its corresponding novelty curve for the track "Geno (Tribute to Dexys Midnight Runners)" by Union of Sound (the metergram of which is shown in Figure 7.2). A typical structure of diagonal and off-diagonal rectangles is observed in the SSM. The darker the colour, the higher the similarity between metergram frames. The diagonal squares reveal the temporal segmentation of the piece, while the off-diagonal dark rectangles reveal similarity between sections spanning different parts of the piece, i.e. repetition of similar metrical structures [63]. Segment boundaries are located at the corners joining two adjacent squares and the area of the square characterises the length of a section of consistent metrical structure. Figure 7.4 shows an example of result obtained using this approach. It appears that the structure made apparent by the SSM correlates with the reference annotations. Similarly, the novelty curve exhibits a structure that appears to be in good agreement with the reference annotations, both in terms of number and location of the peaks. We save the details of the quantitative analysis of the segmentation performance evaluation using this novelty-based method for Section 7.4.6.

7.4 Homogeneity-based automatic metrical structure change detection

In this section, we consider a range of methods to perform *homogeneity*-based segmentation. Here again, we hypothesise that the energy distribution of the metergram frames (i.e. the beat spectra) reflect the metrical structure, so that metric modulations would result in changes in the metergram with segments of consistent metrical structure resulting in homogeneous regions in the metergram. In this context, similar metergram frames are to be clustered together so that segments can be defined as contiguous regions in which the frames belong to the same cluster. Although we refer the reader to Section 2.6 for a description of NMF, we recall that it is known for its ability to learn "parts of objects" in an unsupervised fashion [127] and has previously been proposed to retrieve structural segmentation [218]. Here, we propose to use NMF to learn a decomposition of the metergram $\mathbf{R}$ in order to retrieve the homogeneity-based segmentation.

Consequently, by using NMF to chose **W** and **H** so that

$$\mathbf{R} \approx \mathbf{WH}, \tag{7.6}$$

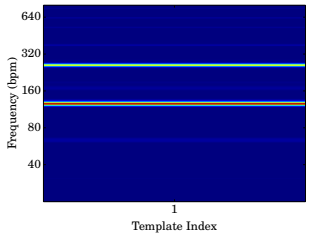
the 'parts' we aim at recovering here are the beat spectra characteristic of the metrical structure of each segment, represented by the templates in matrix **W**. Then, the matrix **H** is expected to reveal the structural segmentation by specifying when the templates are being activated, i.e. when the parts are occurring.

A number of variations on the standard NMF algorithm have been proposed in the literature in order to favour different properties of the learnt decomposition, i.e. to favour certain properties in matrices **W** and/or **H**. We consider a number of these variations in the following as well as the standard k-means clustering algorithm as a baseline. In order to illustrate the differences between the various approaches, we use one music piece from our metric modulation database as an example. From a metrical structure point of view, this track is made of three different parts: a short introduction, and two main alternating parts. The median filtered metergram of this track is given in Figure 7.2, with the segment boundaries reference annotations overlaid.

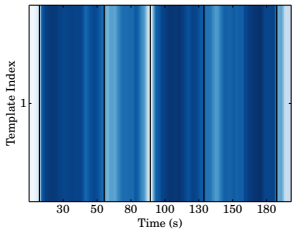
7.4.1 The rank estimation problem

In the NMF framework, the number of templates K is a fixed parameter — also known as the *rank* of the decomposition — that needs to be specified in advance in order to carry out the matrix factorisation. In the following we show that the choice of this parameter has critical impact on the properties of the learnt decomposition and therefore on its possible interpretation in terms of segmentation.

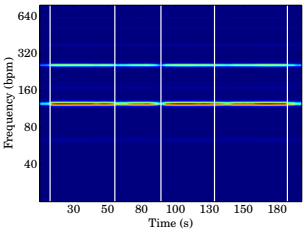
Figure 7.6 shows the template **W**, activations **H** and reconstructed **WH** matrices learnt by NMF to approximate **R** for a range of K. Overall, it appears from the observation of the reconstructed matrix that the metergram can be reconstructed with good accuracy via NMF decomposition with only a few templates (4 or more in this case). This accounts for a very high level of dimensionality compression in the sense that $MK + KN \ll MN$.



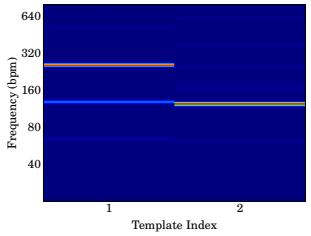
(A) **W** with $K = 1$



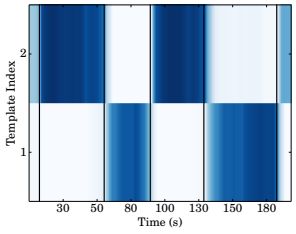
(B) **H** with $K = 1$



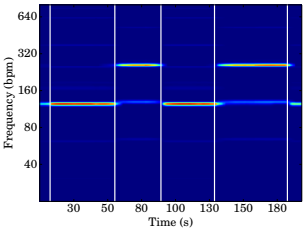
(C) **WH** with $K = 1$



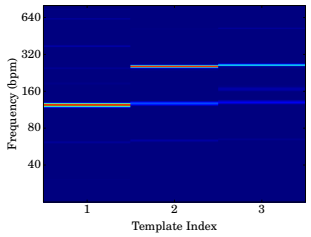
(D) **W** with $K = 2$



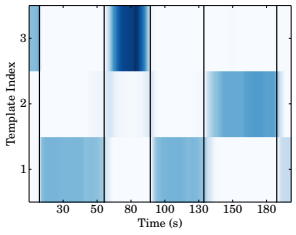
(E) **H** with $K = 2$



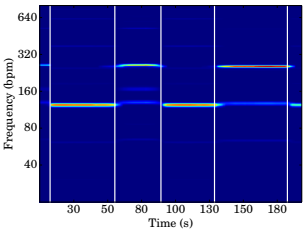
(F) **WH** with $K = 2$



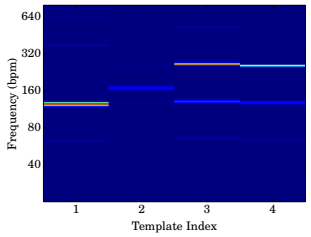
(G) **W** with $K = 3$



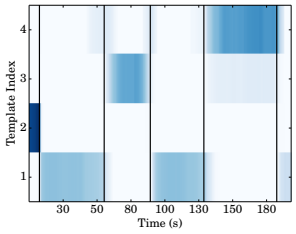
(H) **H** with $K = 3$



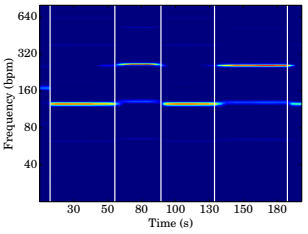
(I) **WH** with $K = 3$



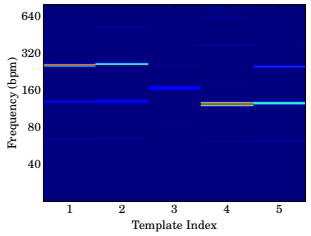
(J) **W** with $K = 4$



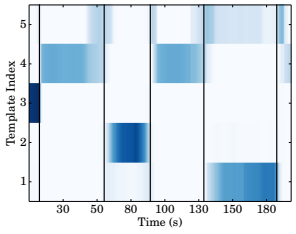
(K) **H** with $K = 4$



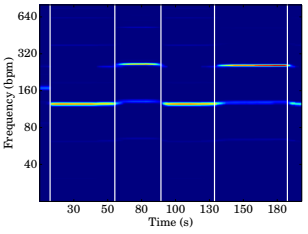
(L) **WH** with $K = 4$



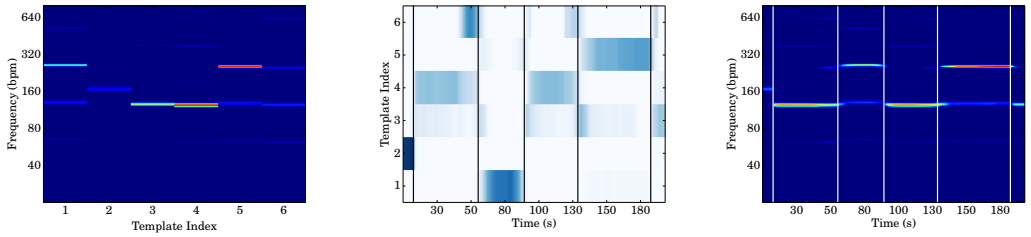
(M) **W** with $K = 5$



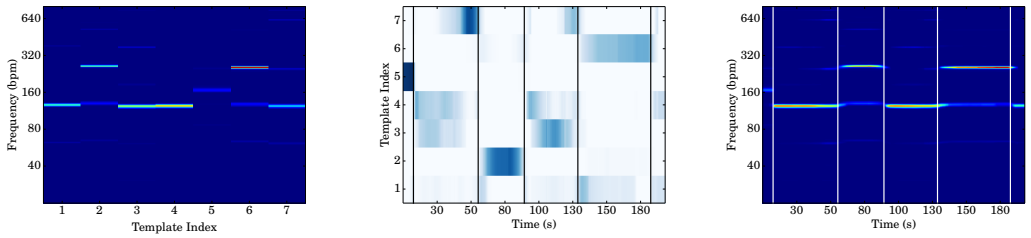
(N) **H** with $K = 5$



(O) **WH** with $K = 5$



(P) **W** with $K = 6$ (Q) **H** with $K = 6$ (R) **WH** with $K = 6$



(S) **W** with $K = 7$ (T) **H** with $K = 7$ (U) **WH** with $K = 7$

FIGURE 7.6: **NMF decompositions of of the track 'Geno' for a range of number of templates.** Each row presents from left to right the template **W**, activations **H** and reconstructed $\mathbf{W} \cdot \mathbf{H}$ matrices for a music piece containing metric modulations.

For $K < 4$ it appears that the reconstruction of **R** is not accurate. In particular, the metergram frames over the first few seconds are erroneously reconstructed: the metergram exhibit a strong energy activation around 160BPM which is not reconstructed properly. Figure 7.7 shows the NMF reconstruction error, computed as the Kullback-Leibler divergence between the matrix to be estimated, **R**, and the NMF reconstructed matrix **WH** for $K \in [1, 7]$. It exhibits a convex shape that accounts for the fact that low rank decompositions do not provide accurate reconstructions but also that further rank augmentation only result in small error decrease. This is also graphically apparent in Figure 7.6 where the reconstructed matrices are not significally different for $K \geq 4$.

In addition, we can observe that the learnt templates exhibit structures that resemble the structures of the metergram. This is in accordance with the initial aim of NMF to learn "the parts of objects" and is a benefit of the non-negativity constraint [127]. However, for the templates to capture individual parts of the metergram, it is necessary that the rank of the decomposition is large enough to learn all parts. In the present example, the templates learnt by NMF fail to represent individual parts if the rank of the decomposition is smaller than the number of parts in **R**. The activation matrix does not clearly capture the segmentation either. The learning process is constrained by the

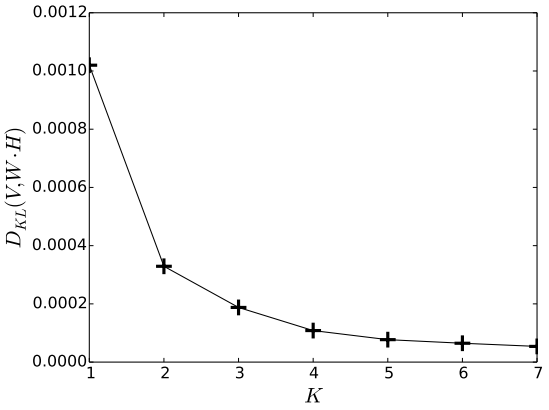


FIGURE 7.7: **NMF reconstruction error for Geno.** Reconstruction error calculated as the Kullbach-Liebler divergence between the matrix to be estimated $\mathbf{R}$ and the NMF reconstructed matrix $\mathbf{WH}$ for $K \in [1, 7]$

minimisation of $D(\mathbf{R}, \mathbf{WH})$, which is applied for the entire matrix. Therefore, a simple intuition to explain this result is that if the rank of the decomposition is too small, the templates that are learnt will jointly capture structures of several parts in order to minimise the reconstruction error. This behaviour is easily observed in Figure 7.6 for small values of K. For instance, for $K = 1$, the learnt template summarises the energy distribution in $\mathbf{R}$ along the time axis. Obviously, such a decomposition does not enable an accurate reconstruction, which results in a high reconstruction error, as can be seen in Figure 7.7.

As the rank of the decomposition increases, more templates are available to more finely reconstruct the data. The drawback of this gain in precision is the dilution of the summarising quality of the templates. As K increases, the decomposition starts producing templates that represent subspaces of parts. In other words, the information that would ideally be captured by one template is now spread over several, so that one template does not necessarily represent a metergram frame. By construction of the NMF decomposition, such a subspace decomposition implies simultaneous activation of several templates in order to produce an accurate reconstruction of $\mathbf{R}$. Each metergram frame is then a linear combination of co-activated templates. Simultaneous activations can for instance be observed in Figure 7.6 (T). Templates 3 and 4 tend to mostly be co-activated. They may therefore be seen as a subspace decomposition of the beat spectrum of the second and fourth segments of the metergram. The gain in precision also manifests itself in

the transition zones between two parts. For $K = 5$ and over, we observe an increasing number of very short activations in matrix $\mathbf{H}$ hence corresponding to transitional states. Some of the learnt templates are strictly dedicated to the description of the transitions e.g. template 7 in Figure 7.6 (T). In these circumstances the segmentation does not appear very clearly in the activation matrix. In conclusion, it is clear that the rank of the NMF decomposition has a large impact on the properties of the learnt representation of the data. When chosen too small, an accurate reconstruction of the initial data (matrix $\mathbf{R}$) is not possible. Conversely, when chosen too large, the reconstruction is very accurate but the learnt representation does not represent the segmentation in a straightforward fashion. The choice of the optimal rank of the decomposition therefore appears to be critical to retrieve accurate segmentation by frame clustering. We note that a similar observation was made in a different context in [172]. In that case, the authors also approached homogeneity-based segmentation as a clustering task, though their problem formulation as well as the task they addressed were different. They employed a shift-invariant Probabilistic Latent Component Analysis (SI-PLCA) in which the templates were 2-dimensional matrices representing chord sequences and also noted that an accurate rank estimation is instrumental in achieving good segmentation performance.

We are interested here in recovering the segmentation of a music piece into segments with consistent metrical structure. As a result, recovering information about the transitions between segments, which may be a valuable information in itself, is of little interest in this context. On the other hand, recovering all the parts is essential. Obtaining a decomposition in which templates correspond to parts of the metergram and activations correspond to the temporal extent of the parts represents our desired outcome. It appears from the discussion above that choosing the rank equal to (or in the close vicinity of) the number of different parts in the data seems ideal. Assuming the number of different parts in a music piece is not known a priori, in the following we present a variety of methods to automatically determine the optimal rank or circumvent the rank determination problem by applying sparsity constraints to the NMF approximation.

7.4.2 Heuristic automatic rank determination baseline

When the chosen rank is too small, the factorisation cannot be accurate (cf. Figure 7.6), implying a large reconstruction error. This error is expected to decrease when the rank increases, becoming reasonably small when K is equal to the number of different segments in the track, with small decreases in error for further rank augmentation. On this premise, we devise a baseline automatic rank estimation method, notated NMF-K_e. For each track, an NMF decomposition and the reconstruction error is computed for a range of ranks, i.e. $K \in \{1, ..., 10\}$. The effective rank K_e is selected so that

$$K_e \triangleq K : D_{KL}(\mathbf{R}, \mathbf{WH})_K \geq \epsilon \text{ and } D_{KL}(\mathbf{R}, \mathbf{WH})_{K+1} < \epsilon \tag{7.7}$$

with $\epsilon = 2.10^{-4}$. The activation matrix from the factorisation of rank $K = K_e$ is then used to retrieve segmentation. We refer to Section 7.4.6 for the evaluation of the segmentation performance obtained with this method.

7.4.3 Sparse-NMF

Owing to non-negativity constraints, the data representations produced by NMF tend to be relatively sparse [127], but in its basic formulation NMF does not allow control over the degree of sparsity. Nevertheless, it has been shown to be beneficial for a variety of tasks to enforce sparsity constraints [142, 219–223] or to explicitly control the sparseness of the learnt decomposition [224]. Sparsity constraints may be applied either on the activation or templates matrix or both. Because the learnt templates are expected to represent parts of the input data, it is to be expected that the density of the templates resembles that of the input data, which is therefore not necessarily sparse in general. In our case, the input data (the metergram $\mathbf{R}$) is very sparse — only a few of its coefficients are significantly non-zero — hence leading to the learning of very sparse templates.

It has been shown in Section 7.4.1 that the rank of the NMF decomposition has a significant impact on the properties of the data representation that is learnt. When the rank is too small, the data cannot be represented well, so this situation must be avoided. When the rank is too large, the activation matrix gets fragmented, simultaneous

activations develop and templates start learning sub-spaces of the parts of the data, if not being redundant. Given that the optimal rank is not known a priori, we propose to choose it purposefully too large and apply sparsity constraints on the activation matrix so that fragmentation of activations and co-activations are penalised. We assume unlikely that a piece contains more than 10 different metrical structures and therefore set the rank $K = 10$. One may also consider estimating the maximum number of modulations per piece from the dataset. Although this approach is perfectly justifiable in general, it goes against the fully unsupervised philosophy adopted here. Note that we have chosen $K = 10$ in this study, but any arbitrary value for K would be suitable, provided that it is purposefully chosen to be too large. In a hypothetical scenario where the maximum number of modulation is not known and cannot be estimated, the larger the value chosen for K, the lower the probability that a piece featuring K distinct metrical structures is encountered. In the following we consider a range of sparsity constraints to be enforced in the NMF approximation optimisation.

7.4.3.1 NMF with L_1 activation sparsity constraint

The first method we consider is a standard KL divergence NMF with the addition of an activation penalty defined as the L_1-norm of the activation matrix:

$$\Upsilon = ||\mathbf{H}||_1 \tag{7.8}$$

where the L_1-norm is given by $||\mathbf{H}||_1 = \sum_{i,j} h_{ij}$. This constraint encourages factorisations that keep the activations in $\mathbf{H}$ to a minimum by penalising non-zero entries. Replacing equation (7.8) in equation (2.29), the cost function to minimise is:

$$D_{KL}(\mathbf{R}|\mathbf{WH}) + \alpha\,||\mathbf{H}||_1 \tag{7.9}$$

Minimising the cost function implies the minimisation of Υ alongside the reconstruction error $D_{KL}(\mathbf{R}|\mathbf{WH})$, with the tradeoff between the respective weight of the reconstruction error and the penalty being controlled by α. Then, increasing α leads to greater sparsity

and vice-versa. Multiplicative update rules can be derived to include this constraint in the optimisation of the factorisation, following [225]:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T(\frac{\mathbf{R}}{\mathbf{WH}})}{\mathbf{W}^T\mathbf{J} + \alpha} \tag{7.10}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{(\frac{\mathbf{R}}{\mathbf{WH}})\mathbf{H}^T}{\mathbf{J}\mathbf{H}^T} \tag{7.11}$$

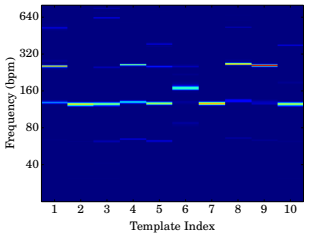
where $\mathbf{J}$ is a matrix of ones of dimension $M \times N$. Let us notate this method as SNMF-L. A trivial solution to minimise the penalty term of equation (7.9) is to scale down the activations in $\mathbf{H}$ while increasing the norm of the templates in $\mathbf{W}$ so that the product $\mathbf{WH}$ is not affected. In this case, the optimisation problem is equivalent to a standard NMF with no penalty. In order to prevent such a behaviour, a third update implements the L_1-normalisation of the columns of the templates matrix $\mathbf{W}$ at each iteration [225]:

$$\mathbf{W} \leftarrow \frac{\mathbf{W}}{\text{repmat}(\mathbf{J}_{1,M}\mathbf{W}, M, 1)} \tag{7.12}$$

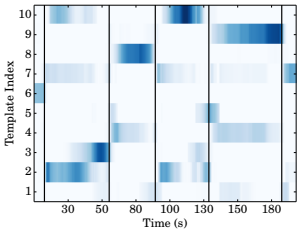
where $\mathbf{J}_{1,M}$ is a matrix of ones of size $1 \times M$ and $\text{repmat}(\mathbf{J}_{1,M}\mathbf{W}, M, 1)$ is a matrix of dimension $M \times K$ which M rows are the repetition of vector $\mathbf{J}_{1,M}\mathbf{W}$ of dimension $1 \times K$. Figure 7.8 shows the factorisations obtained with SNMF-L method, for a range of values of the sparsity penalty weight α. It is clear that equations (7.10) and (7.11) are equivalent to equations (2.24) and (2.25) — that is to say standard NMF — when $\alpha = 0$. The factorisation obtained for $\alpha = 0$ exhibits a typical NMF behaviour. Even with very large weight, the sparsity constraint does not seem to have a very significant influence on the learnt activation matrix.

7.4.3.2 Monotonic algorithm for NMF with L_1 activation sparsity constraint

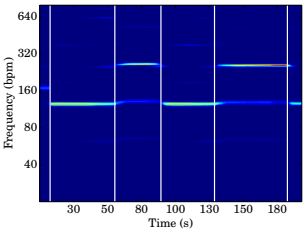
A shortcoming of the method described in Section 7.4.3.1 is that the addition of the normalisation update (7.12) to the multiplicative update rules (7.10) and (7.11), does not guarantee the decrease of the cost function at each iteration. In this section we summarise and examine a NMF algorithm with with L_1 activation sparsity constraint



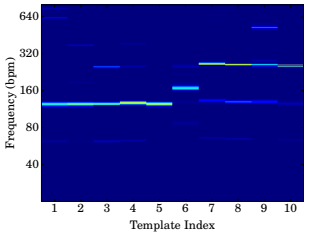
(A) **W** with $\alpha = 0$



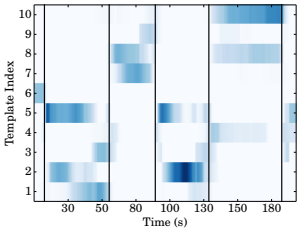
(B) **H** with $\alpha = 0$



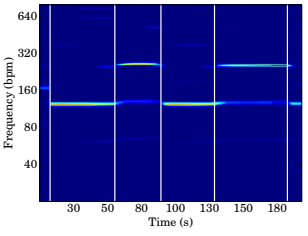
(C) **WH** with $\alpha = 0$



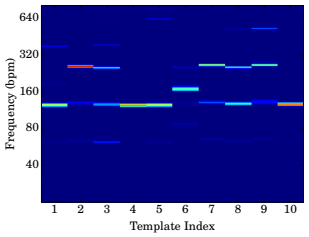
(D) **W** with $\alpha = 0.01$



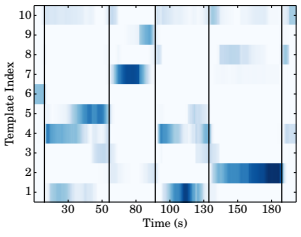
(E) **H** with $\alpha = 0.01$



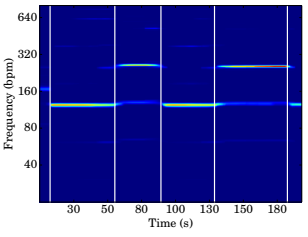
(F) **WH** with $\alpha = 0.01$



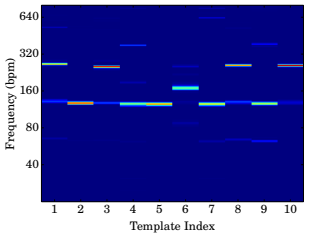
(G) **W** with $\alpha = 0.1$



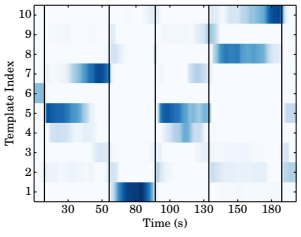
(H) **H** with $\alpha = 0.1$



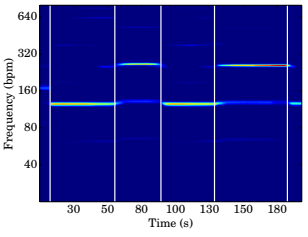
(I) **WH** with $\alpha = 0.1$



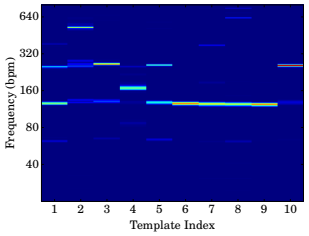
(J) **W** with $\alpha = 1$



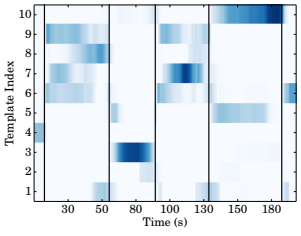
(K) **H** with $\alpha = 1$



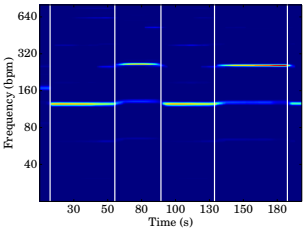
(L) **WH** with $\alpha = 1$



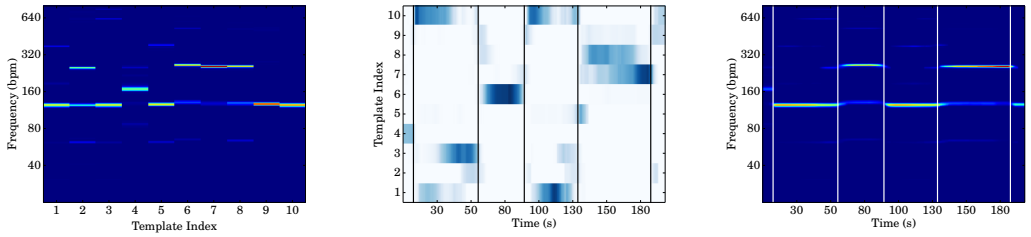
(M) **W** with $\alpha = 10$



(N) **H** with $\alpha = 10$



(O) **WH** with $\alpha = 10$



(P) **W** with $\alpha = 100$ (Q) **H** with $\alpha = 100$ (R) **WH** with $\alpha = 100$

FIGURE 7.8: **Sparse-NMF decompositions for a range of values of α.** Each row presents from left to right the template **W**, activations **H** and reconstructed **WH** matrices for a music piece containing metric modulations: "Geno (Tribute to Dexys Midnight Runners)" by Union of Sound. All decompositions are computed with $K = 10$

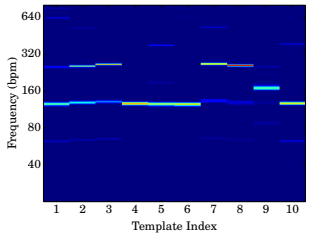
that guarantees a monotonic descent. The multiplicative update rules considered are [142, 219]:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\bar{\mathbf{W}}^T \left(\frac{\mathbf{R}}{\bar{\mathbf{W}}\mathbf{H}} \right)}{\bar{\mathbf{W}}^T \mathbf{J} + \alpha} \tag{7.13}$$

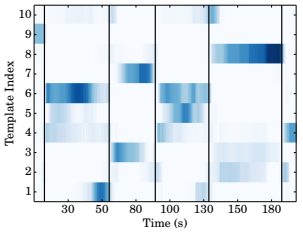
$$\mathbf{W} \leftarrow \bar{\mathbf{W}} \odot \frac{\left(\frac{\mathbf{R}}{\bar{\mathbf{W}}\mathbf{H}} \right) \mathbf{H}^T + \bar{\mathbf{W}} \odot \left(\mathbf{1} \left(\mathbf{J}\mathbf{H}^T \odot \bar{\mathbf{W}} \right) \right)}{\mathbf{J}\mathbf{H}^T + \bar{\mathbf{W}} \odot \left(\mathbf{1} \left(\frac{\mathbf{R}}{\bar{\mathbf{W}}\mathbf{H}} \mathbf{H}^T \odot \bar{\mathbf{W}} \right) \right)} \tag{7.14}$$

where $\bar{\mathbf{W}}$ is the column-normalised templates matrix, $\mathbf{J} \in \mathbb{R}_{\geq 0}^{M \times N}$ is a matrix of ones of the same size as **R** and $\mathbf{1} \in \mathbb{R}_{\geq 0}^{M \times M}$ is a square matrix of ones. Let us label this method SNMF-S.

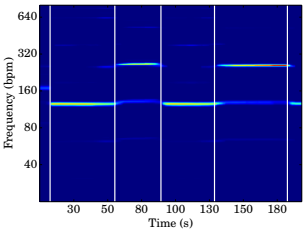
Figure 7.9 shows the NMF decompositions computed for a range of values of the sparsity parameter α. The sparsity of matrix **H** appears to be noticeably affected for values around $\alpha \approx 10$ and above. It is also to be noted that when sparsity parameter values get large, i.e. $\alpha \gg 1$, the resemblance between the learnt templates and the original data is affected. As a result, the fidelity of the reconstructed matrix **WH** is also adversely impacted. This effect is very prominent in the case $\alpha = 100$ depicted in Figure 7.9 (P) (Q) and (R). On the other hand, the activation matrix **H** exhibits a structure that is close to the structural segmentation of the piece. Moreover, the role of the learnt components is semantically much more meaningful than for lower sparsity value. For instance, component 8 clearly corresponds to the opening part of the track that is never repeated. Component 3 corresponds to the 2$^{\text{nd}}$, 4$^{\text{th}}$ and 6$^{\text{th}}$ part, which have the same



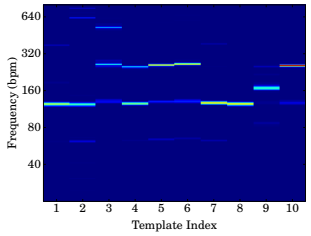
(A) **W** with $\alpha = 0$



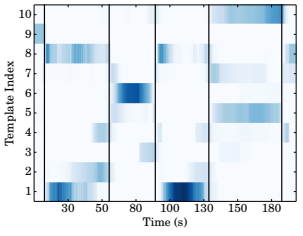
(B) **H** with $\alpha = 0$



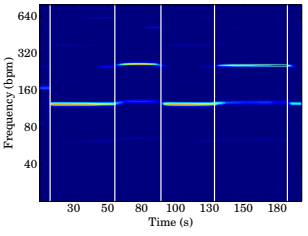
(C) **WH** with $\alpha = 0$



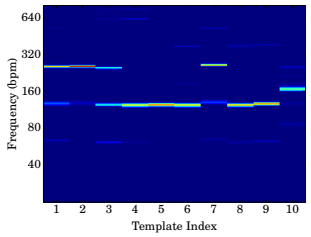
(D) **W** with $\alpha = 0.01$



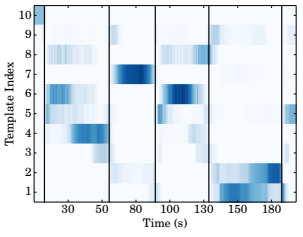
(E) **H** with $\alpha = 0.01$



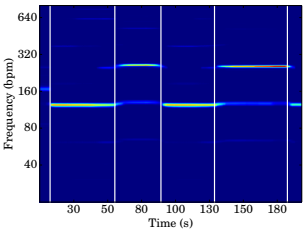
(F) **WH** with $\alpha = 0.01$



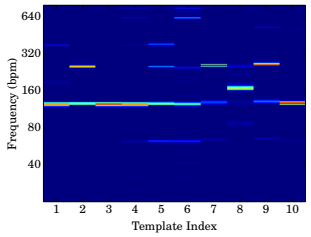
(G) **W** with $\alpha = 0.1$



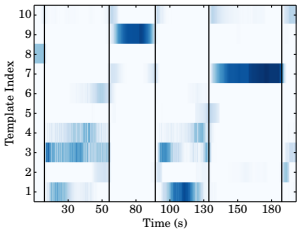
(H) **H** with $\alpha = 0.1$



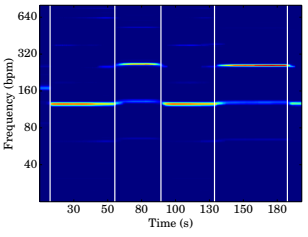
(I) **WH** with $\alpha = 0.1$



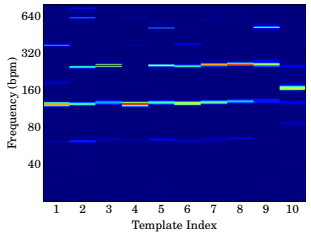
(J) **W** with $\alpha = 1$



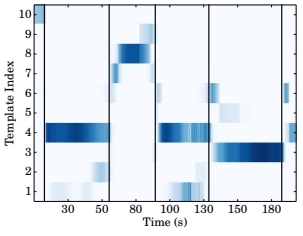
(K) **H** with $\alpha = 1$



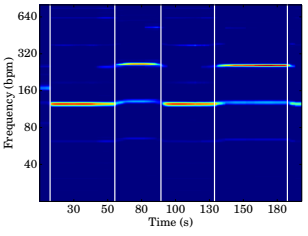
(L) **WH** with $\alpha = 1$



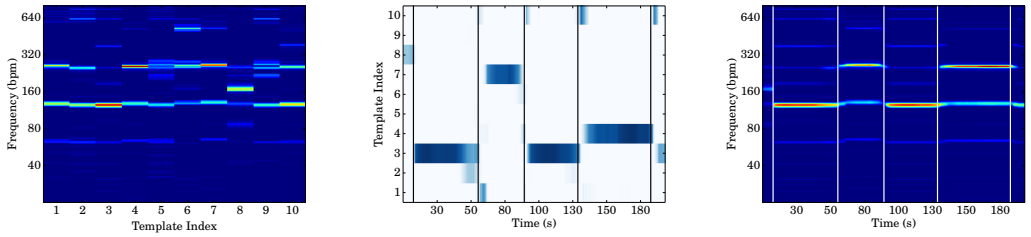
(M) **W** with $\alpha = 10$



(N) **H** with $\alpha = 10$



(O) **WH** with $\alpha = 10$



(P) **W** with $\alpha = 100$ (Q) **H** with $\alpha = 100$ (R) **WH** with $\alpha = 100$

FIGURE 7.9: **Sparse-NMF decompositions for a range of values of α.** Each row presents from left to right the template **W**, activations **H** and reconstructed **WH** matrices for a music piece containing metric modulations: "Geno (Tribute to Dexys Midnight Runners)" by Union of Sound. All decompositions are computed with $K = 10$

metrical structure, and are therefore repeated parts from a metrical structure standpoint. Components 1, 6 and 10 exclusively feature very short activations located at the border between sections, which clearly suggest that they correspond to transitional components. Interestingly, a template is learnt for components 5 and 9, but they are pruned out of the model by never being activated. In conclusion, it appears that the effect of the sparsity constraint is only significant when very large weight is given to the sparsity constraint, i.e. $\alpha \gg 10$, and produces semantically meaningful structures in the activation matrices for values of the order of $\alpha \approx 100$. This comes at the expense of inaccurate template learning and therefore inaccurate reconstruction, however.

7.4.3.3 Sparse β-NMF with L_β penalty

The observations made in Section 7.4.3.2 suggest that the structure of the activation matrix is significantly affected only when very strong sparsity constraints are applied — realised by very large values for the penalty weight α. In our scenario, greater sparsity appears to be desirable. An alternative approach to setting very large penalty weight could be choosing a stronger penalty to enforce the sparsity constraint. Defining the penalty as the L_0-norm of **H** so that $\Upsilon = ||\mathbf{H}||_0$ provides a much stronger sparsity constraint [226] but leads to a cost function with a large number of local minima and is therefore difficult to meaningfully minimise [219].

In this section we propose a method using a L_β-norm penalty to enforce the sparse activation constraint in a β-divergence NMF algorithm, inspired from the group sparsity

method developed in [144]. We refer the reader to Appendix C for the details of the mathematical derivation but summarise the key elements of this approach in the following. The reconstruction error is measured by the β divergence given in (2.26) the penalty term enforcing the sparsity constraint is:

$$\Upsilon = \frac{1}{\beta}\sum_{n=1}^{N}\|\mathbf{y}_n\|_\beta^\beta \tag{7.15}$$

where

$$y_{k,n} = h_{k,n} \times \|\mathbf{w}_k\|_2 \tag{7.16}$$

Given that the scale relationship

$$\frac{D_\beta(\mathbf{r}|\mathbf{v})}{\|\mathbf{h}\|_\beta^\beta} = \frac{D_\beta(g\mathbf{r}|g\mathbf{v})}{\|g\mathbf{h}\|_\beta^\beta} \tag{7.17}$$

is verified for $\beta > 0$ and $g \in \mathbb{R}^*$ [144], a L_β^β-norm penalty is scale-invariant to the β-divergence. In other words, choosing to pair a L_β^β-norm penalty with a β-divergence reconstruction error implies a consistent penalisation regardless of the scale of the matrices coefficients. Note that when setting $\beta = 1$ we see that the L_1-norm penalty is scale invariant to the KL divergence (cf. sections 7.4.3.1 and 7.4.3.2).

Another interesting property of this penalty term is that it ties together the L_2-norm of the metergram templates $\mathbf{w}_k$ and its activation at n^{th} time frame $h_{k,n}$ in (7.16). By trying to minimise the penalty, the algorithm therefore tries to jointly minimise the norm of the templates and their activations. As we will show later, this property is instrumental in achieving good results because it allows to prune templates out of the model by turning their norm and activation to zero. By virtue of the multiplicative updates, a template whose L_2-norm becomes zero (i.e. all its coefficients are zero) at a given iteration can no longer take non-zero values, which effectively excludes it from future updates.

The intent here is to derive a monotonic algorithm to enforce sparsity constraints significantly stronger than L_1 constraints. In order to do so, β must be chosen so that $0 < \beta < 1$. In our experiments we set $\beta = \frac{1}{2}$. Subsituting β and equation (7.15) in

(2.29), the cost function function to minimise in order to optimise the factorisation is then:

$$D_{\frac{1}{2}}(\mathbf{R}|\mathbf{WH}) + 2\alpha \sum_{n=1}^{N} \|\mathbf{y}_n\|_{\frac{1}{2}}^{\frac{1}{2}} \tag{7.18}$$

Multiplicative updates to monotonically minimise the cost function are obtained by substituting

$$\Psi_{\mathbf{W}} = \mathbf{W} \odot \text{repmat}\left(\sum_j \sqrt{h_{i,j}^T}, M, 1\right), \tag{7.19}$$

$$\Psi_{\mathbf{H}} = 1/\sqrt{\mathbf{H}} \tag{7.20}$$

and

$$\varphi(\beta) = (3-\beta)^{-1} \tag{7.21}$$

into (2.30) and (2.31). The activation and template matrices are then normalised:

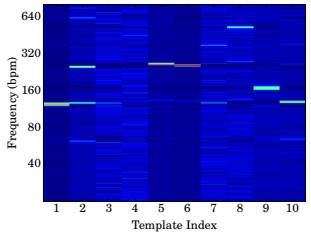
$$h_{k,n} = h_{k,n} \times \|\mathbf{w}_k\|_2 \tag{7.22}$$

$$\mathbf{w}_k = \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|_2} \tag{7.23}$$

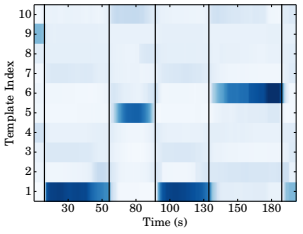
Note that given (7.16), this normalisation step does not affect the value of the cost function. Let us refer to this method as L_β-S-β-NMF in the remainder of this chapter.

Figure 7.10 shows the factorisations obtained with this method for a range of sparsity weights α. The condition $\alpha = 0$ means that the sparsity constraint is not applied to the factorisation, and therefore provides a baseline. The effect of the sparsity constraint is significantly felt for $\alpha \gtrapprox 1$. In this condition, some components are pruned out of the model: the template vectors tend to zero, so that by virtue of the multiplicative updates, once they are zero vectors they cannot be updated further. Alternatively, even if templates are non-zero vectors, they may effectively be pruned out of the model by not being activated (cf. Figure 7.10 (M), (N) and (O)). When the weight given to the sparsity penalty is too large ($\alpha \approx 10$), all components but one are pruned out of the model. The representative power of the factorisation is then lost, and the segmentation

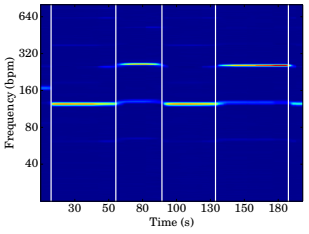
is no longer captured. The structure of **H** seems to optimally capture the segmentation for $\alpha \approx 5$. We also note that the quality of the segmentation captured in the structure of the activation matrix is significantly better with the current method than with SNMF-S (cf. Figure 7.10 (N) vs. Figure 7.9 (Q)).



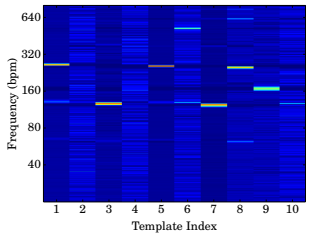
(A) **W** with $\alpha = 0$



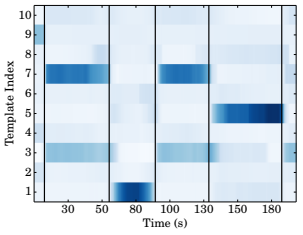
(B) **H** with $\alpha = 0$



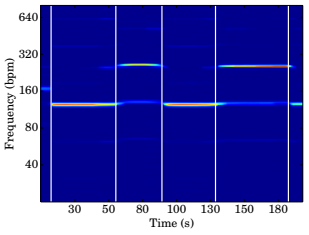
(C) **WH** with $\alpha = 0$



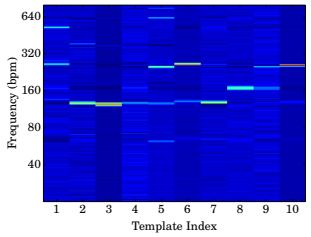
(D) **W** with $\alpha = 0.01$



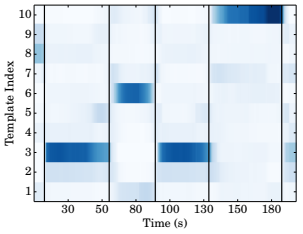
(E) **H** with $\alpha = 0.01$



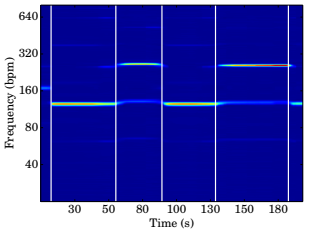
(F) **WH** with $\alpha = 0.01$



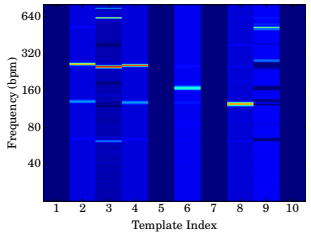
(G) **W** with $\alpha = 0.1$



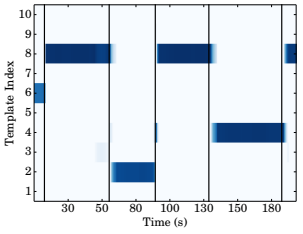
(H) **H** with $\alpha = 0.1$



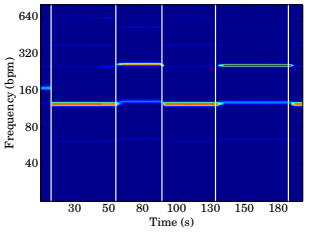
(I) **WH** with $\alpha = 0.1$



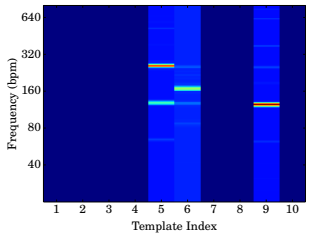
(J) **W** with $\alpha = 1$



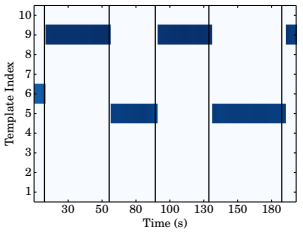
(K) **H** with $\alpha = 1$



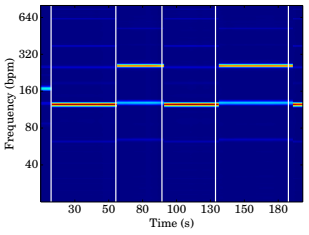
(L) **WH** with $\alpha = 1$



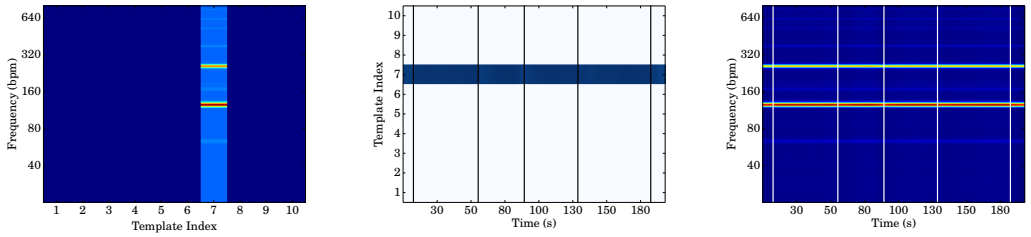
(M) **W** with $\alpha = 5$



(N) **H** with $\alpha = 5$



(O) **WH** with $\alpha = 5$



(P) **W** with $\alpha = 10$ (Q) **H** with $\alpha = 10$ (R) **WH** with $\alpha = 10$

FIGURE 7.10: **L_β-S-β-NMF decompositions for a range of values of α.** Each row presents from left to right the template **W**, activations **H** and reconstructed **WH** matrices for a music piece containing metric modulations: "Geno (Tribute to Dexys Midnight Runners)" by Union of Sound. All decompositions are computed with $\beta = \frac{1}{2}$

7.4.3.4 L_1-ARD for β-NMF

For comparison then, we present in this section a method proposed by Tan and Févotte in [227] to perform Automatic Relevance Determination (ARD) for β-NMF. It is an extension of β-NMF which effectively aims at pruning templates that only explain a little part of the observed data (matrix **R**) during the iterative optimisation. The optimal rank of the NMF decomposition can then be computed a posteriori.

A key feature of this method is that the rows of **H** are coupled with the corresponding columns of **W** via a scaling vector $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_K)$ containing the relevance weights so that the k^{th} row of the activation matrix **H** is tied with the k^{th} column of the templates matrix **W** via the coefficient λ_k. The strategy implemented by Tan and Févotte is to update the relevance weights vector $\boldsymbol{\lambda}$ as well as **H** and **W** at each iteration so that the template-activation pairs (tied by a single coefficient λ_k) of low relevance are gradually scaled down until they ultimately become zero entries. Using multiplicative update rules, the zero entries then remain zero in further iterations. Conversely, template-activations pairs of high relevance are maintained at non-zero values. As a result, starting with a rank purposefully chosen to be too large, the superfluous templates are gradually deactivated by their relevance weights tending towards zero. An effective rank estimation can then be performed by counting the number of templates that are effectively being used.

Although we refer the reader to the original publication [227] for a detailed derivation of the algorithm, we will present some of its key elements in the following. The algorithm based on multiplicative update rules is summarised in pseudo-code in Algorithm

3. The iterative algorithm takes four parameters: a, ϕ, β and τ. The parameter τ is

Algorithm 3 L_1-ARD for β-NMF

Parameters: a, ϕ, β, τ
Init: tol $= -\infty$
while tol $< \tau$ **do**
$\mathbf{H} \leftarrow \mathbf{H} \odot \left[\frac{\mathbf{W}^T\left[(\mathbf{WH})^{(\beta-2)} \odot \mathbf{R}\right]}{\mathbf{W}^T\left[(\mathbf{WH})^{(\beta-1)}\right] + \phi/\text{repmat}(\boldsymbol{\lambda}, 1, N)} \right]^{\gamma(\beta)}$
$\mathbf{W} \leftarrow \mathbf{W} \odot \left[\frac{\left[(\mathbf{WH})^{(\beta-2)} \odot \mathbf{R}\right]\mathbf{H}^T}{\left[(\mathbf{WH})^{(\beta-1)}\right]\mathbf{H}^T + \phi/\text{repmat}(\boldsymbol{\lambda}, M, 1)} \right]^{\gamma(\beta)}$
$\lambda_k \leftarrow \left(\sum_m w_{mk} + \sum_n h_{kn} + b\right)/c$ for all k
tol $\leftarrow \max_{k=1,\ldots,K} |(\lambda_k - \tilde{\lambda}_k)/\tilde{\lambda}_k|$
Calculate K_e as in Equation 7.27

the convergence stopping criterion. It is set to $\tau = 5.10^{-7}$ in our experiments[2]. Tan and Févotte report that the choice of a in a data-driven manner is not straightforward and not satisfactory. As a result, they recommend to chose it small compared to $M + N$ in order to minimise its influence. They also report that the best results were obtained under this condition. We explored values in the $a \in [5, 1000]$ range in our experiments. Here again β refers to the β-divergence. In our experiments $\beta \in \{0, 1, 2\}$.

The parameter ϕ controls the tradeoff between data fidelity and regularisation: the larger the value of ϕ, the stronger pruning of latent components. As such, ϕ has an effect on the sparsity of $\mathbf{H}$ and $\mathbf{W}$ and plays a similar role as the penalty weight α in the previous sections. In fact, we note at this point that the multiplicative update rules for $\mathbf{W}$ and $\mathbf{H}$ given in algorithm 3 closely resemble the generic form of sparse-β-NMF updates given in equations (2.30) and (2.31) where $\phi/\text{repmat}(\boldsymbol{\lambda}, 1, N)$ and $\phi/\text{repmat}(\boldsymbol{\lambda}, M, 1)$ are in lieu of $\alpha\Psi_{\mathbf{H}}$ and $\alpha\Psi_{\mathbf{W}}$ respectively.

The other terms used in Algorithm 3 are defined as:

$$c = M + N + a + 1 \tag{7.24}$$

[2]We also add a limit to the number of iterations allowed (200 in our experiments) in order to limit the computation time. As a consequence, the iterative algorithm is stopped either when tol $< \tau$ or when the maximum number of iterations is reached

$$\gamma(\beta) = \begin{cases} 1/(2-\beta) & \beta < 1 \\ 1 & 1 \leq \beta \leq 2 \\ 1/(\beta-1) & \beta > 2 \end{cases} \tag{7.25}$$

$$b = \sqrt{\frac{(a-1)(a-2)\hat{\mu}_{\mathbf{R}}}{K}} \tag{7.26}$$

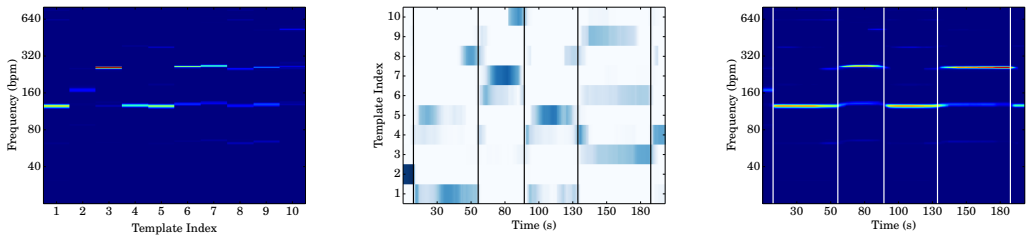
where $\hat{\mu}_{\mathbf{R}}$ is the variance of $\mathbf{R}$ and $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_K)$ and $\hat{\boldsymbol{\lambda}} = (\hat{\lambda}_1, ..., \hat{\lambda}_K)$ are the vectors of relevance weights at the current (updated) and previous iteration respectively. The notations repmat$(\boldsymbol{\lambda}, 1, N)$ and repmat$(\boldsymbol{\lambda}, M, 1)$ respectively represent matrices of $\mathbb{R}^{K \times N}$ in which each column is a $\boldsymbol{\lambda}$ vector and of $\mathbb{R}^{M \times K}$ in which each row is a $\boldsymbol{\lambda}$ vector.

Tan and Févotte propose a simple rationale to estimate the effective rank K_e of the decomposition after convergence. By construction, the λ_k are bounded so that $\lambda_k \geq \frac{b}{c}$. The bound is reached when the L_1 norms of the k^{th} column of $\mathbf{W}$ and of the k^{th} row of $\mathbf{H}$ are zero, which directly implies that the corresponding matrix coefficients are zero because of the non-negativity constraint. In other words, the relevance weights λ_k reach their lower bound when the corresponding component is pruned out of the model. The effective rank K_e is then set as the number of relevant components according to:

$$K_e = \left| \left\{ k \in \{1, ..., K\} : \frac{\lambda_k - \frac{b}{c}}{\frac{b}{c}} > \tau \right\} \right| \tag{7.27}$$

Analytically, the relevance condition is satisfied if $\frac{\lambda_k - \frac{b}{c}}{\frac{b}{c}} > 0$ but setting a small positive threshold guarantees numerical robustness. For convenience, this threshold is set to be the same as the iteration stopping criterion τ [227].

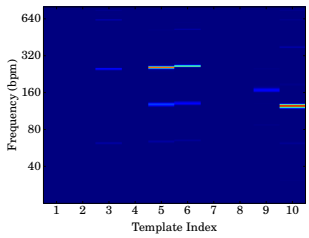
Figure 7.11 shows the factorisations computed with the L_1-ARD β-NMF algorithm for $\phi \in \{0, 0.01, 0.1, 1\}$ for the track "Geno (Tribute to Dexys Midnight Runners)" by Union of Sound. The initial rank of the decomposition is fixed to $K = 10$, $\beta = 1$ and $a = 500$. When $\phi = 0$, the regularisation term at the denominator of the update rules disappears so that they become equivalent to a standard β-NMF decomposition (equations (2.27) and (2.28)). In this case, the behaviour observed is naturally that of a standard NMF: the learnt decomposition features good fidelity of the reconstructed matrix, templates



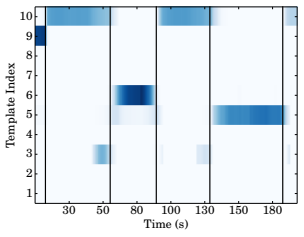
(A) **W** with $\phi = 0$

(B) **H** with $\phi = 0$

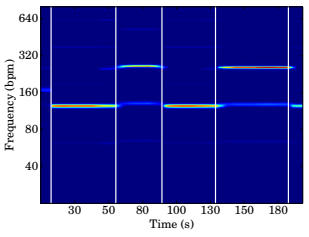
(C) **WH** with $\phi = 0$



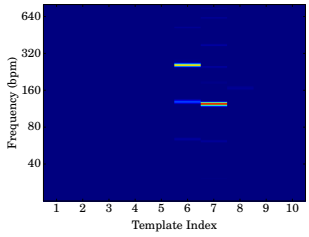
(D) **W** with $\phi = 0.01$



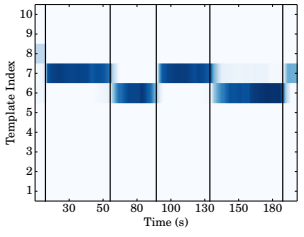
(E) **H** with $\phi = 0.01$



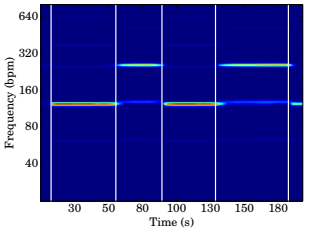
(F) **WH** with $\phi = 0.01$



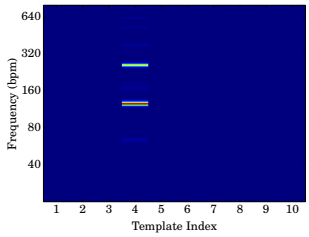
(G) **W** with $\phi = 0.1$



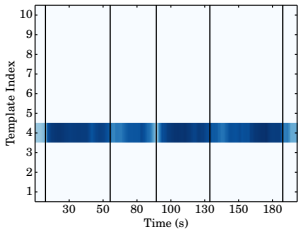
(H) **H** with $\phi = 0.1$



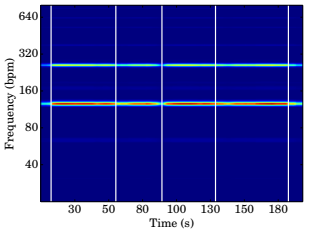
(I) **WH** with $\phi = 0.1$



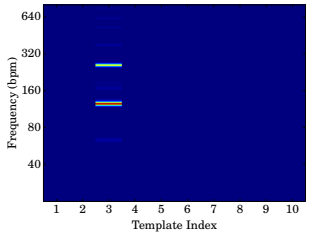
(J) **W** with $\phi = 1$



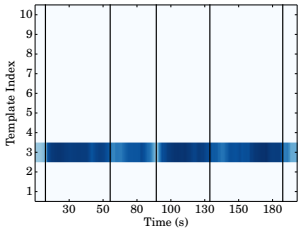
(K) **H** with $\phi = 1$



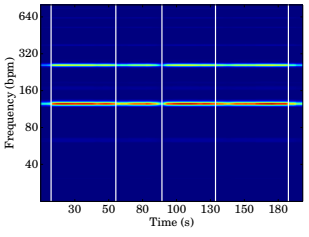
(L) **WH** with $\phi = 1$



(M) **W** with $\phi = 10$



(N) **H** with $\phi = 10$



(O) **WH** with $\phi = 10$

FIGURE 7.11: **L_1-ARD β-NMF decompositions for a range of values of ϕ.** Each row presents from left to right the template **W**, activations **H** and reconstructed **WH** matrices for a music piece containing metric modulations: "Geno (Tribute to Dexys Midnight Runners)" by Union of Sound. All decompositions are computed with $\beta = 1$, $a = 500$ and $K = 10$

learning sub-spaces of parts of the matrix $\mathbf{R}$ and a very scattered activation matrix $\mathbf{H}$ (cf. Section 7.4.1). The effect of the pruning of irrelevant components is visible when ϕ increases. As expected, the higher ϕ, the more aggressive the pruning. When ϕ gets too large, the pruning effect is so pronounced that only one component remains, learning a template that is a form of mean of $\mathbf{R}$ over time. The tradeoff between pruning and fidelity is also visible in Figure 7.11: the greater the pruning (and therefore the sparsity), the lower the fidelity of the reconstruction $\mathbf{WH}$. More interestingly, this method shows very promising behaviour for $\phi = 0.1$. Although the fidelity of the reconstruction is not perfect, the segmentation of the piece is very well captured in $\mathbf{H}$.

It is interesting to note at this point that the factorisations, and in particular the structure of the activation matrices, learnt with ARD-β-NMF are similar to those learnt with L_β-S-β-NMF. Tan and Févotte note in [228] that ARD-β-NMF bears some similarities to re-weighted L_1-minimisation [229], which is a way to enforce sparsity constraints stronger than the standard L_1 penalty. L_β-S-β-NMF with $\beta < 1$ also aims at enforcing sparsity constraints stronger than L_1 but does so by employing an alternative penalisation (L_β-norm). In addition, we note that the term $\mathbf{y}_n$ in the penalty used in L_β-S-β-NMF (equation (7.15)) ties together the norm of a template with its activation at a given time frame, and therefore constitutes another point of similarity with ARD-β-NMF, which ties together templates vectors and rows of $\mathbf{H}$ with the relevance parameter $\boldsymbol{\lambda}$.

Nevertheless, we note a significant difference between the two methods. ARD-β-NMF relies on prior distributions parametrised by hyper-paremeters a, b and c. Tan and Févotte outline analytical relationships between these hyper-parameters so that b and c are determined by the value of a. In other word, the only parameter to be fixed is a. However, it remains unclear how to systematically determine a. L_β-S-β-NMF, on the other hand, does not require this extra parameter and therefore greatly simplifies the model selection.

7.4.4 Comparison with K-means

As opposed to the novelty detection based techniques presented in Section 7.3, the NMF-based approach to automatic segmentation retrieval exposed in Section 7.4 is related to

a clustering problem. The goal is to learn latent components that represent parts of the input data (matrix **R**). Then, each frame of the input data is approximated by a weighted combination of learnt templates. These weights are more commonly referred to as activations in the NMF framework. Then, the activation matrix **H** may be interpreted as a cluster assignment matrix. In this section we propose to use the popular K-means clustering method for comparison with NMF-based methods. We refer to Section 2.7 for a brief description of the K-means clustering method and the algorithm used to implement it, as well as the definition of some notation.

Here again, we consider the metergram as the observation matrix. The K-means clustering procedure assigns each column of **R** to one and only one cluster. The assignments are stored in a vector $C = (c_1, ..., c_N)$, each element representing the index of the corresponding cluster, i.e. $c_i = k$ with $i \in [\![1, N]\!]$ and $k \in [\![1, K]\!]$. The result of the K-means clustering can be represented in a matrix decomposition similar to NMF. The template matrix **W** is straightforwardly defined as the matrix whose columns are the cluster centroids. The activation matrix **H** is derived from the cluster assignments so that each entry is:

$$h_{ij} = \delta_{ic_j} \tag{7.28}$$

where δ is the Kronecker symbol. Then, an approximative reconstruction of **R** can be computed as **WH**.

K-means is a "hard" clustering method in that it assigns every observation to one and only one cluster. A direct consequence of this property is that the activation matrix **H** obtained by the method described above from the K-means decomposition is binary. In this respect, it differs from the NMF framework that may be seen as a "soft" clustering method in which the cluster assignments (the activations) take continuous values and template co-activations are possible.

Figure 7.12 shows a series of NMF-like matrix decomposition and reconstructions obtained by K-means clustering for a range of K values. A behaviour comparable to NMF is observed in the activation matrix. As K increases, the long parts are learnt first and templates for transitional components start being learnt (for $K \geq 4$ in the example).

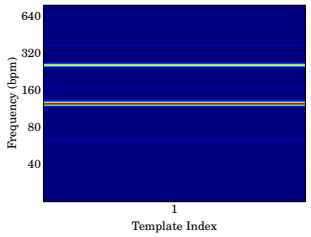
Similar conclusions as in the case of NMF developed in Section 7.4.1 can be drawn in the case of K-means based decompositions with the exception that the hard clustering property prevents the learning of cluster centroids that form sub-spaces of the ideal parts decomposition of $\mathbf{R}$. K-means owes its popularity more to its simplicity and the efficiency of the algorithms that implement it than to its accuracy, which is inferior to many other clustering techniques. This is visible in the reconstruction matrices $\mathbf{WH}$ that tend to be coarser than with NMF. However, K-means also produces activation matrices with structure that meaningfully relates to the segmentation of a piece. In this respect, K-means clustering may be a competitive alternative to NMF.

As with NMF though, the rank estimation problem persists with K-means as K must still be set in advance. In our experiments, we apply the heuristic automatic rank estimation method presented in Section 7.4.2 in order to provide a K-means-based baseline to compare other algorithms. We refer to this method as K-means-K_e.

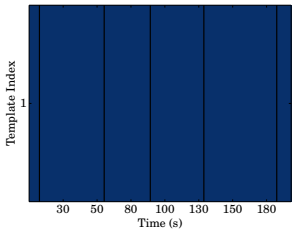
7.4.5 Hidden Markov Model for final segmentation

We aim at retrieving the segmentation from the activation matrix $\mathbf{H}$, regardless of the technique used to compute it (NMF or k-means). It has be shown above that its arrangement closely relates to the annotated reference segmentation when the number of latent components is optimally chosen. When the number of components is too large and regularisation is not optimally applied, transitional and, only if using NMF, simultaneous activations arise. However, the structure of a piece is modelled here as a series of non-overlapping segments and we do not seek to capture transitional components. In order to retrieve segmentation with a structure that matches these requirements, we propose the use of a simple Hidden Markov Model (HMM) to make the final segmentation decision. We refer the reader to Section 2.9 for a brief presentation of HMM and the definition of the corresponding notation.

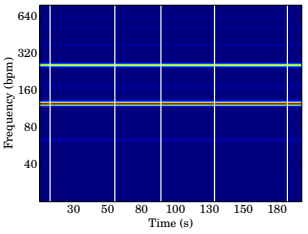
We assume that the structure of the activation matrix $\mathbf{H}$ already largely correlates to the segmentation, i.e. segments are highly correlated to contiguous series of large activation coefficients. We also assume that there may be transitional and simultaneous activations



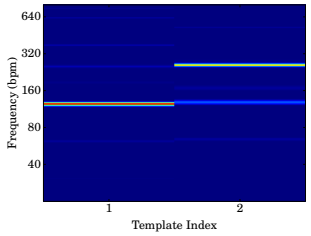
(A) **W** with $K = 1$



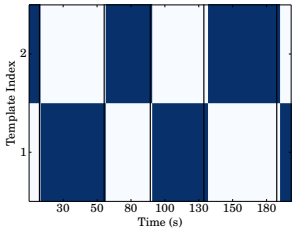
(B) **H** with $K = 1$



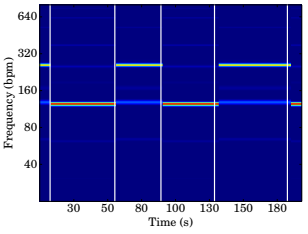
(C) **WH** with $K = 1$



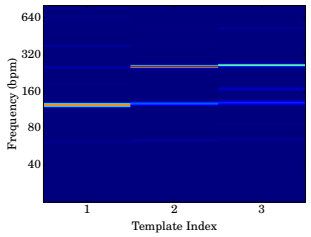
(D) **W** with $K = 2$



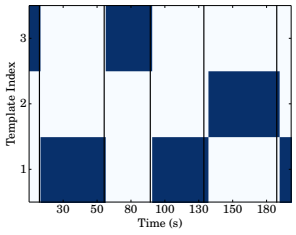
(E) **H** with $K = 2$



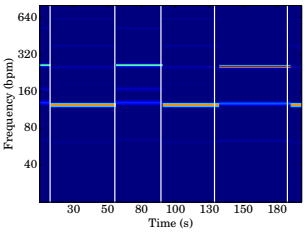
(F) **WH** with $K = 2$



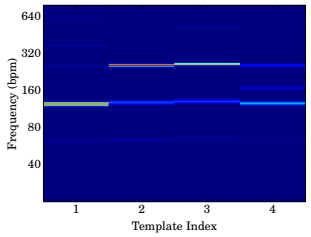
(G) **W** with $K = 3$



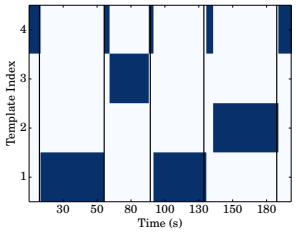
(H) **H** with $K = 3$



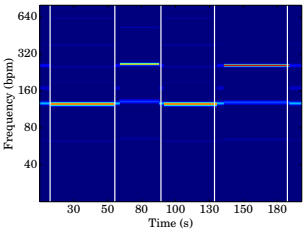
(I) **WH** with $K = 3$



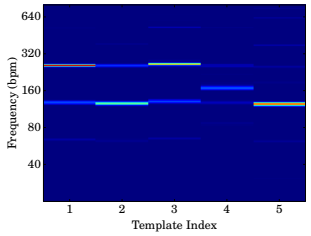
(J) **W** with $K = 4$



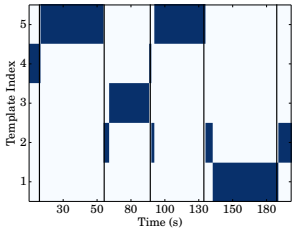
(K) **H** with $K = 4$



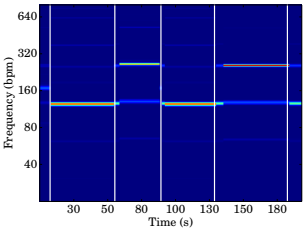
(L) **WH** with $K = 4$



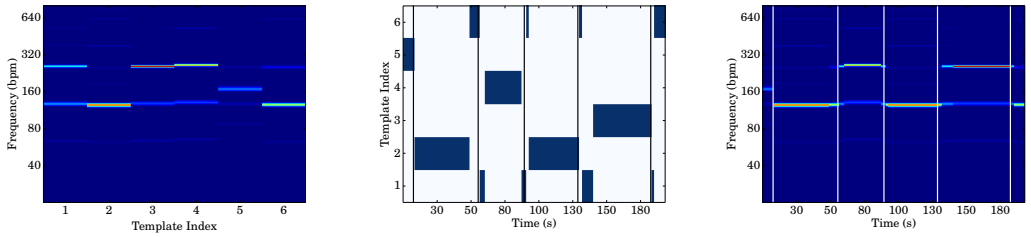
(M) **W** with $K = 5$



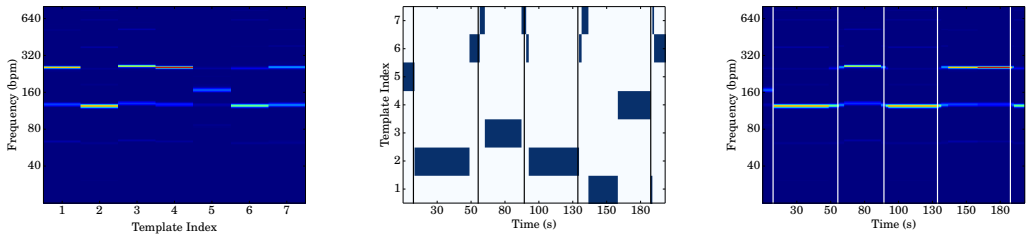
(N) **H** with $K = 5$



(O) **WH** with $K = 5$



(P) **W** with $K = 6$ (Q) **H** with $K = 6$ (R) **WH** with $K = 6$



(S) **W** with $K = 7$ (T) **H** with $K = 7$ (U) **WH** with $K = 7$

FIGURE 7.12: **K-means decompositions of and reconstructions for a range of number of clusters.** Each row presents from left to right the cluster centroids **W**, activations **H** and reconstructed **WH** matrices for a music piece containing metric modulations: "Geno (Tribute to Dexys Midnight Runners)" by Union of Sound.

(cf. Figure 7.6(Q) for example). The HMM then produces an estimate of the most likely non-overlapping segmentation that could have generated the observed data (matrix **H**).

The number of hidden states is set equal to the rank K, each state being associated with the *true* activation of a component — i.e. of the activation of the metrical structure specific to a segment. We define the emission probability $\pi(e_k|\psi_k)$, i.e. the probability of emitting the component index k from the k^{th} hidden state ψ_k as:

$$\pi(e_k|\psi_k) = \frac{\exp\left(-A_{k,n}\right)}{\sum_{k=1}^{K}\exp\left(-A_{k,n}\right)} \tag{7.29}$$

with

$$A_{k,n} = \frac{(h_{k,n} - \mu)^2}{2\sigma^2} \tag{7.30}$$

where $h_{k,n}$ is the activation coefficient of the k^{th} component at the n^{th} time frame, $\mu = \max_{k,n}(h_{k,n})$, and $\sigma = \mu$. The emission probability $\pi(e_k|\psi_k)$ is therefore large for large activation coefficients and vice versa.

The transition probabilities are defined in two classes: the probability of remaining in the same state $P(\psi_i|\psi_i)$ and the probability of a transition from state ψ_i to state ψ_j notated $P(\psi_j|\psi_i)$. Since we consider a scenario in which no prior knowledge regarding the metrical structure nor the metric modulations is assumed, no particular transitions are favoured. As a consequence the transition probabilities are uniform across the state space for $i \neq j$:

$$\forall(i,j) \in \{1,...,K\}, \begin{cases} P(\psi_j|\psi_i) = P_d & i = j \\ P(\psi_j|\psi_i) = \frac{1-P_d}{K-1} & i \neq j \end{cases} \tag{7.31}$$

where P_d is the probability of remaining in the same state and K is the number of states, set to equal the rank of the NMF decomposition. The probability P_d effectively controls the inertia of the model: the higher P_d, the higher the likelihood to stay in the same state and vice versa. The HMM is employed here to filter out transitional components and co-activations. In order to do so, it should therefore penalise numerous transitions between states. For this reason, a model high inertia (i.e. large P_d) is desirable. We set $P_d = 0.9$ in our experiment, as it has been found to produce the best results in our preliminary experiments.

The HMM is decoded using the Viterbi algorithm to reveal the most likely state sequence $\psi = \{\psi_{k1}, \cdots, \psi_{kN}\}$, where $\psi_{k,n}$ denotes the HMM state ψ_k at the n^{th} time frame. Segment boundary timeframes $\hat{n}$ are easily extracted from the HMM state sequence by detecting state transitions:

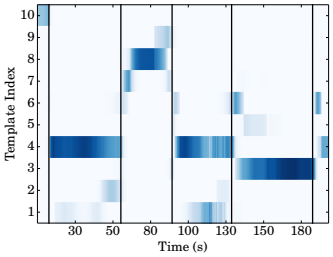
$$\hat{n} \triangleq n : \psi_{k,(n-1)} \neq \psi_{k,n} \tag{7.32}$$

For ease of visual comparison, an activation matrix corresponding to the HMM state sequence $\mathbf{H}_{\text{HMM}} \in \mathbb{R}_{\geq 0}^{K \times N}$ is formed in a similar fashion as with K-means, i.e. defining each entry as:

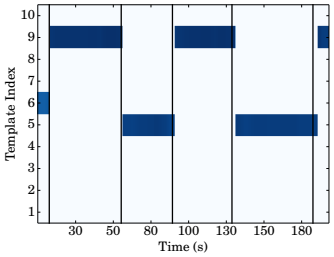
$$h_{k,n} = \delta_{k,\psi_{k,n}} \tag{7.33}$$

where δ is the Kronecker symbol.

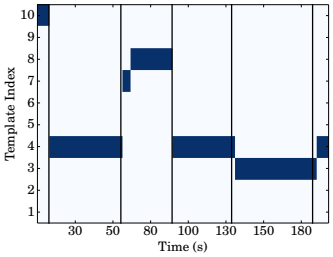
Figure 7.13 shows example of such matrices $\mathbf{H}_{\text{HMM}}$ as well as the activation matrices $\mathbf{H}$ they are derived from for different NMF-based methods. It suggests that the improvement brought by the use of the HMM depends on the properties of the activation matrix



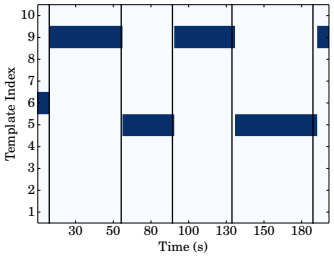
(A) **H** for SNMF-S with $\alpha = 10$



(B) **H** for L_β-S-β-NMF with $\alpha = 5$



(C) $\mathbf{H}_{\text{HMM}}$ for SNMF-S with $\alpha = 10$



(D) $\mathbf{H}_{\text{HMM}}$ for L_β-S-β-NMF with $\alpha = 5$

FIGURE 7.13: **HMM-based segmentation inference.** (A) and (C) show the activation matrix **H** and the HMM segmentation estimate in the form of an activation matrix $\mathbf{H}_{\text{HMM}}$ for SNMF-S with $\alpha = 10$. Similarly, (B) and (D) are matrices **H** and $\mathbf{H}_{\text{HMM}}$ for L_β-S-β-NMF with $\alpha = 5$. All matrices are processed from the track "Geno (Tribute to Dexys Midnight Runners)" by Union of Sound.

H. In a case where **H** contains transitional activations and co-activations, the HMM seems to improve the segmentation. On the other hand, Figure 7.13 (B) and (D) suggest that when the segmentation is already accurately captured by **H**, the HMM cannot bring any further improvement, though it is important to note that it does not have a deleterious impact on the segmentation. Finally, it suggests that despite the use of the HMM, the quality of the segmentation captured by **H** has a direct impact on the final segmentation result (cf. Figure 7.13 (B), which features a spurious short segment around the 55s mark). This suggest that even if the HMM can improve an untidy segmentation, the use of a technique that produces activation matrices accurately capturing the segmentation is still preferable.

7.4.6 Results and Discussion

We present here an evaluation of the performance of the various segmentation strategies described above, using the metrics presented in Section 2.10.3 on the metric modulations

dataset presented in Section 3.6.

7.4.6.1 Effect of the rank

It has been discussed on an example in Section 7.4.1 how critical the impact of the choice of the rank of the NMF (or K-means) decomposition is on the properties of the representation learnt. Figure 7.14 presents the influence of the rank on the average performance of NMF and K-means decompositions over the entire dataset with respect to a selection of metrics. When $K = 1$, the model cannot capture more than one metrical structure and is therefore not able to capture changes (i.e. no segment boundary detected), hence the F-measure equalling zero. The absence of segment boundaries implies that the track is regarded as made of a unique segment, which is a case of extremee under-segmentation, hence the very low S_u though it is non-zero as there is at least one annotated segment overlapping with a part of the extracted segment. The *pfm* is non-zero for the same reason. When only one estimated segment, that is $N_e = 1$, it implies $\log_2 N_e = 0$. As a consequence, Equation 2.58 is not defined and the over-segmentation score S_o cannot be calculated. It is forced to zero in this case.

As the rank of the decomposition increases, so that $K > 1$, S_u quickly saturates for $K \gtrapprox 3$, i.e. the average under-segmentation is very small under this condition and further rank increase do not bring significant performance improvement. This suggests that the minimum rank required to achieve good average performance on this dataset lies in the vicinity of $K \approx 3$. Considering the three other metrics, it appears that two-templates models ($K = 2$) lead to the best average results for both NMF and K-means. The performance decreases when the order of the model increases, in a particularly dramatic manner in the case of *pfm* metric. The fall in performance can be explained by the increasing over-segmentation, reflected by a decrease in S_o, and related to the scattering of the activation matrices easily apparent in Figure 7.6 and Figure 7.12. This result suggests that rank $K = 2$ optimally suits this dataset. Note that it is not to be expected that this optimal rank holds in general. Nevertheless, this result also demonstrates that the choice of the rank has a significant impact on the segmentation performance, and it

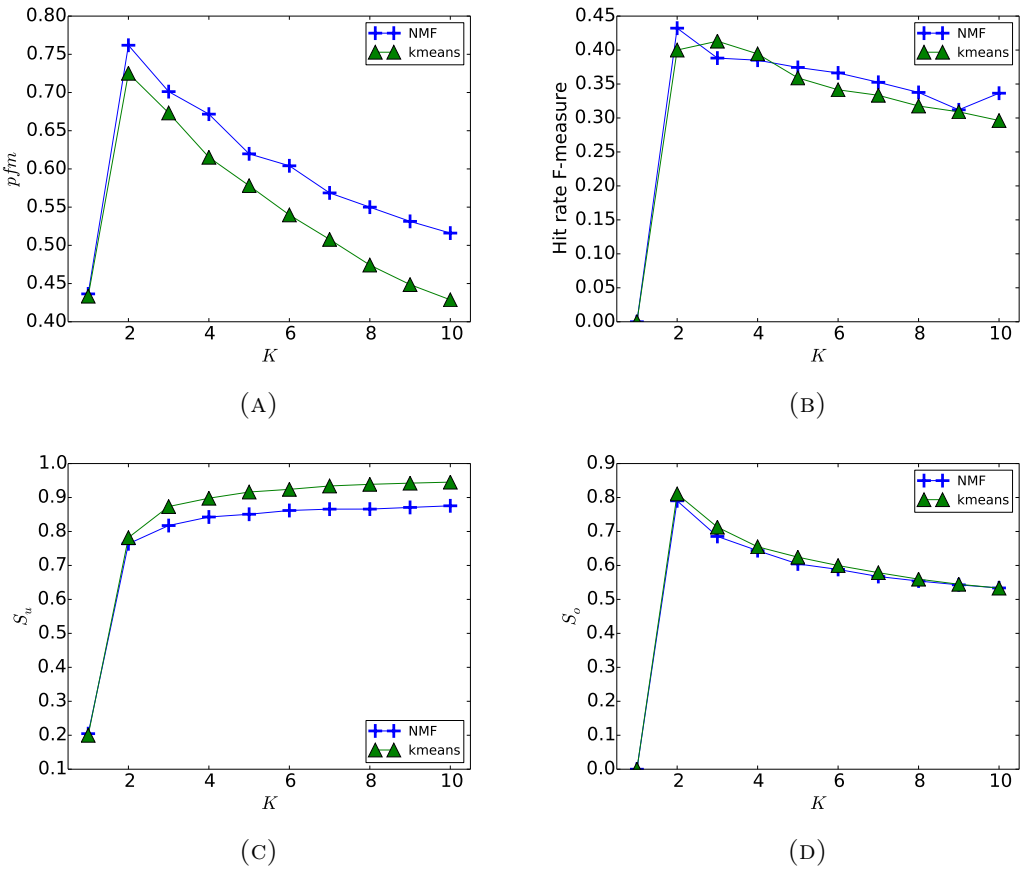


FIGURE 7.14: **Segmentation performance metrics as a function of K for NMF and k-means.** (A) Pairwise F-measure *pfm*, (B) Hit rate F-measure Fm_3, (C) Under-segmentation score S_u and (D) Over-segmentation score S_o. For each metric, the scores being shown are the average scores across the entire dataset, for each value of K.

is to be expected that this observation generalises well, irrespectively of the value of the optimal rank.

7.4.6.2 Methods comparison

At the exception of the baselines NMF-K_e and K-means-K_e, all segmentation methods have been run for a range of parameter values. For synthetic result analysis, we present for each method the results obtained with the parameter configuration leading to the highest pairwise F-measure *pfm* in Table 7.1. The best result for each metric is highlighted in bold characters. However, it is to be noted that the parameter configurations that produce the highest *pfm* also produce the highest hit rate F-measure and S_f in the vast majority of the cases. In other words, the performance of the methods tends to peak in the same area of the parameter space for all F-measure metrics.

Reminder: Methods summary

ARD	β-NMF-based Automatic Relevance Determination
L_β-S-β-NMF	L_β-Sparse β-NMF (proposed algorithm)
SNMF-S	Monotonic L_1-Sparse NMF
SNMF-L	L_1-Sparse NMF
NMF-K_e	NMF with heuristic automatic rank determination
K-means-K_e	K-means with heuristic automatic rank determination
SSM Foote	Self-Similarity Matrix and checkerboard kernel

Reminder: Evaluation metrics summary

ppr	Frame clustering pairwise precision rate
prr	Frame clustering pairwise recall rate
pfm	Frame clustering pairwise F-measure
Fm_3	Hit rate F-measure (3s window)
Fm_8	Hit rate F-measure (8s window)
S_o	Normalised conditional entropy over-segmentation score
S_u	Normalised conditional entropy under-segmentation score
S_f	Normalised conditional entropy F-measure
TTG	True-to-guess median deviation
GTT	Guess-to-true median deviation

Considering the pairwise frame clustering metrics (i.e. *ppr*, *prr* and *pfm*), it is interesting to note that the ARD method leans towards high recall whereas other NMF-based methods lean towards higher precision, with the exception of the L_β-S-β-NMF which exhibits a very balanced performance. ARD with $\beta = 1$ produces the best *pfm* performance, with L_β-S-β-NMF closely following. The examination of under- and over-segmentation scores reveals that all methods tend to over-segment more than they under-segment ($S_o < S_u$) and L_β-S-β-NMF produces the highest S_f score.

It may be noted that the maximum hit rate F-measure achieved across all methods and all configurations is 0.42 with a 3s threshold. The median GTT represents the median distance between an extracted boundary and the closest annotated boundary. The values obtained for median GTT are significantly higher than 3s for all methods. For instance,

TABLE 7.1: Segmentation performance results for all methods considered. For each method, we present the results obtained with the parameter configuration leading to the best *pfm*. For each metric, the highest score is in bold characters and the symbol * is used to denote a statistically significant difference to the highest score ($p < 0.05$).

Methods	*ppr*	*prr*	*pfm*	S_o	S_u	S_f	Fm_3	Fm_8	TTG(s)	GTT(s)
ARD $\beta = 0$	0.59*	**0.92**	0.66*	0.59*	0.58*	0.58*	0.22*	0.38*	32.88*	10.51*
ARD $\beta = 1$	0.70*	0.91	**0.75**	0.69	0.70*	0.70	**0.42**	**0.53**	12.65*	**6.63**
ARD $\beta = 2$	0.61*	0.84*	0.66*	0.61*	0.63*	0.62*	0.28*	0.36*	27.97*	14.82*
L_β-S-β-NMF	0.77	0.78*	0.73	**0.72**	0.84	**0.75**	0.41	0.52	5.82*	9.00
SNMF-S	0.84	0.50*	0.57*	0.58*	0.85	0.69	0.31*	0.43*	5.15	19.75*
SNMF-L	0.87	0.44*	0.52*	0.54*	0.87	0.67	0.31*	0.44	**3.96**	19.33*
NMF-K_e	0.80	0.67*	0.67*	0.66*	0.81*	0.73	0.39*	**0.53**	8.55*	12.80*
K-means-K_e	**0.89**	0.47*	0.57*	0.61*	**0.90**	0.73	0.37*	0.45	4.78	17.64*
SSM Foote	0.66*	0.81*	0.68*	0.68*	0.68*	0.68*	0.07*	0.42*	24.10*	15.22*

in the case of ARD $\beta = 1$, the median GTT is 6.63s, meaning that many extracted boundaries are located outside of the 3s window required for them to be counted as a hit. For every NMF+HMM method, raising the hit rate threshold from 3s to 8s improves the F-measure score by about 0.1 points, which suggests that the precise localisation of the boundaries is a significantly challenging problem which should be a focus of future work. The effect is even more pronounced in the case of novelty-based segmentation (SSM Foote), which suggests that the NMF+HMM strategy leads to more precise boundary locations estimates than peak-picking a Foote novelty curve.

Overall it appears that L_β-S-β-NMF and ARD with $\beta = 1$ share the highest scores on all F-measure metrics (i.e. *pfm*, S_f,Fm_3 and Fm_8), often exhibiting close scores. These are the only two methods to consistently equal or outperform SSM Foote and automatic rank determination baselines and may therefore be considered as the two best performing methods. They are also the two methods enforcing the strongest sparsity constraints in the NMF decomposition. In addition, SNMF-S performs best when the weight of its sparsity constraint, which is comparatively weaker, is extremely large ($\alpha = 100$). This suggests in more general terms that very strong sparsity constraints are beneficial for the quality of the segmentation produced.

TABLE 7.2: Best results for each metric for each method. As a consequence, for each method the parameter configurations leading to the best result might differ from metric to metric. For each metric, the highest score is in bold characters and the symbol * is used to denote a statistically significant difference to the highest score ($p < 0.05$).

Methods	*ppr*	*prr*	*pfm*	S_o	S_u	S_f	Fm_3	Fm_8	TTG(s)	GTT(s)
ARD $\beta = 0$	0.59*	**1.00**	0.66*	0.61*	0.59*	0.60*	0.24*	0.38*	18.2*	**0.21**
ARD $\beta = 1$	0.85	**1.00**	**0.75**	0.71	0.85	0.73	**0.42**	**0.53**	**0.69**	0.67
ARD $\beta = 2$	0.85	**1.00**	0.66*	0.63*	0.86	0.71	0.34*	0.48	4.18	0.73
L_β-S-β-NMF	0.83	0.98	0.73	**0.72**	0.84	**0.75**	0.41	0.52	5.82	9.00*
SNMF-S	0.88	0.50*	0.57*	0.58*	0.88	0.69	0.32*	0.44	3.75	19.75*
SNMF-L	0.87	0.44*	0.52*	0.54*	0.88	0.67	0.33*	0.44	3.54	19.33*
NMF-K_e	0.80	0.67*	0.67*	0.66*	0.81*	0.73	0.39*	**0.53**	8.55*	12.80*
K-means-K_e	**0.89**	0.47*	0.57*	0.61*	**0.90**	0.73	0.37*	0.45	4.78	17.64*
SSM Foote	0.66*	**1.00**	0.68*	0.68*	0.68*	0.68*	0.07*	0.46	24.10*	6.67*

7.4.6.3 Performance upper bound

While Section 7.4.6.2 aims at analysing results for a typical optimal parameter configuration, this section investigates the maximum performance achievable for each method across its parameter space with respect to all metrics. In particular, Table 7.2 presents in each of its cells the best result obtained for a given method with respect to a given metric. As a consequence, for each method the parameter configurations leading to the best result might differ from metric to metric. The best result according to each metric is highlighted in bold characters.

Firstly, it appears that a sizeable number of the maximum scores reported in Table 7.2 exactly correspond to those reported in Table 7.1. Detailed inspection revealed that the parameter configuration leading to these scores is identical in both cases. It is particularly true of F-measure type of metrics (i.e. *pfm*, S_F, Fm_3 and Fm_8). This reveals that the optimum compromise between recall and precision tends to be found in the same area of the parameter space for all F-measure type of metrics. As a result, this observation augments the generality of the conclusions drawn from the analysis carried out in Section 7.4.6.2.

Some observations persist from the analysis exposed above. In particular, for the vast majority of methods, the maximum over-segmentation score is lower than the maximum under-segmentation score ($S_o < S_u$). This means that, even with the most optimal parameter configuration, these methods tend to be biased towards over-segmentation

more than under-segmentation. Considering the pairwise frame clustering metrics reveals that the L_β-S-β-NMF, ARD and SSM Foote methods are capable of generating higher recall than precision as opposed to other NMF-based methods as well as K-means-K_e. Hit rate F-measure scores (Fm_3 and Fm_8) are not significantly greater than those reported in Table 7.1. This results corroborates the observation made earlier that extracting precise boundary positions is a challenge for all techniques considered here.

The median true-to-guess (TTG) and median guess-to-true (GTT) are reported here for consistency, although a meaningful interpretation of their value cannot be made. These metrics are only informative in conjunction with other metrics. For instance, in the case where only very few boundaries are retrieved and that these boundaries happen to be very close to the reference annotation boundaries, the median deviation will be very small. Taken in isolation, this figure would suggest a good performance in terms of boundary localisation, but may not be significant.

Here again L_β-S-β-NMF and ARD with $\beta = 1$ exhibit very close scores. This result is consistent with the theoretical similarity between these two techniques highlighted in Section 7.4.3.4. These two techniques also compare with or outperform most of other methods, which validates further their relative superiority. We also note that the baseline methods NMF-K_e and K-means-K_e tend to outperform the common sparse-NMF approaches (SNMF-L and SNMF-S) and thereby validates the application of very strong sparsity penalties.

7.4.6.4 Comparison with state of the art structural segmentation algorithms

The task of detecting metric modulations is formulated as a segmentation problem in this thesis. Because it is, to the best of our knowledge, the first attempt at detecting metric modulations with such an approach, there is not prior art to compare our results to. In order to somewhat contextualise our results, in this section we compare them to results obtained by state of the art algorithms on a similar but different task that is a well documented area of MIR research, namely structural segmentation, in terms of musical form such as Verse, Chorus, Bridge etc. (cf. Section 2.10).

For comparison, we consider a selection of some of the best performing algorithms submitted to the MIREX Structural Segmentation task from 2012 to 2016 [3]. Table 7.3 regroups the corresponding results obtained on the SALAMI dataset (cf. Section 3.1). The comparison of algorithms performance is straightforward because the metrics used in the previous sections of this chapter are identical to those used in the MIREX challenge, with the exception of Fm_8. As a results the results of Table 7.3 should be compared to those of Table 7.1.

The order of magnitude of performance according to *ppr*, *prr*, S_u and S_o metrics is comparable in the two cases. L_β-S-β-NMF and ARD-NMF exhibit a sizeable positive difference in terms of pairwise clustering F-measures *pfm* and S_f with respect to best performing structural segmentation algorithms. On the other hand, they produce lower performance in terms of hit rate F-measure (Fm_3) and median deviations (GTT and TTG). This reveals that state of the art structural segmentation algorithms produce more accurate boundary locations that the metrical-structure based segmentation methods we investigated here. Overall, in comparison to the state of the art in structural segmentation, the methods we proposed here tend to produce a segmentation that overlaps very well with the annotated segmentation, but that does not produce very accurate segment boundary locations. These results show that it is possible to achieve better boundary precision on a task analogous to the one considered in this thesis (i.e. another segmentation task) and therefore suggest that there may be a good prospect for improving metric modulation boundary detection accuracy by taking inspiration from structural segmentation methods. Therefore we assume that improving on the boundary location accuracy in future work could bring a gain to the quality of metrical structure based segmentation.

Boundary location accuracy aside, the methods investigated in this chapter seem to produce segmentation that is significantly closer to the annotated segmentation than state of the art structural segmentation algorithms are. It is now well understood that the structure of a musical piece, in terms of musical form, is multifaceted [178] and happens

[3] http://www.music-ir.org/mirex/wiki/2016:Structural_Segmentation

TABLE 7.3: Performance score of the best performing algorithms in the MIREX Structural Segmentation task over the years 2012-2016 for the SALAMI dataset

Algorithm	*ppr*	*prr*	*pfm*	S_o	S_u	S_f	Fm_3	TTG(s)	GTT(s)
GS1 [230]	0.41	**0.80**	0.51	0.54	0.82	**0.65**	0.62	2.20	**2.37**
SUG1 [231]	0.73	0.44	0.48	0.52	0.74	0.61	0.61	6.20	2.75
SUG2 [231]	**0.88**	0.30	0.42	0.45	**0.88**	0.60	**0.69**	**1.44**	4.15
MHRAF1 [232]	0.56	0.67	0.57	**0.63**	0.52	0.57	0.42	6.24	5.50
SMGA1 [233]	0.68	0.58	**0.58**	0.62	0.68	**0.65**	0.49	1.77	6.71

at multiple layers simultaneously [172], which makes it very hard to define what is a "correct" segmentation. Such considerations have motivated the creation of more elaborate evaluations and datasets such as SALAMI [182] but also reveal the difficulty to be faced when designing automatic systems to retrieve structural segmentation in this context. Conversely, in this chapter we focus on metric modulations, for which we can provide a definition that is sharper than in the case of structural segmentation. While choosing which musical feature(s) to base the automatic structural segmentation algorithms on is a difficult and dataset-dependant exercise, the metric modulation detection algorithms investigated here are naturally restrained to metrical structure changes. In fact, in its current formulation, metric modulation detection can be seen as a sub-task of the more general structural segmentation task.

7.4.6.5 Performance per modulation class

The previous sections have concentrated on evaluating the automatic metrical structure based segmentation, irrespective of the nature of the modulations. Let us now consider the following question: "Are some types of modulation harder to detect than others?"

For each annotated boundary, the corresponding reference metric modulation type is derived from the annotation of the two adjacent segments on the basis of the adapted taxonomy presented in Section 6.3. This is then compared to the automatically produced boundaries. The number of successfully detected boundaries per type of modulation is counted. A successful boundary detection is defined in line with the hit rate metric: a hit is recorded if an extracted boundary is present in a window around the annotated boundary, a miss is recorded otherwise. Consistently with the Fm_3 and Fm_8 metrics

presented earlier, windows of 3s and 8s are used. For each modulation type, the recall rate is calculated

$$\text{Modulation Recall} = \frac{TP}{TP + FN} \tag{7.34}$$

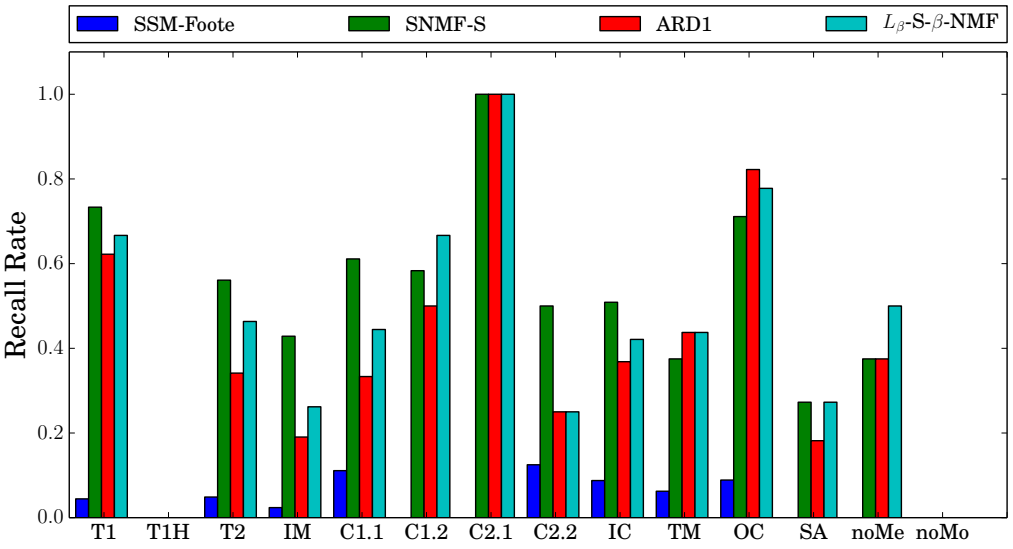
where TP is the number of true positives, i.e. the number of hits, and FN is the number of false negatives, i.e. the number of misses. The recall rates for each metric modulation type for a selection of method and for both the 3s and 8s hit threshold window are presented in Figure 7.15.

In this case the results are naturally influenced by the overall recall of the each method. The increase in absolute performance when raising the hit threshold from 3s to 8s already reported in Table 7.1 is apparent on Figure 7.15 (A). For instance, it is very obvious that the SSM-Foote method performs very poorly with a hit rate window of 3s. There are no examples of T1H nor noMo modulations in the annotations of this dataset, so that such boundaries cannot be recalled.

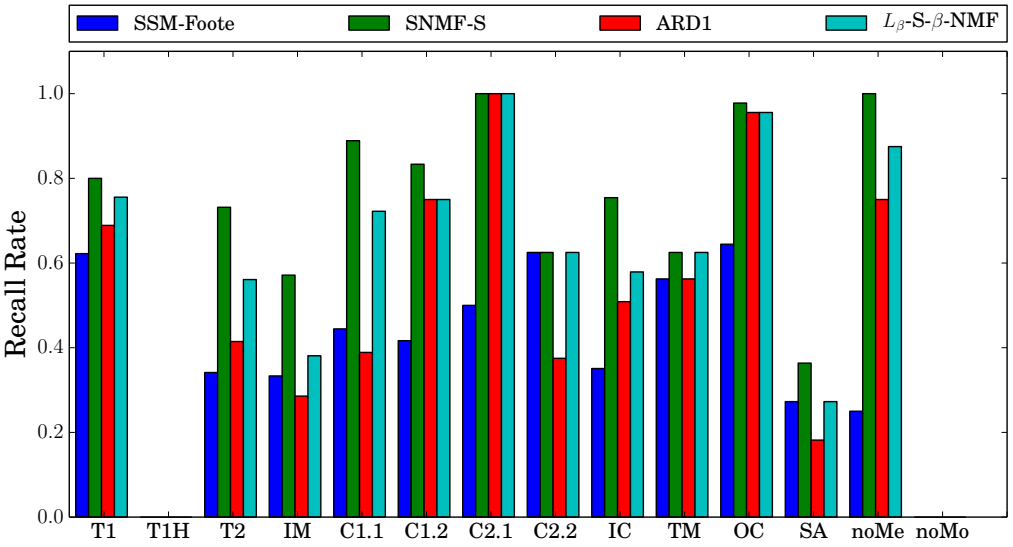
In order to get insight into the relative performance against metric modulation types, we normalise the Modulation Recall by the Global Recall (i.e. the recall for the whole dataset):

$$\text{Normalised Recall} = \frac{\text{Modulation Recall}}{\text{Global Recall}} \tag{7.35}$$

As a result, Figure 7.16 gives a graphic representation of the relative boundary retrieval performance across modulation types. Overall it appears that, with the exception of a couple of outliers, the tendencies are similar for every method. In other words, all the methods considered appear to respond in a comparable fashion to the type of metric modulation performance-wise. In particular, SA and IM modulations appear to be comparatively more challenging for the algorithms than other types of modulations whereas C2.1, OC, T1 and C1.2 are the easiest modulations to retrieve. This relative performance discrepancy may be related to the amount of change in energy distribution in the metergram they result in. We refer the reader back to Section 6.3 in which the diagrams for each metric modulations closely relate to the energy distribution change in a metergram around the modulation. For instance, Subdivision Addition (SA) is a subtle change in



(A) Hit threshold window = 3s



(B) Hit threshold window = 8s

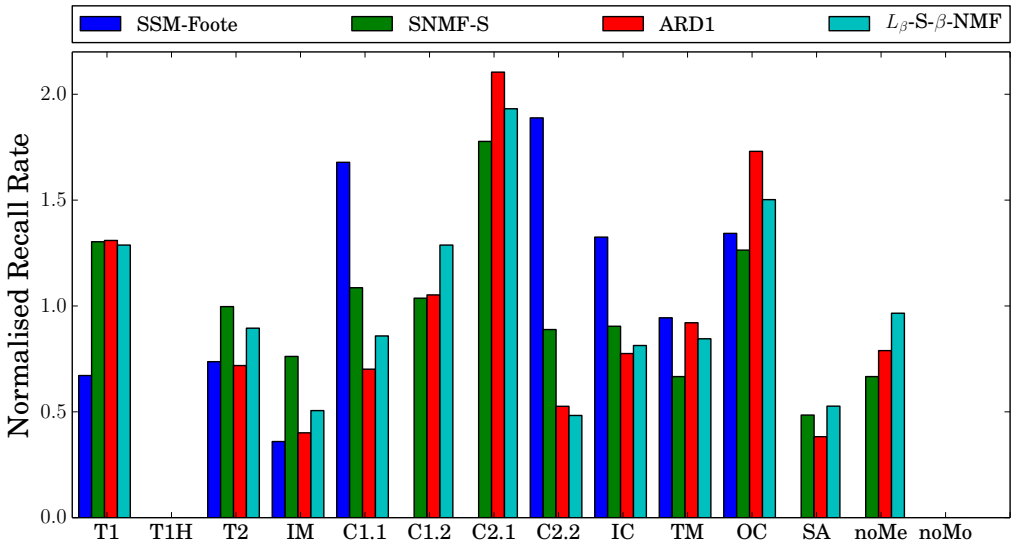
FIGURE 7.15: **Recall rate of metric modulation detection, per modulation type.** Results are presented for four NMF-based segmentation methods corresponding to the results presented in Table 7.1. (A) With a 3s hit rate threshold window, (B) With a 8s hit rate threshold window.

the sense that it implies that all metrical levels are preserved and one is added. Conversely, OC implies that no metrical level pulse rate is preserved by the modulation. In other words, in the latter case, the change of energy distribution in the metergram frames around the modulation is far greater than in the former case. The methods considered in this chapter for automatic segmentation proceed either by detection or novelty of frame clustering. The results obtained in this section corroborate the idea, which can be intuitively formulated, that detecting modulations discriminating two segments in which the energy distribution in the metergram frames is similar is a more arduous task than when the energy distributions are drastically different. In other words, the results obtained in this section demonstrate that metric modulations inducing small alterations of metrical structure are harder to detect than modulations inducing more substantial changes.

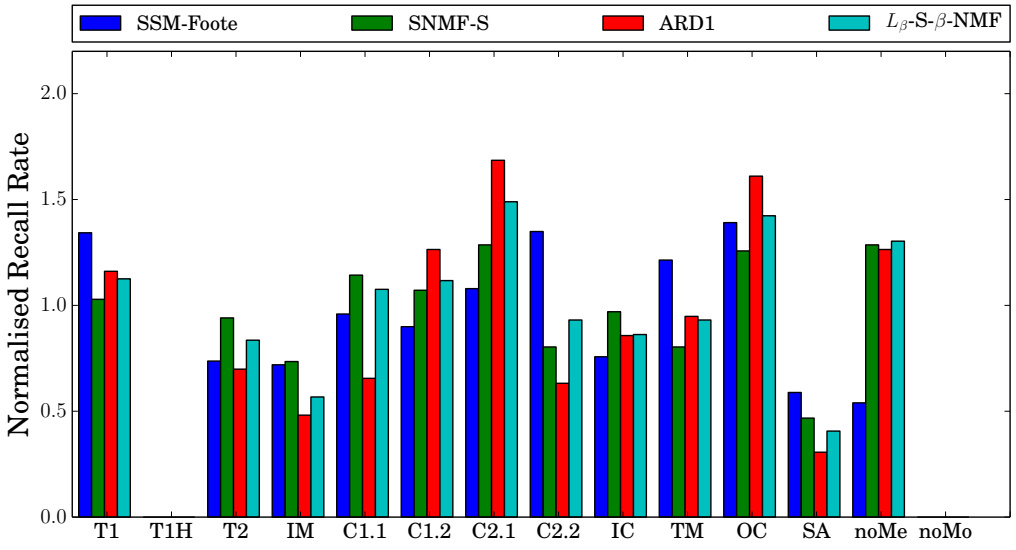
7.5 Conclusions

In this chapter we proposed to consider automatic metric modulation detection as a task in itself and formulated it as a segmentation retrieval problem for the first time. We assumed the metric modulations to occur as a change from one segment of relatively stable metrical structure to another segment of relatively stable, but different metrical structure. No other prior knowledge prior knowledge was assumed and we therefore proposed to address this problem in an unsupervised fashion.

In order to perform this segmentation retrieval task from audio we proposed an extension of existing rhythmogram processing in order to form the *metergram*, which is the feature from which the segmentation is estimated. Both novelty-based as well as homogeneity-based methods were applied to the metergram in order to retrieve the segmentation. While Foote's novelty-based method detects metric modulations as changes of energy distribution in the rhythmogram frames, a range of NMF-based methods were considered to cluster together frames of consistent metrical structure. The standard k-means algorithm was employed for comparison. We additionally introduced a new variation of sparse-NMF algorithm to the existing methods considered. Finally, we proposed to



(A) Hit threshold window = 3s



(B) Hit threshold window = 8s

FIGURE 7.16: **Normalised recall rate of metric modulation detection, per modulation type.** Results are presented for four NMF-based segmentation methods corresponding to the results presented in Table 7.1. (A) With a 3s hit rate threshold window, (B) With a 8s hit rate threshold window.

employ a Hidden Markov Model on the data representation learnt in order to produce the final segmentation estimate in the case of homogeneity-based methods.

Experiments have demonstrated that optimal rank estimation is critical to obtain meaningful results using standard NMF, although the rank information is not known a priori. Experiments have also shown that enforcing very strong sparsity constraints in NMF decompositions allows to circumvent the rank estimation problem, therefore making this technique practically usable and also leading to the best performance amongst all method considered. Since tackling the detection of metric modulations as a segmentation problem had never been attempted before, no prior art is available to compare our results to. We therefore contextualise them by comparing our results to the results obtained by state of the art algorithms on a different, but similar task, namely structural segmentation. We then demonstrated that our proposed approach is competitive with state-of-the art structural segmentation algorithms. More precisely, it outperforms state of the art algorithms on *pfm* and S_f metrics, while it is still inferior on Fm_3 metric. Exploiting the taxonomy of metric modulations introduced in Chapter 6, we showed that not all metric modulations are equally easy to detect: modulations inducing substantial alterations of the metrical structure tend to be easy to detect and vice-versa.

Chapter 8

Conclusion

8.1 Summary

This thesis took a number of steps towards the automatic analysis of metric modulations. After giving a general introduction and laying out the research questions to be addressed in Chapter 1 and then reviewing the necessary elements of background in Chapter 2, we presented the datasets used in this work in Chapter 3. In particular, we introduce two new datasets for the evaluation of metrical structure and metric modulation detection algorithms respectively. In addition, we evaluated the inter-annotator agreement in the GTZAN-Met dataset, which enabled us to isolate intractable tracks and asses the upper limit of performance achievable by an algorithm on this dataset.

In Chapter 4 we investigated the use of the rhythmogram as a feature to estimate the metrical structure, thereby assessing its suitability to lay the ground for the detection of metric modulations. Our experiments demonstrate that the metrical pulse rates are related to peaks in the periodicity spectrum but that the reciprocal proposition is not necessarily true. On that account we proposed a peak-picking algorithm that enforces hierarchical constraints derived from music theory to extract metrical levels from the periodicity spectrum. We then demonstrate its efficiency by evaluating it on the GTZAN-Met dataset. Moreover, comparing the algorithm scores to inter-annotator agreement

revealed that its performance closely approaches human performance on a number of genres while there is still room for improvement for tracks in the Jazz genre.

Chapter 5 focused on the estimation of rhythm feature extraction reliability. As a matter of fact, feature extraction algorithms occasionally fail. Since feature extraction is often the first step in a mote complex system (e.g. recommendation engine) or to compute composite features (e.g. beat-synchronous chromagram), failures in feature extraction lay unstable ground for the system. On the other hand, some causes of systematic failure are well known to the MIR research community. On that account, our proposed approach to estimating reliability of rhythm features extraction consists in capturing properties of the musical signal that are known for causing failures. We then demonstrate that the entropy of the rhythmogram may be used as a predictor of the reliability of the extraction of several rhythm features.

Finally, in chapters 6 and 7 we introduce a metric modulation taxonomy with its corresponding classifier and address the automatic detection of metric modulations, respectively. Considering metric modulations as changes of metrical structure over time, we proposed to approach their automatic detection as a metrical structure based segmentation retrieval problem, the segment boundaries representing the temporal location of the modulations. Furthermore, we consider a scenario in which no prior knowledge about the modulations or metrical structure is assumed. In this context, we propose to address the automatic detection of metric modulations in an unsupervised fashion and considered both novelty-based and homogeneity-based segmentation retrieval strategies. A variety of numerical methods were examined to perform metergram frame clustering in the context of homogeneity-based segmentation. We showed that the estimation of the rank of the decomposition (i.e. the number of clusters used) is critical for obtaining good segmentation results. We therefore introduce an algorithm for computing a β-sparse β-NMF that enables the automatic pruning of components and therefore overcomes the rank estimation problem. Our results showed that a NMF approach enabling automatic rank estimation (via pruning of un-necessary components) outperforms all other methods under scrutiny. In order to enable a musicologically meaningful analysis, we proposed for the first time to classify metric modulations against a dedicated taxonomy. Since, to

the best of our knowledge, there does not exist a taxonomy that relies on features automatically extractable from musical recordings, we reformulated a taxonomy designed for score-based musicological analysis in terms of metrical level pulse rates. By doing so we showed that not all modulations are equally easy to detect: modulations implying large changes of metrical level pulse rates are typically easier to detect than modulations implying only subtle changes.

In the following sections we relate our investigations to the research questions established in Chapter 1 and subsequently deduce possible directions for future work.

8.2 Discussion of Research Questions

RQ1: How can we automatically estimate the metrical structure of music?

This thesis is concerned with the automatic analysis of metric modulations, that is to say changes of metrical structure over time. On that account, it is related to the topic of metrical structure estimation, which, in the current state of the art, is by no means a solved problem. Although the aim of this thesis is not to contribute to the advance of metrical structure estimation methods, it is necessary for the analysis of metric modulation to capture at least some properties of the metrical structure.

Relating to RQ2, we demonstrated that the rhythmogram feature, and in particular the *metergram* variation of it, has the ability to capture, in an unsupervised fashion, some metrical structure information on which a metric modulation detection system can be based. However, we showed in Chapter 4 that there is no direct mapping between the periodicity rates observed in the metergram and metrical pulse rates. Since the metrical structure typically exhibits hierarchical organisation, we introduced a peak-picking algorithm enforcing hierarchical constraints to retrieve metrical pulse rates from the rhythmogram. Taking the inter-annotator disagreement in account (cf. RQ3), we demonstrated that applying this algorithm to the metergram very closely matches human

performance on a number western popular music genres. It also shed a light on some weaknesses that will be discussed in Section 8.3.

RQ2: To which extent does the rhythmogram capture the metrical structure of music?

It has been suggested by a number of authors that the rhythmogram is capable of capturing metrical structure related information, and more precisely that peaks in its frames (that we call periodicity spectra) correspond to metrical level pulse rates. Nonetheless, it is also widely known that 'harmonics' of the metrical level pulse rates are present in the periodicity spectra (cf. Chapter 2). The lack of congruence between these two observations motivates the a direct examination of their respective limits. In Chapter 4 we presented an experiment investigating the correspondance between metrical level pulse rates and peaks in the rhythmogram.

Our results confirm the informal observations made by previous authors: the metrical level pulse rates correspond to peaks in the FFT and ACF rhythmogram, but not all the peaks correspond to metrical level pulse rates. Starting from the idea that the FFT rhythmogram produces harmonics of the metrical pulse rates while the ACF rhythmogram produces sub-harmonics, Peeters proposed multiply the FFT and ACF rhythmograms in order to eliminate harmonics so that only the peaks corresponding to metrical level pulse rates remain. This hypothesis had not been explicitly tested, however. Our experiments show that the correspondance between metrical level pulse rates and peaks in the rhythmogram is greatly improved using this method, even though some harmonics persist. In the light of this result we chose to use the multiplication of the FFT and ACF rhythmogram (that we label 'metergram' for clarity) as a feature for extraction of metric modulations.

RQ3: What is the impact of human judgment discrepancies on the evaluation of automatic metrical structure extraction algorithms?

MIR algorithms are typically evaluated against expert human annotations, which are then considered as 'ground truth'. It is then clear that if the annotations are provided by one annotator only, the evaluation is merely a measure of the ability of the algorithm to reproduce the annotator's bias. The generalisability of results obtained in such a setting is therefore questionable. This issue can be addressed by collecting multiple annotations (from distinct annotators) for each musical excerpt in a dataset and accounting for potential discrepancies between these. In Chapter 3 we presented a dataset of metrical structure annotations in which every track was annotated by two to three distinct annotators. In Chapter 4, we evaluated our metrical structure extraction algorithm using our newly created dataset.

It has been shown in previous work that the level of inter-annotator agreement puts an upper bound to algorithm performance. By evaluating both the algorithm and the level of inter-annotator agreement we demonstrated that a performance score might be misleading when not compared to inter-annotator disagreement. For example, the algorithm relatively performed poorly on tracks in the 'classical' genre category, but human annotators also tend to disagree a lot on this subset of the corpus so that the average performance of the humans and the algorithm is identical. As a result, if the inter-annotator agreement had not been evaluated, the performance score would have suggested that there is room to improve the algorithm on data resembling that in the 'classical' genre category when it is not actually the case: it already performs as well as human experts do.

In Chapter 3 the agreement between annotators (as well as self-agreement) was examined in greater detail. This analysis enabled us to draw additional conclusions regarding the impact of inter-annotator agreement on algorithms evaluation. In particular, tracks for which human annotators were not able to produce an annotation were isolated and labelled *intractable* on that account. It is thus unclear how to meaningfully evaluate algorithms against such intractable tracks. For that reason, we excluded them from the evaluation procedure in Chapter 4. Overall, the evaluation of human annotations used

to evaluate algorithms appears to greatly improve the quality of the conclusions that can be drawn.

RQ4: How can automatic feature extraction failures be predicted?

Some properties of the musical signal are known to create challenging conditions for rhythm features extraction. One example of such property is 'soft onsets', which are known to be the cause of mediocre rhythm feature extraction performance at best. As a result, we hypothesised that measuring these properties should enable the computation of a predictor of the reliability of rhythm features extraction. Chapter 5 is dedicated to the investigation of this research question.

We consider a feature extraction system to be reliable if it consistently produces estimates that are deemed to be good. We then proposed to use the entropy of a rhythmogram as a predictor of the feature extraction reliability. Our experiments demonstrated that this descriptor relates to the reliability of three rhythm-related tasks, namely tempo estimation, beat tracking and metrical structure estimation.

RQ5: How can we automatically detect metric modulations?

A metric modulation is defined, for the purpose of this thesis, as a relatively abrupt change of metrical structure over time. This implicitly assumes a segmentation of musical pieces in sections of relatively consistent metrical structure, the boundaries of which represent metric modulations. On that account, we formulated the problem of automatic detection of metric modulations as a metrical structure based segmentation retrieval task. Relating to RQ1 and RQ2, it appeared that the metergram feature captures attributes of the metrical structure and we therefore proposed to use it as a feature on which to base the detection of metric modulations. We proposed an unsupervised approach to perform the segmentation and examined a number of methods in Chapter 7.

Two classes of approaches were employed: novelty-based and homogeneity-based segmentation. Novelty methods aim at detecting metric modulations by detecting a change

of metrical structure while homogeneity-based methods aim at clustering together contiguous regions of consistent metrical structure. Clustering algorithms typically require the number of clusters, i.e. the number of different metrical structures to be found to be set in advance. However, in the blind scenario we are considering, this information is not known a priori. In order to circumvent this issue we introduced a new algorithm to compute a β-NMF decomposition with β-sparse constraints, which automatically prunes out un-necessary components. Results show that the decomposition rank selection is instrumental in achieving good performance and reveal state of the art performance for our proposed method.

RQ6: Can computational methods be used to automate musicological analyses of metric modulations?

Traditional musicology is typically carried out by relying on a score for analysing musical features through the lens of a specific research question or framework, see for example [234–236]. Musicological questions typically involve the study of high (semantic) level, and possibly abstract, concepts such as leitmotif whereas computational methods are typically better suited for low level analysis. As a consequence, addressing musicological questions with computational methods remains a very challenging task.

In a bid to progress in this direction, we proposed to employ a taxonomy of metric modulations. By grounding the taxonomy in musicological theory, the musicological depth is hard coded in the structure of the taxonomy itself. Starting from an existing metric modulation taxonomy designed for score-based analysis and given that a score is not available for all musical recordings, we proposed in Chapter 6 to adapt Bouchard's taxonomy so that it is applicable to computational analyses. In particular we reformulated Bouchard's taxonomy so that it relies on features that can be automatically extracted from audio recordings, namely the metrical level pulse rates. On this basis, we constructed a metric modulations classifier.

The comparison of the classification results obtained with the annotated metrical pulse rates from the GTZAN-Met dataset with automatically extracted metrical pulse rate revealed that the classification accuracy obtained from automatically extracted features

depends on the which level of granularity of the taxonomy is considered. While the classification of families of metric modulations showed promising results, the classification into individual modulation categories was significantly less accurate. The discussion of this result uncovers directions for future work, which we will examine in the next section. Finally, we have shown in Chapter 7 that this musicologically-driven and taxonomy-based classification is useful to help gaining a deeper understanding of the of segmentation performance: categories of metric modulations that imply a large change in the metrical structure are easier to detect than metric modulations implying only subtle changes.

8.3 Future Work

Finally, in this section we outline the main avenues for future work that could be identified on the basis of the results obtained in this thesis.

Overcoming the onset detection bottleneck

As described in Chapter 2, onset detection is the first step of the prototypical MIR rhythm analysis pipeline. The rhythm analyses are then derived from a feature analogous to an onset detection function. Moreover, an onset canonically characterises the time instant (of zero duration) associated with the beginning of a musical event, often closely related to the concept of sound transient. The results of Chapter 4 and Chapter 5 as well as a number of observations reported in the literature tend to suggest that the limited scope of this type of approach to onset detection puts a limit to the performance of subsequent analyses. In other words, when the musical signal does not feature the characteristics that enable reliable onset detection, the premises on which systems for the estimation of rhythm attributes rely are not verified, which leads to failures.

In the recent years, the introduction of neural networks based onset detection algorithms has improved the performance, but the underlying estimation paradigme has seldom been modified. As noted by Schlüter [75], onset detection is approached as the audio analogue to edge detection, i.e. local (ideally punctual) changes.

Great successes have been achieved in recent work in computer vision to detect complex objects (e.g. faces) by applying neural networks to images. Systems producing the expected result for a given task (i.e. detection of faces) by directly feeding the raw data (e.g. the image) to a neural network are known as *end-to-end* systems. The network architectures are typically hierarchically structured so that the first layers typically learn elementary filters such as edge detectors and topmost layers learn representations of complex objects [237]. Recent network introspection studies tend to suggest that this assumption is verified [238, 239]. On the other hand it has also been shown that end-to-end learning can be applied to music [100, 101]. Pursuing the analogy with computer vision, it may therefore be hypothesised that if neural networks can learn to recognise a variety of complex objects they may also be capable of learning to recognise complex musical objects — i.e. non punctual onsets. For these reasons, we view the design of end-to-end rhythm analysis systems as a possibly promising avenue for future work that may enable to better handle challenging musical signals, such as pieces featuring with soft onsets.

Towards a better location of metric modulation boundaries

It has been shown in Chapter 7 that regardless of the numerical method used for homogeneity-based segmentation a higher *pfm* than Fm_3 or Fm_8 is always obtained. Moreover, in all cases the Fm_8 is about 0.1 points greater than Fm_3. These results suggest that although the methods under scrutiny tend isolate the right segments, their ability to precisely locate segment boundaries (i.e. metric modulations) is comparatively inferior. This tendency is even clearer in the case of novelty-based segmentation. An avenue for future work may therefore consist in improving the accuracy of metric modulations location. Given that all methods rely on the metergram and that the trend we observed is shared by all methods, we hypothesise that the source of this imprecision may lie in the computation of the metergram. More precisely, the windows used for computing the metergram are very long (12s) so that long metric cycles (e.g. bar cycle) may be captured. The use of such long windows implies a smearing of the time evolution of metrical structure. One way to overcome this limitation may be to employ

a multi-resolution metergram, thereby combining long windows that allow the capture of long metrical cycles and shorter windows that enable more accurate location of metric modulations. Another approach may consist in combining the metergram with other features providing a finer temporal resolution, e.g. using the metergram to produce a rough boundary location estimate that may then be refined by synchronising it with the closest downbeat.

It has to be noted, however, that seeking to capture accurate metric modulation locations in the terms used above implicitly assumes that a modulation is a punctual segment boundary. But this is a somewhat simplistic model. Metric modulations are transitions between one segment of consistent metrical structure to another segment of different structure, and are typically set up so that they unfold in a musically meaningful way. Defining the temporal position of a metric modulation is an arduous task, even from a music theoretical point of view. We thus argue that aiming at characterising metric modulations as a region of change rather than an punctual alteration may be a fruitful approach for future work. We also note that this concept is applicable to other segmentation tasks for which a punctual boundary model does not accurately represents the musical content.

Towards a better metric modulation taxonomy

The results of automatic classification of metric modulations obtained in Chapter 6 and Chapter 7 revealed that a meaningful insight can be gained from the classification in families of metric modulations but that there is room for improving the classification in finer individual modulation types, as suggested by the number of undefined modulations detected. In its current form, the metric modulation classifier is made of hand-crafted modules that make binary classification decisions. Our results suggest that either the modulation types specified were not optimally chosen or that binary decisions is too harsh a process for metric modulations classification, or a combination of both.

As a consequence, future work should investigate the design of a metric modulation taxonomy that relies on softer constraints, possibly refining or refactoring the modulation types considered here. We recall that the reason for introducing a metric modulation

taxonomy was to enable musicologically meaningful automatic analyses. Musicological considerations should therefore still be considered in future work. In particular, devising adequate metric modulation types on the basis of features that may be automatically extracted is not straightforward and we argue that involving musicologists in this process should be instrumental in improving the taxonomy. Secondly, since the hard classification architecture used in this thesis has shown limitations, we hypothesise that employing a framework enforcing softer constraints (e.g. probabilistic framework) should lead to improved classification performance in future work. Neural networks have been successfully applied to a variety of classification tasks and therefore represent a promising option for the technical implementation of such classification system. Given that neural networks learn latent representations from the data, it may be hypothesise in first approximation that they would learnt adequate representations of metric modulations. However, recent work shows that the representations learnt by neural networks do not necessarily respect sanity constraints of a given task [240]. A part of such future investigation may then consist in encoding the desired musicological constraints in the network architecture and/or via the training method [241]. As a further extension, rather than explicit annotations, the ground truth used for training such models may also be produced from psychological studies, or by directly learning models the the human perception of metric modulations.

Towards alternative musical concepts to circumvent intractability

When given the possibility, human annotators may chose not to annotate certain tracks because they are not able to do so (cf. Chapter 3 and existing work such as [3]). Such tracks were consequently labelled in this thesis as *intractable*. This observation raises the following question: why are these pieces intractable? One possible reason could be that the musical content of the piece is 'complex' (e.g. very fast, or including heavy syncopation) makes the annotation of the corresponding musical feature (say the metrical structure) difficult to track. The intractability then reflects the limited abilities of the annotators. In this case, intractability can be overcome by selecting annotators with more advanced skills. Another hypothesis could be that the task expected from the

annotator is not relevant to the piece at hand. For instance, the task of attempting to annotate the key is unlikely to be relevant to an atonal piece.

Going back to metrical structure, we have observed in Chapter 3 that the majority of intractable tracks were from the 'classical' genre category. Does this necessarily mean that human annotators are bad at tracking metrical structure of classical music? Or does it mean that the musical concepts underlying the task annotators have to complete do not optimally apply to classical music? Annotators were asked to annotated metrical level pulse rates, which implicitly assumes a somewhat steady underlying pulse. We therefore hypothesise that some of the classical pieces were intractable simply because they do not feature a steady underlying pulse or clear quasi-punctual onsets. However, it does not mean that these pieces do not have a metrical structure. Indeed different concept of metrical structure might be more relevant. Since some cases of intractability reveal the limits of our conceptual model of metrical structure future work should then consider alternative descriptions of the metrical structure that would fit musical pieces that do not feature a clear steady pulse.

8.4 Closing words

The estimation of the metrical structure is now an active area of MIR research. In most of the related works, the metrical structure is assumed either to be constant or to only slowly vary (e.g. slow tempo drifts, or local inconsistencies). The detection of metric modulations is comparatively rarely addressed. In contrast, in this thesis we have been concerned with taking a number of steps towards the automatic detection of metric modulations. We hypothesise that one of the reasons for the lack of works addressing the detection of metric modulation is the absence of appropriate dataset. In a bid to overcome this barrier, we created a publicly available dataset for the evaluation of metric modulation detection systems. With the aim of designing a method that would be as general as possible, we proposed an blind scenario in which we formulate the detection of metrical structure as a segmentation problem and no prior knowledge of the metrical

structure nor the metric modulations is assumed. We then proposed an unsupervised approach to address this task.

There is substantial scope for expanding on the work presented in this thesis. The major avenues for future research that could be derived from our results are outlined in Section 8.3. Some of them consist in technical or methodological improvements of the frameworks used for detecting metric modulations (e.g. improving the location of modulations). On the other hand, potential future research may concern more fundamental models of musical concepts, such as onsets. This is motivated by the fact that the limitations of these concepts are inherited by the methods that they underpin. Therefore, examining further existing concepts or devising new concepts is a necessary step to overcome these limitations.

Another aspect that may be of interest for future work but was not considered in this thesis is the applications of metric modulation detection. Like many other MIR tasks, in may be useful for applications in computational musicology or the management, navigation and discovery of large music collections. Regarding a metric modulation as of similar nature as some other MIR features, such as chords, structural segmentation or tempo, we envision that it would naturally integrate in systems employing MIR for computational musicology and database navigation. Creative applications may also be considered. For instance in systems designed for automatic sequencing or mashup creation. There already exist research endeavours and industrial applications of such systems. In the current state of the art, the transitions are typically generated by choosing two (or more) tracks having similar rhythmic attributes. Adding the capability for handling metric modulations would then enable the production of wider variety of transitions, potentially producing interesting rhythmic effects.

Although the task of metric modulation detection is currently seldom considered in current research, we view it as an exciting opportunity for development of field of MIR. This thesis presents an effort to progress in that direction and we hope to see more work carried out on this topic in the future.

Appendix A

Metric Modulations Dataset Tracklist

Trackname	Artist	Release	ISRC
2 + 2 = 5	Radiohead	Hail To The Thief	GBAYE0300801
Another Day	Paul McCartney	Wings Greatest	GBCCS0700020
Band On The Run	Paul McCartney and Wings	All The Best (US Version)	GBCCS0700034
I Want It All (Single Version)	Queen	The Miracle (Deluxe Remastered Version)	GBUM71029624
Live and Let Die	Paul McCartney and Wings	Live and Let Die	
Shankbon	The Slackers	Close my Eyes	
Superjeilezick - Original Version	Brings	Best Of	DEC680001342
The Lazarus Heart	Sting	...Nothing Like The Sun	USAM18700038
Tom Sawyer (Album Version)	Rush	Moving Pictures (2011 Remaster)	USMR18180103

Magical Mystery Tour	The Beatles	Magical Mystery Tour	
21st Century Schizoid man	King Crimson	In the Court of Crimson King	
Innuendo	Queen	Innuendo	
Free Bird	Lynyrd Skynyrd	Family	USMC17301722
Immediate Circle	Catatonia	Paper Scissors Stone	GBAHT0105618
Killing In The Name (Album Version)	Rage Against The Machine	Rage Against The Machine	USSM19200317
Old Dog	The Slackers	Close my Eyes	
...And Justice For All	Metallica	...And Justice For All	
Master of Puppets	Metallica	Master of Puppets	
Spaceman	Babylon Zoo	The boy with the X-ray eyes	
Child In Time	Deep Purple	Deep Purple In Rock	USWB19903563
Dracula Mountain	Lightning Bolt	Wonderful Rainbow	US33K0404103
Geno - (Tribute to Dexys Midnight Runners)	Studio Allstars	Music From Ashes To Ashes Series 1	GBQRF0814853
Rapunzel	Dave Matthews Band	Listener Supported	USRC19901243
Roundabout	Yes	The Family Tree	FR6V80051479
Vernie	Blind Melon	Soup	USCA29500710
Lucy in the sky with Diamonds	The Beatles	Sgt. Pepper's Lonely Hearts Club Band	
Eye of the Beholder	Metallica	...And Justice For All	

Music	John Miles	Decca Singles 1975-79	
Bicycle Race (Digital Remaster)	Queen	The A-Z of Queen Vol. 1	GBCEE0100116
Dazed And Confused	Led Zeppelin	Led Zeppelin (Deluxe Edition)	USAT21300919
I Call Your Name (Album Version)	The Mamas & The Papas	Greatest Hits: The Mamas & The Papas	USMC16646370
I Me Mine	The Beatles	Let it Be	
Phantom of the Opera	Iron Maiden	Iron Maiden	GBAYE9801362
Back to Black	Amy Winehouse	Back to Black	
Four Sticks	Led Zeppelin	Led Zeppelin IV (Deluxe Edition)	USAT21300961
Midnight Rambler	The Rolling Stones	Let It Bleed	USA176910070
Oily Way - Original	Gong	The World Of Daevid Allen And Gong CD2	USFB20608652
Sorry (To Be Me)	Two Ton Shoe	Figures...	USHM80454265
I Want You (She's so heavy)	The Beatles	Abbey Road	
Cannabis (Album Version)	Nino Ferrer	Nino Ferrer	FRZ017200250
Dear Jessie	Madonna	Like A Prayer	USWB10002781
Hang on in There	John Legend & The Roots	Wake Up!	USSM11002248
The Mirror	Dream Theater	Awake	USEW29400071
Charlotte the Harlot	Iron Maiden	Iron Maiden	GBAYE9801365
Some Velvet Morning	Nancy Sinatra And Lee Hazlewood	Nancy & Lee	USASE0510172

(You're The) Devil In Disguise	Elvis Presley	The 50 Greatest Hits	USRC16305834
Some Velvet Morning	Lydia Lunch	Shotgun Wedding	GBBLY1300438
Words (Between The Lines Of Age)	Neil Young	Harvest	USRE10900207
Harry's House-Centerpiece	Joni Mitchell	Misses	USEE10170377
One Rainy Wish	The Jimi Hendrix Experience	Axis: Bold As Love	USQX90900762
All you Need is Love	The Beatles	Magical Mystery Tour	
Good Morning Good Morning	The Beatles	Sgt. Pepper's Lonely Hearts Club Band	
Uncle Albert / Admiral Halsey	Paul McCartney	RAM	
Armageddon Blues	Gary Willis	Bent	
Bye Bye Bye	Scott Bradley's Postmodern Jukdebox	PMJ and chill	
The continuing story of Bungalow Bill	The Beatles	the White Album	
Happiness is a warm Gun	The Beatles	the White Album	
Lie	Dream Theater	Awake	
La Malinche	Feu! Chatterton	Ici le jour (a tout enseveli)	
Songs of Yesterday	Free	Songs of Yesterday	
Grenade	Scott Bradley's Postmodern Jukdebox	Swing the Vote!	

Run to the Hills	Iron Maiden	The Number of the Beast	
Oops! ? did it again	Scott Bradley's Postmodern Jukdebox	Swipe right for vintage	
Smooth Criminal	Patax	Patax plays Michael (a tribute)	
They Don't Care About Us	Patax	Patax plays Michael (a tribute)	
Love is Stronger Than Justice	Sting	Ten Summoner's Tales	
The judge	Twenty One Pilots	Blurryface	

Appendix B

Descending PPK algorithm

The Descending counterpart of the ascending peak-picking kernel (PPK) described in chapter 4 is given in pseudo-code below.

Algorithm 4 Descending Peak-picking kernel: $\mathcal{K}'(\omega_j, \mathcal{M})$

Require: ω_j is the level under analysis and $\mathcal{M}$, the metrical structure candidates

while $\frac{\omega_j}{\omega_{j-1}} \notin \mathbb{N}$ **do**
    $\omega_{j-1} \leftarrow \omega_{j-2}$
$\omega_q \leftarrow \omega_{j-1}$
if $\frac{\omega_{q-1}}{\omega_j} \in \mathbb{N}$ **then**
    if $\frac{\omega_q}{\omega_{q-1}} \notin \mathbb{N}$ **then**
        $\mathcal{M}_1 \leftarrow \mathcal{M}$
        $\mathcal{M}_2 \leftarrow \mathcal{M}$
        $\omega_j \leftarrow \omega_q$
        $(\omega_j, \mathcal{M}_1) \leftarrow \mathcal{K}'(\omega_j, \mathcal{M}_1)$ ▷ call peak-picking kernel
        $\mathcal{M} \leftarrow \{\mathcal{M}, \mathcal{M}_1\}$
        $\omega_j \leftarrow \omega_{q-1}$
        $(\omega_j, \mathcal{M}_2) \leftarrow \mathcal{K}'(\omega_j, \mathcal{M}_2)$ ▷ call peak-picking kernel
        $\mathcal{M} \leftarrow \{\mathcal{M}, \mathcal{M}_2\}$
    else
        append ω_{j-1} to $\mathcal{M}$
        $\omega_j \leftarrow \omega_{j-1}$
else
    append ω_{j-1} to $\mathcal{M}$
    $\omega_j \leftarrow \omega_{j-1}$
return ω_j, $\mathcal{M}$

Appendix C

Derivation of update rules for L_β-S-β-NMF

C.1 Majorisation-Minimisation and β-divergence

Given a non-negative matrix $\mathbf{V} \in \mathbb{R}^{M \times N}$ NMF seeks to find $\mathbf{W} \in \mathbb{R}^{M \times K}$ and $\mathbf{H} \in \mathbb{R}^{K \times N}$ such that their product approximates $\mathbf{V}$:

$$\mathbf{V} \approx \mathbf{WH} \tag{C.1}$$

The β-divergence [242]:

$$\mathcal{C}_\beta^{(\beta)}(v|z) = \frac{v^\beta}{\beta(\beta-1)} + \frac{z^\beta}{\beta} + \frac{vz^{\beta-1}}{\beta-1} \tag{C.2}$$

is a well known generalised cost function for solving this task.

Majorisation-minimisation methods are employed to derive monotonic NMF algorithms [141, 243]. These are effected through the use of an auxiliary function $\mathcal{G}(h|\hat{h})$ which has the properties:

$$\mathcal{G}(h|\hat{h}) \geq \mathcal{C}(h) \tag{C.3}$$

$$\mathcal{G}(h|h) = \mathcal{C}(h) \tag{C.4}$$

where $\hat{h}$ is referred to as an auxiliary variable. In practice, $\hat{h}$ is the value of h at the current iteration. These properties guarantee that optimisation of the auxiliary function results in optimisation of the original function. An auxiliary function for the β-divergence is given in [141]. For values of $0 < \beta < 1$, the range of interest here, the β-divergence is composed of a convex and concave function, i.e. $\mathcal{C} = \breve{\mathcal{C}} + \widehat{\mathcal{C}}$, and it is shown that the auxiliary function is similarly composed [141, 243]:

$$\mathcal{G}(h|\hat{h}) = \breve{\mathcal{G}}(h|\hat{h}) + \widehat{\mathcal{G}}(h|\hat{h}) \tag{C.5}$$

It is also stated that the β-divergence auxiliary function is separable in each element i.e.

$$\mathcal{G}(\mathbf{H}|\hat{\mathbf{H}}) = \sum_{k,n} \mathcal{G}(h_{k,n}|\hat{h}_{k,n}) \tag{C.6}$$

As seen above (C.2), the third term of the β-divergence is convex, for which Jensen's inequality is used to derive an auxiliary function relative to $\mathbf{H}$:

$$\mathcal{G}(h_{k,n}|\hat{h}_{k,n}) = a^H_{k,n} h_{k,n} \quad \text{where} \quad a^H_{k,n} = \sum_m w_{m,k} v_{m,n} z^{\beta-2}_{m,n} \tag{C.7}$$

where $\mathbf{Z} = \mathbf{WH}$. Similarly, relative to $\mathbf{W}$:

$$\mathcal{G}(w_{m,k}|\hat{w}_{m,k}) = a^W_{m,k} w_{m,k} \quad \text{where} \quad a^W_{m,k} = \sum_n v_{m,n} z^{\beta-2}_{m,n} h_{k,n} \tag{C.8}$$

The second term in (C.2) is concave, and is majorised using Taylor expansion, which gives the auxiliary function relative to $\mathbf{H}$:

$$\mathcal{G}(h_{k,n}|\hat{h}_{k,n}) = \frac{b^H_{k,n}}{1-\beta} \hat{h}_{k,n} \left(\frac{h_{k,n}}{\hat{h}_{k,n}} \right)^{\beta-1} \quad \text{where} \quad b^H_{k,n} = \sum_m w_{m,k} z^{\beta-1}_{m,n} \tag{C.9}$$

and relative to $\mathbf{W}$:

$$\mathcal{G}(w_{m,k}|\hat{w}_{m,k}) = \frac{b^W_{m,k}}{1-\beta} \hat{w}_{m,k} \left(\frac{w_{m,k}}{\hat{w}_{m,k}} \right)^{\beta-1} \quad \text{where} \quad b^W_{m,k} = \sum_n z^{\beta-1}_{m,n} h_{k,n} \tag{C.10}$$

Meanwhile, the first term in (C.2) is constant in terms of $\mathbf{W}$ and $\mathbf{H}$ and can be ignored in the optimisation.

C.2 Auxiliary function for proposed penalty

Let us define the matrix $\mathbf{Y} \in \mathbb{R}^{K \times N}$:

$$[Y]_{k,n} = h_{k,n} \times \|\mathbf{w}_k\|_2 \tag{C.11}$$

The considered penalty is given as:

$$\mathcal{C}_p^{(\beta)}(\mathbf{Y}) = \frac{1}{\beta} \sum_n \|\mathbf{y_n}\|_\beta^\beta = \frac{1}{\beta} \sum_{k,n} y_{k,n}^\beta = \frac{1}{\beta} \sum_{k,n} h_{k,n}^\beta \times \|\mathbf{w}_k\|_2^\beta = \sum_{k,n} \mathcal{C}_p^{(\beta)}(y_{k,n}) \tag{C.12}$$

Two separate auxiliary functions need to be derived for this penalty, one with respect to $\mathbf{H}$ and one with respect to $\mathbf{W}$. For the activation update, consider that the dictionary is normalised and is not updated in the iteration; in which case:

$$\|\mathbf{w}_k\| = \|\hat{\mathbf{w}}_k\| = 1 \, \forall k \tag{C.13}$$

The element-wise cost function is then given as:

$$\mathcal{C}_p^{(\beta)}(y_{k,n}) = \frac{h_{k,n}^\beta}{\beta} \tag{C.14}$$

Since we only consider here cases where $0 < \beta < 1$, the above term is concave. Therefore, an auxiliary function for the penalty can be derived using the Taylor expansion :

$$\mathcal{G}_p^\beta(y_{k,n}|\hat{\mathbf{H}}) = h_{k,n} \hat{h}_{k,n}^{\beta-1} + cst \tag{C.15}$$

with gradient:

$$\frac{d\mathcal{G}_p^\beta(y_{k,n}|\hat{\mathbf{H}})}{dh_{k,n}} = \hat{h}_{k,n}^{\beta-1} \tag{C.16}$$

For the dictionary update, it is considered that the activations are constant i.e. $\mathbf{H} = \hat{\mathbf{H}}$ and the dictionary is normalised before the update, i.e. $\|\hat{\mathbf{w}}_\mathbf{k}\| = 1$. The cost function can then be written:

$$\mathcal{C}_p^{(\beta)}(\mathbf{Y}) = \frac{1}{\beta} \sum_k \left[\|\mathbf{w_k}\|_2^\beta \sum_n \hat{h}_{k,n}^\beta \right] \tag{C.17}$$

which is not separable relative to the individual dictionary elements $w_{m,k}$. In order to create a separable auxilary function, an alternative majorisation using the weighted arithmetic-geometric inequality, similar to [144] is employed:

$$(a^v + b^w)^{\frac{1}{v+w}} \leq \frac{va + wb}{v + w} \tag{C.18}$$

Setting $a = \|\mathbf{w}_k\|_2^2, b = \|\hat{\mathbf{w}}_k\|_2^2$, $v = \beta, w = 2 - \beta$, leads to:

$$\|\mathbf{w}_k\|_2^\beta \leq \frac{\beta}{2} \frac{\|\mathbf{w}_k\|_2^2}{\|\hat{\mathbf{w}}_k\|_2^{2-\beta}} + \left(1 - \frac{\beta}{2}\right) \|\hat{\mathbf{w}}_k\|_2^\beta = \frac{\beta}{2} \sum_m w_{m,k}^2 + cst \tag{C.19}$$

which leads to the separable auxiliary function, given $\|\mathbf{w}_k\|_2 = 1$:

$$\mathcal{G}_p^{(\beta)}(w_{m,k}|\hat{\mathbf{W}}) = \sum_{m,k} w_{m,k}^2 \sum_n \hat{h}_{k,n}^\beta = \sum_{m,k} \hat{w}_{m,k}^2 \left(\frac{w_{m,k}}{\hat{w}_{m,k}}\right)^2 \sum_n \hat{h}_{k,n}^\beta \tag{C.20}$$

C.3 Deriving the updates

The total penalised cost function is then given as:

$$\mathcal{C}_{S\beta}^{(\beta)}(\mathbf{V}|\mathbf{W}, \mathbf{H}) = \mathcal{C}_\beta^{(\beta)}(\mathbf{V}|\mathbf{Z}) + \lambda \mathcal{C}_p^{(\beta)}(\mathbf{Y}) \tag{C.21}$$

The auxiliary function is similarly composed. Relative to $\mathbf{H}$ this can be stated as:

$$\mathcal{G}(h_{k,n}|\hat{h}_{k,n}) = a_{k,n}^H h_{k,n} + \frac{1}{1-\beta} b_{k,n}^H \hat{h}_{k,n} \left(\frac{h_{k,n}}{\hat{h}_{k,n}}\right)^{\beta-1} + \lambda h_{k,n} \hat{h}_{k,n}^{\beta-1} \tag{C.22}$$

Taking the gradient and setting to zero:

$$\frac{d\mathcal{G}(h_{k,n}|\hat{h}_{k,n})}{dh_{k,n}} = a_{k,n}^H - b_{k,n}^H \left(\frac{h_{k,n}}{\hat{h}_{k,n}}\right)^{\beta-2} + \lambda \hat{h}_{k,n}^{\beta-1} = 0 \tag{C.23}$$

gives:

$$\frac{a_{k,n} + \lambda \hat{h}_{k,n}^{\beta-1}}{b_{k,n}} = \left(\frac{h_{k,n}}{\hat{h}_{k,n}}\right)^{\beta-2} \tag{C.24}$$

which is exponentiated by $\frac{1}{\beta-2}$ to give:

$$\frac{h_{k,n}}{\hat{h}_{k,n}} = \left[\frac{a_{k,n} + \lambda\hat{h}_{k,n}^{\beta-1}}{b_{k,n}}\right]^{\frac{1}{\beta-2}} = \left[\frac{b_{k,n}}{a_{k,n} + \lambda\hat{h}_{k,n}^{\beta-1}}\right]^{\frac{1}{2-\beta}} \tag{C.25}$$

which is written in matrix form as:

$$\mathbf{H} \longleftarrow \mathbf{H} \odot \left[\frac{\mathbf{W}^T[\mathbf{V} \odot [\mathbf{WH}]^{[\beta-2]}]}{\mathbf{W}^T[[\mathbf{WH}]^{[\beta-1]}] + \lambda\mathbf{H}^{[\beta-1]}}\right]^{\left[\frac{1}{2-\beta}\right]} \tag{C.26}$$

where $\odot$ denotes element-wise multiplication, $\mathbf{X}^{[.]}$ denotes element-wise exponentiation and the division is also element-wise.

In terms of $\mathbf{W}$, the auxiliary function can be stated as

$$\mathcal{G}(w_{m,k}|\hat{w}_{m,k}) = a_{m,k}^W w_{m,k} + \frac{b_{m,k}^W}{1-\beta}\hat{w}_{m,k}\left(\frac{w_{m,k}}{\hat{w}_{m,k}}\right)^{\beta-1} + \lambda\hat{w}_{m,k}^2\left(\frac{w_{m,k}}{\hat{w}_{m,k}}\right)^2\sum_n \hat{h}_{k,n}^\beta \tag{C.27}$$

Unlike the case of $\mathbf{H}$, a further majorisation step is required in order to make this optimisation more malleable. This follows the approach of [227], using Lemma 1 therein, whereby the first term of (C.27) is majorised in order to have a common multiplier as the third term. In particular, the first term can be rewritten as $a_{m,k}^W w_{m,k} = a_{m,k}^W \hat{w}_{m,k}\frac{w_{m,k}}{\hat{w}_{m,k}}$ to which the identity can be applied:

$$\frac{w_{m,k}}{\hat{w}_{m,k}} - 1 \leq \frac{1}{2}\left[\left(\frac{w_{m,k}}{\hat{w}_{m,k}}\right)^2 - 1\right] \tag{C.28}$$

giving the majorisation:

$$a_{m,k}^W w_{m,k} \leq a_{m,k}^W \hat{w}_{m,k}\left(\frac{w_{m,k}}{\hat{w}_{m,k}}\right)^2 + cst. \tag{C.29}$$

The term on the right of (C.29) can then be substituted into (C.27), before taking the gradient and setting to zero in a similar fashion to taken and set to zero similar to (C.23) above, leading to:

$$a_{m,k}^W\left(\frac{w_{m,k}}{\hat{w}_{m,k}}\right) + \lambda\hat{w}_{m,k}\left(\frac{w_{m,k}}{\hat{w}_{m,k}}\right)\sum_n \hat{h}_{k,n}^\beta = b_{m,k}^W\hat{w}_{m,k}\left(\frac{w_{m,k}}{\hat{w}_{m,k}}\right)^{\beta-2} \tag{C.30}$$

Setting $[\Psi]_{m,k} = \sum_n \hat{h}_{k,n}^{\beta}$ and dividing both sides by $\left(\frac{w_{m,k}}{\hat{w}_{m,k}}\right)$ gives:

$$a_{m,k}^{W} + \lambda \hat{w}_{m,k}[\Psi]_{m,k} = b_{m,k}^{W} \hat{w}_{m,k} \left(\frac{w_{m,k}}{\hat{w}_{m,k}}\right)^{\beta-3} \tag{C.31}$$

which can be manipulated similar to (C.24), (C.25), (C.26) leading to the multiplicative update:

$$\mathbf{W} \longleftarrow \mathbf{W} \odot \left[\frac{[\mathbf{V} \odot [\mathbf{WH}]^{[\beta-2]}]\mathbf{H}^T}{[[\mathbf{WH}]^{[\beta-2]}]\mathbf{H}^T + \lambda[\mathbf{W} \odot \Psi]}\right]^{\left[\frac{1}{3-\beta}\right]} \tag{C.32}$$

Bibliography

[1] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 13(5):1035–1047, 2005. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1495485`.

[2] Ugo Marchand, Quentin Fresnel, and Geoffroy Peeters. GTZAN_rhythm: Extending the GTZAN test-set with beat, downbeat and swing annotations. In *International Society for Music Information Retrieval (ISMIR) Conference - Late breaking session*, 2015.

[3] Andre Holzapfel, Matthew E.P. Davies, José R. Zapata, João Lobato Oliveira, and Fabien Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, 2012. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6220849`.

[4] Michael A. Casey, Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4472077`.

[5] J. Stephen Downie. Music information retrieval. *Annual review of information science and technology*, 37(1):295–340, 2003. URL `http://onlinelibrary.wiley.com/doi/10.1002/aris.1440370108/full`.

[6] Xavier Serra, Michela Magas, Emmanouil Benetos, Magdalena Chudy, Simon Dixon, Arthur Flexer, Emilia Gómez, Fabien Gouyon, Perfecto Herrera, Sergi

Jorda, and others. *Roadmap for music information research*. MIRES Consortium, 2013. URL http://openaccess.city.ac.uk/2763/1/MIRES_Roadmap_ver_1.0.0.pdf.

[7] Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer, 2015. URL https://books.google.com/books?hl=fr&lr=&id=HCI_CgAAQBAJ&oi=fnd&pg=PR7&dq=Fundamentals+of+Music+Processing+%E2%80%93+Audio,+Analysis,+Algorithms,+Applications&ots=Vn_mNlp7eQ&sig=pBUT760AiZ3UZL1KkYWlmyrOmvc.

[8] Norberto Degara, Antonio Pena, Matthew EP Davies, and Mark D. Plumbley. Note onset detection using rhythmic structure. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 5526–5529, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5495220.

[9] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2015. URL https://pdfs.semanticscholar.org/7dc4/7c8971e8d6126dd7a94cb7fd15e5a035702a.pdf.

[10] Anders Elowsson and Anders Friberg. Modeling the perception of tempo. *The Journal of the Acoustical Society of America*, 137(6):3163–3177, 2015. URL http://scitation.aip.org/content/asa/journal/jasa/137/6/10.1121/1.4919306.

[11] Masataka Goto and Yoichi Muraoka. A real-time beat tracking system for audio signals. In *Proceedings of the international computer music conference*, pages 171–174. San Francisco: International Computer Music Association, 1995. URL https://staff.aist.go.jp/m.goto/PAPER/ICMC95goto.pdf?origin=publication_detail.

[12] Simon Dixon and Emilios Cambouropoulos. Beat tracking with musical knowledge. In *ECAI*, pages 626–630, 2000. URL http://users.auth.gr/~emilios/papers/ecai2000.pdf.

[13] Tristan Jehan. Downbeat prediction by listening and learning. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 267–270, 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1540221.

[14] Maksim Khadkevich, Thomas Fillon, Gaël Richard, and Maurizio Omologo. A probabilistic approach to simultaneous extraction of beats and downbeats. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 445–448, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6287912.

[15] Anssi P. Klapuri, Antti J. Eronen, and Jaakko T. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1561290.

[16] Geoffroy Peeters and Helene 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):1754–1769, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5664773.

[17] Matthew EP Davies, Philippe Hamel, Kazutomo Yoshii, and Misako Goto. AutoMashUpper: automatic creation of multi-song music mashups. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1726–1737, 2014. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6876193.

[18] Geraint A. Wiggins, Daniel Müllensiefen, and Marcus T. Pearce. On the non-existence of music: Why music theory is a figment of the imagination. *Musicae Scientiae*, 14(1_suppl):231–255, 2010. URL http://journals.sagepub.com/doi/abs/10.1177/10298649100140S110.

[19] Justin London. *Hearing in time*. Oxford University Press, 2012. URL https://books.google.com/books?hl=fr&lr=&id=XZ92CpoLlVoC&oi=fnd&pg=PP2&dq=hearing+in+time&ots=H4H5qLRb6C&sig=4znSss-WMagQ_RVD-kiSm3JK5Sw.

[20] Caroline Palmer and Carol L. Krumhansl. Mental representations for musical meter. *Journal of Experimental Psychology: Human Perception and Performance*, 16(4):728, 1990. URL http://psycnet.apa.org/journals/xhp/16/4/728/.

[21] M. McKinney and Dirk Moelants. Deviations from the resonance theory of tempo induction. *Abstracts of the Conference on Interdisciplinary Musicology*, pages 124–125, 2004. URL http://hdl.handle.net/1854/LU-218105.

[22] Marcus T. Pearce and Andrea R. Halpern. Age-related patterns in emotions evoked by music. *Psychology of Aesthetics, Creativity, and the Arts*, 9(3):248, 2015. URL http://psycnet.apa.org/journals/aca/9/3/248/.

[23] Andrea R. Halpern, Ioanna Zioga, Martin Shankleman, Job Lindsen, Marcus T. Pearce, and Joydeep Bhattarcharya. That note sounds wrong! Age-related effects in processing of musical expectation. *Brain and cognition*, 113:1–9, 2017. URL http://www.sciencedirect.com/science/article/pii/S0278262616300434.

[24] Peter Desain. A (de) composable theory of rhythm perception. *Music Perception: An Interdisciplinary Journal*, 9(4):439–454, 1992. URL http://mp.ucpress.edu/content/9/4/439.abstract.

[25] Bastiaan van der Weij, Marcus Pearce, and Henkjan Honing. A probabilistic model of meter perception: Simulating enculturation. *Frontiers in Psychology*, 8:824, 2017. URL http://journal.frontiersin.org/article/10.3389/fpsyg.2017.00824.

[26] Erin E. Hannon, Joel S. Snyder, Tuomas Eerola, and Carol L. Krumhansl. The role of melodic and temporal cues in perceiving musical meter. *Journal of Experimental Psychology: Human Perception and Performance*, 30(5):956, 2004. URL http://psycnet.apa.org/journals/xhp/30/5/956/.

[27] Nick Whiteley, Ali Taylan Cemgil, and Simon J. Godsill. Bayesian Modelling of Temporal Structure in Musical Audio. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 29–34, 2006. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.113.1354&rep=rep1&type=pdf.

[28] Florian Krebs, Andre Holzapfel, Ali Taylan Cemgil, and Gerhard Widmer. Inferring metrical structure in music using particle filters. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(5):817–827, 2015. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7055854`.

[29] Elio Quinton, Christopher Harte, and Mark Sandler. Extraction of Metrical Structure from Music Recordings. In *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx)*, 2015.

[30] Elio Quinton, Mark Sandler, and Simon Dixon. Estimation of the reliability of multiple rhythm features extraction from a single descriptor. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 256–260. IEEE, 2016. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7471676`.

[31] Elio Quinton, Ken O'Hanlon, Simon Dixon, and Mark B. Sandler. Tracking Metrical Structure Changes with Sparse-NMF. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[32] Elio Quinton, Christopher Harte, and Mark Sandler. Audio tempo estimation using fusion of time-frequency analyses and metrical structure. In *Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2014.

[33] Elio Quinton, Ken O'Hanlon, Simon Dixon, and Mark Sandler. Automatic Detection of Metrical Structure Changes. In *Digital Music Research Network 1-Day Workshop (DMRN+11)*, 2016.

[34] Barış Bozkurt, Ruhi Ayangil, and Andre Holzapfel. Computational analysis of turkish makam music: Review of state-of-the-art and challenges. *Journal of New Music Research*, 43(1):3–23, 2014. URL `http://www.tandfonline.com/doi/abs/10.1080/09298215.2013.865760`.

[35] Andre Holzapfel, Florian Krebs, and Ajay Srinivasamurthy. Tracking the "Odd": Meter Inference in a Culturally Diverse Music Corpus. In *International Society for Music Information Retrieval (ISMIR) Conference*,

pages 425–430, 2014. URL https://pdfs.semanticscholar.org/3127/8daefc5c82b661d275eff87319930d9b1bd0.pdf.

[36] Martin Scherzinger. Temporal geometries of an African music: A preliminary sketch. *Music Theory Online*, 16(4), 2010. URL http://www.visualculturenow.org/wp-content/uploads/Temporal-Geometries-of-an-African-Music.pdf.

[37] Leonardo Nunes, Martın Rocamora, Luis Jure, and Luiz WP Biscainho. Beat and downbeat tracking based on rhythmic patterns applied to the Uruguayan Candombe drumming. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2015. URL https://www.researchgate.net/profile/Martin_Rocamora/publication/282859198_BEAT_AND_DOWNBEAT_TRACKING_BASED_ON_RHYTHMIC_PATTERNS_APPLIED_TO_THE_URUGUAYAN_CANDOMBE_DRUMMING/links/561fa76e08ae93a5c924190f.pdf.

[38] Godfried T. Toussaint. Quantifying Musical Meter: How Similar are African and Western Rhythm? *Analytical Approaches to World Music*, 4(1), 2015. URL http://www.aawmjournal.com/articles/2015a/Toussaint_AAWM_Vol_4_1.pdf.

[39] J. Seppanen. Tatum grid analysis of musical signals. In *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, pages 131–134, 2001. doi: 10.1109/ASPAA.2001.969560.

[40] Jarno Seppänen, Antti J. Eronen, and Jarmo Hiipakka. Joint Beat & Tatum Tracking from Music Signals. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 23–28, 2006. URL http://www.ismir.net/proceedings/index.php?table_name=papers&function=details&where_field=Id&where_value=410.

[41] Fred Lerdahl and Ray S. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983. ISBN 978-0-262-62107-6.

[42] Michael Kennedy, Tim Rutherford-Johnson, and Joyce Kennedy. *The Oxford dictionary of music*. Oxford University Press, 2013. URL https://books.google.com/books?hl=fr&lr=&id=XX2sAQAAQBAJ&oi=fnd&pg=PP2&dq=The+Oxford+Dictionary+of+Music.&ots=V2is3AK7-F&sig=yN7t23PIo8Q4qtDgCL7K_Uret2Y.

[43] Matthias Robine, Pierre Hanna, and Mathieu Lagrange. Meter Class Profiles for Music Similarity and Retrieval. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 639–644, 2009. URL `http://www.ismir2009.ismir.net/proceedings/PS4-11.pdf`.

[44] Justin London. Tactus /= Tempo: Some Dissociations Between Attentional Focus, Motor Behavior, and Tempo Judgment. *Empirical Musicology Review*, 6(1):43–55, January 2011. ISSN 1559-5749. URL `http://hdl.handle.net/1811/49761`.

[45] Martin F. McKinney and Dirk Moelants. Ambiguity in tempo perception: What draws listeners to different metrical levels? *Music Perception: An Interdisciplinary Journal*, pages 155–166, 2006. URL `http://www.jstor.org/stable/10.1525/mp.2006.24.2.155`.

[46] Anders Elowsson, Anders Friberg, Guy Madison, and Johan Paulin. Modelling the speed of music using features from harmonic/percussive separated audio. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2013. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.390.9118`.

[47] Anders Elowsson and Anders Friberg. Modelling perception of speed in music audio. *Forthcoming for Proc. of SMC*, 2013. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.416.7300&rep=rep1&type=pdf`.

[48] Guy Madison and Johan Paulin. Ratings of speed in real music as a function of both original and manipulated beat tempo. *The Journal of the Acoustical Society of America*, 128(5):3032–3040, 2010. URL `http://scitation.aip.org/content/asa/journal/jasa/128/5/10.1121/1.3493462`.

[49] Simon Dixon. Onset detection revisited. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, pages 133–137, 2006. URL `http://138.37.35.209/people/simond/pub/2006/dafx.pdf`.

[50] Brian CJ Moore. *An introduction to the psychology of hearing*. Brill, 2012. URL `https://books.google.com/books?hl=fr&lr=&id=LM9U8e28pLMC&oi=fnd&pg=PP1&dq=An+Introduction+to+the+Psychology+of+Hearing&ots=L1Xqh3MKC8&sig=8fsBAvnUCxKAUMq4JVy0a0u6luA`.

[51] Masataka Goto and Yoichi Muraoka. Beat tracking based on multiple-agent architecture—a real-time beat tracking system for audio signals. In *Proceedings of the Second International Conference on Multiagent Systems*, pages 103–110, 1996. URL `http://www.aaai.org/Papers/ICMAS/1996/ICMAS96-013.pdf`.

[52] Eric D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103:588, 1998. URL `http://link.aip.org/link/?JASMAN/103/588/1`.

[53] Anssi Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 6, pages 3089–3092, 1999. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=757494`.

[54] Chris Duxbury, Mark Sandler, and Mike Davies. A hybrid approach to musical note onset detection. In *Proc. Digital Audio Effects Conf.(DAFX,'02)*, pages 33–38, 2002. URL `http://www.unibw-hamburg.de/EWEB/ANT/dafx2002/papers/DAFX02_Duxbury_Sandler_Davis_note_onset_detection.pdf`.

[55] Nick Collins. A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions. In *Audio Engineering Society Convention 118*. Audio Engineering Society, 2005. URL `http://www.aes.org/e-lib/browse.cfm?conv=118&papernum=6363`.

[56] Walter Andrew Schloss. *On the automatic transcription of percussive music: from acoustic signal to high-level analysis*. PhD thesis, Stanford University, 1985.

[57] Chris Chafe and David Jaffe. Source separation and note identification in polyphonic music. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 11, pages 1289–1292, 1986. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1168727`.

[58] Simon Dixon. A beat tracking system for audio signals. In *Proceedings of the Conference on Mathematical and Computational Methods in Music*, pages 101–110. Citeseer, 1999. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.1409&rep=rep1&type=pdf`.

[59] Brian CJ Moore, Brian R. Glasberg, and Thomas Baer. A model for the prediction of thresholds, loudness, and partial loudness. *Journal of the Audio Engineering Society*, 45(4):224–240, 1997. URL http://www.aes.org/e-lib/browse.cfm?elib=10272.

[60] Xavier Rodet and Florent Jaillet. Detection and modeling of fast attack transients. In *Proceedings of the International Computer Music Conference*, pages 30–33, 2001. URL http://articles.ircam.fr/textes/Rodet01a/index.pdf.

[61] Paul Masri. *Computer modelling of sound for transformation and synthesis of musical signals.* PhD thesis, University of Bristol, 1996. URL http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.246275.

[62] Sebastian Böck and Gerhard Widmer. Maximum filter vibrato suppression for onset detection. In *Proc. of the 16th Int. Conf. on Digital Audio Effects (DAFx)*, 2013. URL http://dafx13.nuim.ie/papers/09.dafx2013_submission_12.pdf.

[63] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *IEEE International Conference on Multimedia and Expo (ICME)*, volume 1, pages 452–455. IEEE, 2000. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=869637.

[64] Chris Duxbury, Juan Pablo Bello, Mike Davies, and Mark Sandler. A combined phase and amplitude based approach to onset detection for audio segmentation. In *Proc. 4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS-03)*, pages 275–280, 2003. URL https://www.researchgate.net/profile/Mark_Sandler2/publication/237304335_A_COMBINED_PHASE_AND_AMPLITUDE_BASED_APPROACH_TO_ONSET_DETECTION_FOR_AUDIO_SEGMENTATION/links/54118dce0cf264cee28b3f67.pdf.

[65] Juan Pablo Bello and Mark Sandler. Phase-based note onset detection for music signals. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages V–441, 2003. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1200001.

[66] Juan P. Bello, Chris Duxbury, Mike Davies, and Mark Sandler. On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, 11(6):553–556, 2004. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1300607`.

[67] Tristan Jehan. Musical signal parameter estimation. *CNMAT report*, 1997.

[68] Harvey Thornburg and Fabien Gouyon. A flexible analysis-synthesis method for transients. In *Int. Computer Music Conference (ICMC)*, pages 400–403, 2000. URL `https://pdfs.semanticscholar.org/e2a8/0bf1fc5fd17a688c8d74ec3721ae485c7ffd.pdf`.

[69] Samer A. Abdallah and Mark D. Plumbley. Probability as metadata: event detection in music using ICA as a conditional density model. In *Proc. 4th Int. Symp. Independent Component Analysis and Signal Separation (ICA2003)*, pages 233–238, 2003. URL `http://www.researchgate.net/profile/Samer_Abdallah/publication/2881085_Probability_As_Metadata_Event_Detection_In_Music_Using_ICA_as_a_Conditional_Density_Model/links/02e7e52319418a9569000000.pdf`.

[70] Matija Marolt, Alenka Kavcic, and Marko Privosnik. Neural networks for note onset detection in piano music. In *Proceedings of the 2002 International Computer Music Conference*, 2002.

[71] Alexandre Lacoste and Douglas Eck. A supervised classification algorithm for note onset detection. *EURASIP Journal on Applied Signal Processing*, 2007(1):153–153, 2007. URL `http://dl.acm.org/citation.cfm?id=1289114`.

[72] Florian Eyben, Sebastian Böck, Björn Schuller, and Alex Graves. Universal Onset Detection with Bidirectional Long Short-Term Memory Neural Networks. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 589–594, 2010. URL `http://www.ismir2010.ismir.net/proceedings/ismir2010-101.pdf`.

[73] Sebastian Böck, Andreas Arzt, Florian Krebs, and Markus Schedl. Online real-time onset detection with recurrent neural networks. In *Proceedings of the 15th*

International Conference on Digital Audio Effects (DAFx-12), York, UK, 2012. URL http://www.cp.jku.at/research/papers/Boeck_etal_DAFx_2012.pdf.

[74] Jan Schlüter and Sebastian Böck. Musical onset detection with convolutional neural networks. In *6th International Workshop on Machine Learning and Music (MML), Prague, Czech Republic*, 2013. URL http://phenicx.upf.edu/system/files/publications/Schlueter_MML13.pdf.

[75] Jan Schlüter and Sebastian Böck. Improved musical onset detection with convolutional neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6979–6983. IEEE, 2014. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6854953.

[76] Mi Tian, György Fazekas, Dawn AA Black, and Mark Sandler. Desing and Evaluation of Onset Detectors Using Different Fusion Policies. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2014. URL http://www.terasoft.com.tw/conf/ismir2014/proceedings/T114_229_Paper.pdf.

[77] Thomas L. Blum, Douglas F. Keislar, James A. Wheaton, and Erling H. Wold. Method and article of manufacture for content-based analysis, storage, retrieval, and segmentation of audio information, June 1999. URL https://www.google.com/patents/US5918223. US Patent 5,918,223.

[78] Ali Taylan Cemgil, Bert Kappen, Peter Desain, and Henkjan Honing. On tempo tracking: Tempogram Representation and Kalman filtering. *Journal of New Music Research*, 29(4):259–273, 2000. ISSN 0929-8215. doi: 10.1080/09298210008565462. URL http://www.tandfonline.com/doi/abs/10.1080/09298210008565462.

[79] Peter Grosche and Meinard Muller. Tempogram toolbox: Matlab implementations for tempo and pulse analysis of music recordings. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2011. URL http://people.mpi-inf.mpg.de/~pgrosche/publications/2011_GroscheMueller_TempogramToolbox_ISMIR-LateBreaking.pdf.

[80] Ajay Srinivasamurthy and Xavier Serra. A Supervised Approach to Hierarchical Metrical Cycle Tracking from Audio Music Recordings. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, April 2014. URL `http://mtg.upf.edu/system/files/publications/talaTrack.pdf`.

[81] Jouni Paulus and Anssi Klapuri. Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1159–1170, 2009. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5109767`.

[82] Kristoffer Jensen. Multiple scale music segmentation using rhythm, timbre, and harmony. *EURASIP Journal on Applied Signal Processing*, (1):159–159, 2007. URL `http://dl.acm.org/citation.cfm?id=1289120`.

[83] Jonathan Foote and Shingo Uchihashi. The Beat Spectrum: A New Approach To Rhythm Analysis. In *ICME*, 2001.

[84] Geoffroy Peeters. Rhythm Classification Using Spectral Rhythm Patterns. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 644–647, 2005. URL `http://recherche.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2005_ISMIR_RhythmClassification.pdf`.

[85] George Tzanetakis, Georg Essl, and Perry Cook. Human perception and computer extraction of musical beat strength. In *Proc. Digital Audio Effects Conf.(DAFX)*, volume 2, 2002. URL `http://www.cs.cmu.edu/~./gtzan/work/pubs/dafx02gtzan.pdf`.

[86] Judith C. Brown. Determination of the meter of musical scores by autocorrelation. *The Journal of the Acoustical Society of America*, 94(4):1953–1957, 1993. URL `http://scitation.aip.org/content/asa/journal/jasa/94/4/10.1121/1.407518`.

[87] Eric D. Scheirer. Pulse tracking with a pitch tracker. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, page 4, 1997. URL `http://pubs.media.mit.edu/pubs/papers/mohonk97.pdf`.

[88] Masataka Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001. URL http://www.tandfonline.com/doi/abs/10.1076/jnmr.30.2.159.7114.

[89] Miguel Alonso, Bertrand David, and Gaël Richard. A study of tempo tracking algorithms from polyphonic music signals. In *Proceedings of the 4th. COST 276 Workshop*, 2003. URL http://www.tsi.enst.fr/~grichard/Publications/cost03.pdf.

[90] Fabien Gouyon and Perfecto Herrera. Determination of the meter of musical audio signals: Seeking recurrences in beat segment descriptors. In *Audio Engineering Society Convention 114*, 2003. URL http://www.aes.org/e-lib/browse.cfm?elib=12583.

[91] F. Gouyon and P. Herrera. A beat induction method for musical audio signals. In *Proceedings of the Fourth European Workshop on Image Analysis for Multimedia Interactive Services*, pages 281–287, 2003. URL http://books.google.com/books?hl=fr&lr=&id=vVvJINURimIC&oi=fnd&pg=PA281&dq=computational+models+of+beat+induction&ots=ElYEB0qmzL&sig=xlWHipEzLVZAlnsEEOf4jklETL4.

[92] Simon Dixon, Elias Pampalk, and Gerhard Widmer. Classification of dance music by periodicity patterns. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2003. URL https://jscholarship.library.jhu.edu/handle/1774.2/23.

[93] Geoffroy Peeters. Time variable tempo detection and beat marking. In *Proceedings of the ICMC*, 2005. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.380.7179&rep=rep1&type=pdf.

[94] Edward W. Large, Felix V. Almonte, and Marc J. Velasco. A canonical model for gradient frequency neural networks. *Physica D: Nonlinear Phenomena*, 239 (12):905–911, 2010. URL http://www.sciencedirect.com/science/article/pii/S0167278910000187.

[95] Andrew J. Lambert, Tillman Weyde, and Newton Armstrong. Adaptive Frequency Neural Networks For Dynamic Pulse And Metre Perception. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2016. URL `http://m.mr-pc.org/ismir16/website/articles/228_Paper.pdf`.

[96] Florian Krebs, Sebastian Böck, and Gerhard Widmer. An Efficient State Space Model for Joint Tempo and Meter Tracking. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2015. URL `https://pdfs.semanticscholar.org/0a56/8dd1faa5fb82cf2d27d59c8251b309974575.pdf`.

[97] Jason Hockman, Matthew EP Davies, and Ichiro Fujinaga. One in the Jungle: Downbeat Detection in Hardcore, Jungle, and Drum and Bass. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 169–174, 2012. URL `https://pdfs.semanticscholar.org/c255/6d7f08f46b8e1d517316ab3f583918707dc4.pdf`.

[98] Simon Durand, Bertrand David, and Gaël Richard. Enhancing downbeat detection when facing different music styles. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3132–3136. IEEE, 2014. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6854177`.

[99] Simon Durand, Juan Pablo Bello, Bertrand David, and Gaël Richard. Robust Downbeat Tracking Using an Ensemble of Convolutional Networks. *arXiv preprint arXiv:1605.08396*, 2016. URL `http://arxiv.org/abs/1605.08396`.

[100] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 6964–6968, 2014. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6854950`.

[101] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. An End-to-End Neural Network for Polyphonic Piano Music Transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, 2016. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7416164`.

[102] Florian Krebs, Sebastian Böck, Matthias Dorfer, and Gerhard Widmer. Downbeat Tracking Using Beat-Synchronous Features And Recurrent Neural Networks. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2016. URL `http://m.mr-pc.org/ismir16/website/articles/249_Paper.pdf`.

[103] Florian Krebs, Sebastian Böck, and Gerhard Widmer. Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 227–232, 2013. URL `http://ismir2013.ismir.net/wp-content/uploads/2013/09/51_Paper.pdf`.

[104] Norberto Degara, Enrique Argones Rúa, Antonio Pena, Soledad Torres-Guijarro, Matthew EP Davies, and Mark D. Plumbley. Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):290–301, 2012. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5934584`.

[105] Simon Dixon, Fabien Gouyon, and Gerhard Widmer. Towards Characterisation of Music via Rhythmic Patterns. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2004. URL `http://www.iua.upf.edu/mtg/ismir2004/review/CRFILES/paper165-b28308914f720be8d4c5f00bf2a5c9aa.pdf`.

[106] Ajay Srinivasamurthy, Andre Holzapfel, Ali Taylan Cemgil, and Xavier Serra. A generalized Bayesian model for tracking long metrical cycles in acoustic music signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 76–80, 2016. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7471640`.

[107] Andre Holzapfel and Yannis Stylianou. Scale transform in rhythmic similarity of music. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):176–185, 2011. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5430891`.

[108] Christian Uhle and Juergen Herre. Estimation of tempo, micro time and time signature from percussive music. In *Proc. Int. Conference on Digital Audio Effects*

(DAFx), 2003. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.58.1897&rep=rep1&type=pdf.

[109] Nick Collins. Towards a style-specific basis for computational beat tracking. In *Proceedings of the 9th International Conference on Music Perception & Cognition (ICMPC)*, 2006. URL http://sro.sussex.ac.uk/1287/.

[110] Nicholas M. Collins. *Towards autonomous agents for live computer music: Realtime machine listening and interactive music systems.* PhD thesis, University of Cambridge, 2007. URL https://pdfs.semanticscholar.org/71c2/2ed2ea650a587e62bdea18eb5bbe0171f5ce.pdf.

[111] Nick Whiteley, A. Taylan Cemgil, and Simon Godsill. Sequential inference of rhythmic structure in musical audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–1321, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4218352.

[112] Simon Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58, 2001. URL http://www.tandfonline.com/doi/abs/10.1076/jnmr.30.1.39.7119.

[113] Simon Dixon. Evaluation of the audio beat tracking system beatroot. *Journal of New Music Research*, 36(1):39–50, 2007. URL http://www.tandfonline.com/doi/abs/10.1080/09298210701653310.

[114] Fabien Gouyon, Simon Dixon, and Gerhard Widmer. Evaluating low-level features for beat classification and tracking. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–1309, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4218349.

[115] Norberto Degara, Matthew EP Davies, Antonio Pena, and Mark D. Plumbley. Onset event decoding exploiting the rhythmic structure of polyphonic music. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1228–1239, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5771974.

[116] Peter Grosche and Meinard Müller. A Mid-Level Representation for Capturing Dominant Tempo and Pulse Information in Music Recordings. In *International*

Society for Music Information Retrieval (ISMIR) Conference, pages 189–194, 2009. URL http://domino.mpi-inf.mpg.de/intranet/ag4/ag4publ.nsf/4e77efd5c6e2ceadc12567530068624d/ab9f9a4b02b329dac125767b002f4c37/$FILE/2009_GroscheMueller_Tempogram_ISMIR.pdf.

[117] P. Grosche and M. Müller. Computing predominant local periodicity information in music recordings. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 33–36, 2009. doi: 10.1109/ASPAA.2009.5346544.

[118] Geoffroy Peeters. Template-based estimation of time-varying tempo. *EURASIP Journal on Applied Signal Processing*, 2007(1):158–158, 2007. URL http://dl.acm.org/citation.cfm?id=1289119.

[119] Fu-Hai Frank Wu, Tsung-Chi Lee, Jyh-Shing Roger Jang, Kaichun K. Chang, Chun-Hung Lu, and Wen-Nan Wang. A Two-Fold Dynamic Programming Approach to Beat Tracking for Audio Music with Time-Varying Tempo. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 191–196, 2011. URL http://www.mirlab.org/conference_papers/International_Conference/ISMIR%202011/papers/PS2-4.pdf.

[120] F. Gouyon, A. 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):1832–1844, 2006. ISSN 1558-7916. doi: 10.1109/TSA.2005.858509.

[121] Matthew EP Davies and Mark D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1009–1020, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4100674.

[122] Daniel PW Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007. URL http://www.tandfonline.com/doi/abs/10.1080/09298210701653344.

[123] Sebastian Böck and Markus Schedl. Enhanced beat tracking with context-aware neural networks. In *Proc. Int. Conf. Digital Audio Effects*, 2011. URL `http://recherche.ircam.fr/pub/dafx11/Papers/31_e.pdf`.

[124] Sebastian Böck, Florian Krebs, and Gerhard Widmer. A Multi-model Approach to Beat Tracking Considering Heterogeneous Music Styles. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 603–608, 2014. URL `https://www.researchgate.net/profile/Sebastian_Boeck/publication/266557018_A_MULTI-MODEL_APPROACH_TO_BEAT_TRACKING_CONSIDERING_HETEROGENEOUS_MUSIC_STYLES/links/543407430cf2dc341daf2e61.pdf`.

[125] Aggelos Gkiokas, Vassilis Katsouros, George Carayannis, and Themos Stajylakis. Music tempo estimation and beat tracking by applying source separation and metrical relations. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 421–424, 2012. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6287906`.

[126] Graham Percival and George Tzanetakis. Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1765–1776, 2014. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6879451`.

[127] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. URL `http://www.nature.com/nature/journal/v401/n6755/abs/401788a0.html`.

[128] Ruairí de Fréin, Konstantinos Drakakis, Scott Rickard, and Andrzej Cichocki. Analysis of financial data using non-negative matrix factorization. In *International Mathematical Forum*, volume 3, pages 1853–1870, 2008. URL `http://repository.wit.ie/2742/`.

[129] David Guillamet, Bernt Schiele, and Jordi Vitria. Analyzing non-negative matrix factorization for image classification. In *16th International Conference on Pattern Recognition*, volume 2, pages 116–119. IEEE, 2002. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1048251`.

[130] Yuan Gao and George Church. Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, 21(21):3970–3975, 2005. URL `http://bioinformatics.oxfordjournals.org/content/21/21/3970.short`.

[131] Marko Helen and Tuomas Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *13th European Signal Processing Conference*, pages 1–4. IEEE, 2005. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7078147`.

[132] Sebastian Ewert and Meinard Müller. Score-informed source separation for music signals. *Dagstuhl Follow-Ups*, 3, 2012. URL `http://drops.dagstuhl.de/opus/volltexte/2012/3467/`.

[133] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013. URL `http://link.springer.com/article/10.1007/s10844-013-0258-3`.

[134] Emmanuel Vincent, Nancy Bertin, and Roland Badeau. Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 109–112, 2008. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4517558`.

[135] Jouni Paulus and Tuomas Virtanen. Drum transcription with non-negative spectrogram factorisation. In *European Signal Processing Conference*, pages 1–4. IEEE, 2005. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7078139`.

[136] Paris Smaragdis and Judith C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 177–180, 2003. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1285860`.

[137] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001. URL http://papers.nips.cc/paper/1861-alg.

[138] Daniel D. Lee and H. Sebastian Seung. Unsupervised learning by convex and conic coding. *Advances in neural information processing systems*, pages 515–521, 1997. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.6629&rep=rep1&type=pdf.

[139] Raul Kompass. A generalized divergence measure for nonnegative matrix factorization. *Neural computation*, 19(3):780–791, 2007. URL http://www.mitpressjournals.org/doi/abs/10.1162/neco.2007.19.3.780.

[140] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. In *International Conference on Independent Component Analysis and Signal Separation*, pages 32–39. Springer, 2006. URL http://link.springer.com/chapter/10.1007/11679363_5.

[141] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural computation*, 23(9):2421–2456, 2011. URL http://www.mitpressjournals.org/doi/abs/10.1162/NECO_a_00168.

[142] Julian Eggert and Edgar Korner. Sparse coding and NMF. In *Proceedings of the International Joint Conference on Neural Networks*, volume 4, pages 2529–2533. IEEE, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1381036.

[143] Steven K. Tjoa and KJ Ray Liu. Multiplicative update rules for nonnegative matrix factorization with co-occurrence constraints. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 449–452, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5495734.

[144] Ken O'Hanlon, Hidehisa Nagano, Nicolas Keriven, and Mark D. Plumbley. Non-negative group sparsity with subspace note modelling for polyphonic transcription.

IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24(3):530–542, 2016. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7384716.

[145] Stuart Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1056489.

[146] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007. URL http://dl.acm.org/citation.cfm?id=128338.

[147] Derry Fitzgerald. Harmonic/percussive separation using median filtering. In *13th International Conference on Digital Audio Effects (DAFX10)*, 2010. URL http://arrow.dit.ie/argcon/67/.

[148] Lawrence Rabiner and B. Juang. An introduction to hidden Markov models. *IEEE assp magazine*, 3(1):4–16, 1986. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1165342.

[149] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=18626.

[150] Taemin Cho and Juan P. Bello. On the relative importance of individual components of chord recognition systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2):477–492, 2014. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6691936.

[151] Alexander Sheh and Daniel PW Ellis. Chord segmentation and recognition using EM-trained hidden markov models. In *International Society for Music Information Retrieval (ISMIR) Conference*, volume 3, pages 183–189, 2003. URL https://jscholarship.library.jhu.edu/handle/1774.2/26.

[152] Hélene Papadopoulos and Geoffroy Peeters. Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *International Workshop*

on Content-Based Multimedia Indexing, pages 53–60. IEEE, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4275055.

[153] Yushi Ueda, Yuki Uchiyama, Takuya Nishimoto, Nobutaka Ono, and Shigeki Sagayama. HMM-based approach for automatic chord detection using refined acoustic features. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010. URL http://hil.t.u-tokyo.ac.jp/publications/download.php?bib=Ueda2010ICASSP03.pdf.

[154] Katy Noland and Mark B. Sandler. Key Estimation Using a Hidden Markov Model. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 121–126, 2006. URL https://www.researchgate.net/profile/Mark_Sandler2/publication/220723459_Key_Estimation_Using_a_Hidden_Markov_Model/links/54118dd10cf29e4a23296c48.pdf.

[155] Mark Levy and Mark Sandler. Structural segmentation of musical audio by constrained clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):318–326, 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4432648.

[156] Emmanuel Vincent and Xavier Rodet. Music transcription with ISA and HMM. In *International Conference on Independent Component Analysis and Signal Separation*, pages 1197–1204, 2004. URL http://link.springer.com/10.1007%2F978-3-540-30110-3_151.

[157] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1054010.

[158] G. David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1450960.

[159] Olivier Gillet, Slim Essid, and Gal Richard. On the correlation of automatic audio and visual segmentations of music videos. *IEEE Transactions on Circuits and*

Systems for Video Technology, 17(3):347–355, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4118238.

[160] Masataka Goto. A chorus-section detecting method for musical audio signals. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, volume 5, pages V–437, 2003. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1200000.

[161] Kristoffer Jensen, Jieping Xu, and Martin Zachariasen. Rhythm-Based Segmentation of Popular Chinese Music. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 374–380, 2005. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.136&rep=rep1&type=pdf.

[162] George Tzanetakis and Perry Cook. Multifeature audio segmentation for browsing and annotation. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 103–106, 1999. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=810860.

[163] Antti Eronen and F. Tampere. Chorus detection with combined use of MFCC and chroma features and image processing filters. In *Proc. of 10th International Conference on Digital Audio Effects*, pages 229–236, 2007. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.384.1101&rep=rep1&type=pdf.

[164] Douglas Turnbull, Gert RG Lanckriet, Elias Pampalk, and Masataka Goto. A Supervised Approach for Detecting Boundaries in Music Using Difference Features and Boosting. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 51–54, 2007. URL http://jimi.ithaca.edu/~dturnbull/Papers/Turnbull_MusicBoundary_ISMIR07.pdf.

[165] Karen Ullrich, Jan Schlüter, and Thomas Grill. Boundary Detection in Music Structure Analysis using Convolutional Neural Networks. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 417–422, 2014. URL https://dav.grrrr.org/public/pub/ullrich_schlueter_grill-2014-ismir.pdf.

[166] Thomas Grill and Jan Schlüter. Music Boundary Detection Using Neural Networks on Combined Features and Two-Level Annotations. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2015. URL `http://grrrr.org/pub/grill_schlueter-2015-ismir.pdf`.

[167] Jouni Paulus, Meinard Müller, and Anssi Klapuri. State of the Art Report: Audio-Based Music Structure Analysis. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 625–636, 2010. URL `http://www.iki.fi/paulus/pubs/paulus_ismir10_star_presentation.pdf`.

[168] Dirk Van Steelant, Bernard De Baets, Hans De Meyer, Marc Leman, Jean-Pierre Martens, Lieven Clarisse, and Micheline Lesaffre. Discovering structure and repetition in musical audio. In *Proceedings of Eurofuse Workshop*, 2002. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.3.7108&rep=rep1&type=pdf`.

[169] Jouni Paulus and Anssi Klapuri. Music structure analysis by finding repeated parts. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 59–68, 2006. URL `http://dl.acm.org/citation.cfm?id=1178733`.

[170] Tom Collins, Sebastian Böck, Florian Krebs, and Gerhard Widmer. Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *53rd Audio Engineering Society Conference: Semantic Audio*, 2014. URL `http://www.aes.org/e-lib/browse.cfm?elib=17096`.

[171] Harald Grohganz, Michael Clausen, Nanzhu Jiang, and Meinard Müller. Converting Path Structures Into Block Structures Using Eigenvalue Decompositions of Self-Similarity Matrices. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 209–214, 2013. URL `http://ismir2013.ismir.net/wp-content/uploads/2013/09/23_Paper.pdf`.

[172] Ron J. Weiss and Juan Pablo Bello. Identifying Repeated Patterns in Music Using Sparse Convolutive Non-Negative Matrix Factorization. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2010.

[173] Brian McFee and Dan Ellis. Analyzing Song Structure with Spectral Clustering. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 405–410, 2014. URL https://bmcfee.github.io/papers/ismir2014_spectral.pdf.

[174] Florian Kaiser, Marina Georgia Arvanitidou, and Thomas Sikora. Audio similarity matrices enhancement in an image processing framework. In *9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 67–72. IEEE, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5972522.

[175] Florian Kaiser and Thomas Sikora. Music Structure Discovery in Popular Music using Non-negative Matrix Factorization. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 429–434, 2010. URL https://pdfs.semanticscholar.org/885d/60fff6de71c562d35564dfacbf4a44e8d1e8.pdf.

[176] Samer Abdallah, Katy Noland, Mark Sandler, Michael A. Casey, and Christophe Rhodes. Theory and evaluation of a Bayesian music structure extractor. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2005. URL http://eprints.gold.ac.uk/2349/.

[177] Hanna M. Lukashevich. Towards Quantitative Measures of Evaluating Song Segmentation. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 375–380, 2008. URL https://books.google.com/books?hl=fr&lr=&id=OHp3sRnZD-oC&oi=fnd&pg=PA375&dq=towards+quantitative+measures+of+evaluationg+song+segmentation&ots=oEPPtLkw83&sig=F2MNqLqBK0nTojv2_2hJEofqUYk.

[178] Jordan BL Smith, Isaac Schankler, and Elaine Chew. Listening as a creative act: Meaningful differences in structural annotations of improvised performances. *Music Theory Online*, 20(3), 2014. URL http://www.mtosmt.org/issues/mto.14.20.3/mto.14.20.3.smith_schankler_chew.html.

[179] Yizhao Ni, Matt McVicar, Raul Santos-Rodriguez, and Tijl De Bie. Understanding effects of subjectivity in measuring chord estimation accuracy. *IEEE Transactions*

on Audio, Speech, and Language Processing, 21(12):2607–2615, 2013. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6587770.

[180] Arthur Flexer. On inter-rater agreement in audio music similarity. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2014. URL http://www.terasoft.com.tw/conf/ismir2014/proceedings/T045_256_Paper.pdf.

[181] Thomas A. Lampert, André Stumpf, and Pierre Gançarski. An empirical study into annotator agreement, ground truth estimation, and algorithm evaluation. *IEEE Transactions on Image Processing*, 25(6):2557–2572, 2016. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7437472.

[182] Jordan Bennett Louis Smith, John Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J. Stephen Downie. Design and creation of a large-scale database of structural annotations. In *International Society for Music Information Retrieval (ISMIR) Conference*, volume 11, pages 555–560, 2011. URL http://www.mirlab.org/conference_papers/International_Conference/ISMIR%202011/papers/PS4-14.pdf.

[183] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE transactions on Speech and Audio Processing*, 10(5):293–302, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1021072.

[184] Bob L. Sturm. The state of the art ten years after a state of the art: Future research in music information retrieval. *Journal of New Music Research*, 43(2):147–172, 2014. URL http://www.tandfonline.com/doi/abs/10.1080/09298215.2014.894533.

[185] Tom LH Li and Antoni B. Chan. Genre classification and the invariance of MFCC features to key and tempo. In *International Conference on MultiMedia Modeling*, pages 317–327. Springer, 2011. URL http://link.springer.com/10.1007%2F978-3-642-17832-0_30.

[186] Ugo Marchand and Geoffroy Peeters. Swing Ratio Estimation. In *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx). Trondheim, Norway, Nov 30 - Dec 3, 2015*, 2015.

[187] Matthias Mauch, Chris Cannam, Matthew Davies, Simon Dixon, Christopher Harte, Sefki Kolozali, Dan Tidhar, and Mark Sandler. OMRAS2 metadata project 2009. In *International Society for Music Information Retrieval (ISMIR) Conference*, page 1, 2009. URL `http://matthiasmauch.de/_bilder/late-breaking-C4DM.pdf`.

[188] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 591–596. University of Miami, 2011. URL `http://academiccommons.columbia.edu/catalog/ac:148381`.

[189] Simon K. Warfield, Kelly H. Zou, and William M. Wells. Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. *IEEE transactions on medical imaging*, 23(7):903–921, 2004. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1309714`.

[190] Simon K. Warfield, Kelly H. Zou, and William M. Wells. Validation of image segmentation by estimating rater bias and variance. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 839–847, 2006. URL `http://link.springer.com/10.1007%2F11866763_103`.

[191] Tomi Kauppi, Joni-Kristian Kamarainen, Lasse Lensu, Valentina Kalesnykiene, Iiris Sorri, Heikki Kälviäinen, Hannu Uusitalo, and Juhani Pietilä. Fusion of multiple expert annotations and overall score selection for medical image diagnosis. In *Scandinavian Conference on Image Analysis*, pages 760–769, 2009. URL `http://link.springer.com/chapter/10.1007/978-3-642-02230-2_78`.

[192] Thomas Robin Langerak, Uulke A. van der Heide, Alexis NTJ Kotte, Max A. Viergever, Marco van Vulpen, and Josien PW Pluim. Label fusion in atlas-based segmentation using a selective and iterative method for performance level estimation (SIMPLE). *IEEE Transactions on Medical Imaging*, 29(12):2000–2008, 2010. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5523952`.

[193] Chris Cannam, Christian Landone, and Mark Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In

Proceedings of the 18th ACM international conference on Multimedia, pages 1467–1468, 2010. URL http://dl.acm.org/citation.cfm?id=1874248.

[194] Olivier Lartillot, Donato Cereghetti, Kim Eliard, Wiebke J. Trost, Marc-Andre Rappaz, and Didier Grandjean. Estimating tempo and metrical features by tracking the whole metrical hierarchy. In *Proceedings of the 3rd International Conference on Music & Emotion (ICME3)*. University of Jyvaskyla, Department of Music, 2013. URL https://jyx.jyu.fi/dspace/handle/123456789/41612.

[195] Arthur Flexer and Thomas Grill. The Problem of Limited Inter-rater Agreement in Modelling Music Similarity. *Journal of New Music Research*, pages 1–13, 2016. URL http://www.tandfonline.com/doi/abs/10.1080/09298215.2016.1200631.

[196] Chunyang Song, Andrew J. R. Simpson, Christopher A. Harte, Marcus T. Pearce, and Mark B. Sandler. Syncopation and the Score. *PLoS ONE*, 8(9):e74692, September 2013. doi: 10.1371/journal.pone.0074692. URL http://dx.doi.org/10.1371/journal.pone.0074692.

[197] Juan Pablo Bello and Jeremy Pickens. A Robust Mid-Level Representation for Harmonic Content in Music Signals. In *International Society for Music Information Retrieval (ISMIR) Conference*, volume 5, pages 304–311, 2005. URL http://www.ismir2005.ismir.net/proceedings/1038.pdf.

[198] Daniel PW Ellis and Graham E. Poliner. Identifyingcover songs' with chroma features and dynamic programming beat tracking. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, volume 4, pages IV–1429, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4218379.

[199] Garth Griffin, Youngmoo E. Kim, and Douglas Turnbull. Beat-sync-mash-coder: A web application for real-time creation of beat-synchronous music mashups. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 437–440, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5495743.

[200] Olivier Lartillot, Tuomas Eerola, Petri Toiviainen, and Jose Fornari. Multi-Feature Modeling of Pulse Clarity: Design, Validation and Optimization. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 521–526, 2008. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.156.2069&rep=rep1&type=pdf`.

[201] Peter Grosche, Meinard Müller, and Craig Stuart Sapp. What Makes Beat Tracking Difficult? A Case Study on Chopin Mazurkas. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 649–654, 2010. URL `http://ismir2010.ismir.net/proceedings/ismir2010-110.pdf`.

[202] Peter Grosche, M. Müller, and Frank Kurth. Cyclic tempogram, a mid-level tempo representation for music signals. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 5522–5525, 2010. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5495219`.

[203] Balaji Thoshkahna, Meinard Müller, Venkatesh Kulkarni, and Nanzhu Jiang. Novel Audio Features for Capturing Tempo Salience in Music Recordings. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2015.

[204] Frederic Font and Xavier Serra. Tempo Estimation For Music Loops And a Simple Confidence Measure. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2016. URL `http://mtg.upf.edu/system/files/publications/font2016ismir.pdf`.

[205] François-Joseph Fétis and Arthur Pougin. *Biographie universelle des musiciens et bibliographie générale de la musique*, volume 4. Firmin-Didot et cie, 1874. URL `https://books.google.com/books?hl=fr&lr=&id=ZONTAAAAcAAJ&oi=fnd&pg=PA1&dq=F%C3%A9tis&ots=mQeznLAiH8&sig=qyKtqXwMZ1sPiBFQmSWm_qVoAgQ`.

[206] François-Joseph Fétis. *Traité complet de la théorie et de la pratique de l'harmonie contenant la doctrine de la science et de l'art.* Brandus et Cía., 1853. URL `https://books.google.com/books?hl=fr&lr=&id=6LJQQ7P4VFUC&oi=fnd&pg=PR1&dq=F%C3%A9tis&ots=tBmQpzPgx4&sig=5VIYxrRGzeo4AwTtIET_mkLh8gk`.

[207] Mary I. Arlin. Metric mutation and modulation: the nineteenth-century speculations of F.-J. Fétis. *Journal of Music Theory*, pages 261–322, 2000. URL http://www.jstor.org/stable/3090680.

[208] Elliott Carter. The time dimension in music. *Music Journal*, 23(8):29, 1965. URL http://search.proquest.com/openview/f3328035580704d9402861b3192e581b/1?pq-origsite=gscholar.

[209] George Peter Tingley. Metric Modulation and Elliott Carter's" First String Quartet". *Indiana Theory Review*, pages 3–11, 1981. URL http://www.jstor.org/stable/24045947.

[210] Jonathan W. Bernard. The evolution of Elliott Carter's rhythmic practice. *Perspectives of New Music*, pages 164–203, 1988. URL http://www.jstor.org/stable/833189.

[211] David Schiff. *The Music of Elliott Carter*. Cornell University Press, 1998.

[212] Denis Vermaelen. *L'oeuvre vocale d'Elliott Carter de 1975 à 1981*. PhD thesis, Tours, 1995. URL http://www.theses.fr/1995TOUR2012.

[213] Jean-Luc Bouchard. *La modulation agogique : définition, typologie et analogie avec la modulation tonale*. Masters Thesis, Faculté de Musique, Universite de Laval, Québec, Canada, 2008.

[214] Fernando Benadon. Towards a theory of tempo modulation. In *Proceedings of the 8th International Conference on Music Perception and Cognition: ICMPC8*, pages 563–567, 2004. URL http://www.ludions.com/aru/electr/resources/Benadon2004.pdf.

[215] Anssi Klapuri. Musical meter estimation and music transcription. In *Cambridge Music Processing Colloquium*, pages 40–45, 2003. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.8559&rep=rep1&type=pdf.

[216] Matthias Mauch, Katy Noland, and Simon Dixon. Using Musical Structure to Enhance Automatic Chord Transcription. In *International Society for Music*

Information Retrieval (ISMIR) Conference, pages 231–236, 2009. URL https://www.eecs.qmul.ac.uk/~simond/pub/2009/ISMIR2009-Mauch-PS2-7.pdf.

[217] Judith C. Brown. Calculation of a constant Q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991. URL http://scitation.aip.org/content/asa/journal/jasa/89/1/10.1121/1.400476.

[218] Oriol Nieto and Tristan Jehan. Convex non-negative matrix factorization for automatic music structure identification. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 236–240. IEEE, 2013. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6637644.

[219] Mikkel N. Schmidt. Speech separation using non-negative features and sparse non-negative matrix factorization. In *Technical report*. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2007. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.135.9337&rep=rep1&type=pdf.

[220] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares. 2006. URL https://smartech.gatech.edu/handle/1853/14461.

[221] Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1066–1074, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4100700.

[222] Shaodong Wang, Nan Wang, Dacheng Tao, Lefei Zhang, and Bo Du. A KL divergence constrained sparse NMF for hyperspectral signal unmixing. In *International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pages 223–228. IEEE, 2014. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6982689.

[223] Jonathan Le Roux, Felix Weninger, and John R. Hershey. Sparse NMF–half-baked or well done? *Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA,*

Tech. Rep., no. TR2015-023, 2015. URL http://www.merl.com/publications/docs/TR2015-023.pdf.

[224] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469, 2004. URL http://www.jmlr.org/papers/v5/hoyer04a.html.

[225] Weixiang Liu, Nanning Zheng, and Xiaofeng Lu. Non-negative matrix factorization for visual coding. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, volume 3, pages III–293, 2003. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1199270.

[226] Robert Peharz and Franz Pernkopf. Sparse nonnegative matrix factorization with l0-constraints. *Neurocomputing*, 80:38–46, March 2012. ISSN 0925-2312. doi: 10.1016/j.neucom.2011.09.024. URL http://www.sciencedirect.com/science/article/pii/S0925231211006370.

[227] Vincent YF Tan and Cédric Févotte. Automatic relevance determination in nonnegative matrix factorization with the β-Divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1592–1605, 2013. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6341758.

[228] Vincent YF Tan and Cédric Févotte. Supplementary material to "Automatic Relevance Determination in Nonnegative Matrix Factorization with the β-Divergence". https://www.ece.nus.edu.sg/stfpage/vtan/supp_mat.pdf, 2012. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.701.4932&rep=rep1&type=pdf.

[229] Emmanuel J. Candes, Michael B. Wakin, and Stephen P. Boyd. Enhancing sparsity by reweighted l 1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008. URL http://link.springer.com/article/10.1007/s00041-008-9045-x.

[230] Thomas Grill and Jan Schlüter. Structural Segmentation With Convolutional Neural Networks MIREX Submission. In *Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2015. URL `http://www.music-ir.org/mirex/abstracts/2015/GS1.pdf`.

[231] Jan Schlüter, Karen Ullrich, and Thomas Grill. Structural segmentation with convolutional neural networks mirex submission. *Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2014. URL `http://www.music-ir.org/mirex/abstracts/2014/SUG2.pdf`.

[232] Benjamin Martin, Pierre Hanna, Matthias Robine, Pascal Ferraro, and others. Structural analysis of harmonic features using string matching techniques. *Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2012. URL `http://www.music-ir.org/mirex/abstracts/2011/MHRAF1.pdf`.

[233] Joan Serra, Meinard Müller, Peter Grosche, and Josep LI Arcos. The importance of detecting boundaries in music structure annotation. In *The Music Information Retrieval Evaluation eXchange (MIREX)*, 2012. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.422.1192`.

[234] Carl Schachter. The Prelude from Bach's Suite No. 4 for violoncello solo: The submerged urlinie. *Current Musicology*, 56:54–71, 1994. URL `https://academiccommons.columbia.edu/catalog/ac:182274`.

[235] Miriam K. Whaples. Bach's Recapitulation Forms. *The Journal of Musicology*, 14 (4):475–513, 1996. URL `http://www.jstor.org/stable/764070`.

[236] Eric Chafe. Key structure and tonal allegory in the passions of JS Bach: An introduction. *Current Musicology*, (31):39, 1981. URL `http://search.proquest.com/openview/51bb730f1f7e973291334a070af9d06d/1?pq-origsite=gscholar&cbl=1819340`.

[237] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. URL `http://www.nature.com/nature/journal/v521/n7553/abs/nature14539.html`.

[238] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015. URL `https://arxiv.org/abs/1506.06579`.

[239] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. 2016. URL `http://arxiv.org/abs/1605.09304`.

[240] Yongxin Yang, Yu Zheng, and Timothy M. Hospedales. Gated Neural Networks for Option Pricing: Rationality by Design. *arXiv preprint arXiv:1609.07472*, 2016. URL `https://arxiv.org/abs/1609.07472`.

[241] Yaser S. Abu-Mostafa. A method for learning from hints. In *Proceedings of the 5th International Conference on Neural Information Processing Systems*, pages 73–80, 1992. URL `http://dl.acm.org/citation.cfm?id=2987071`.

[242] Andrzej Cichocki, Shun-ichi Amari, Rafal Zdunek, Raul Kompass, Gen Hori, and Zhaohui He. Extended SMART algorithms for non-negative matrix factorization. In *International Conference on Artificial Intelligence and Soft Computing*, pages 548–562. Springer, 2006. URL `http://link.springer.com/chapter/10.1007/11785231_58`.

[243] Masahiro Nakano, Hirokazu Kameoka, Jonathan Le Roux, Yu Kitano, Nobutaka Ono, and Shigeki Sagayama. Convergence-guaranteed multiplicative algorithms for nonnegative matrix factorization with β-divergence. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, pages 283–288. IEEE, 2010. URL `http://ieeexplore.ieee.org/abstract/document/5589233/`.