Sponsoring Committee: Professor Morwaread M. Farbood
Professor Juan P. Bello
Doctor Tristan Jehan

DISCOVERING STRUCTURE IN MUSIC:

AUTOMATIC APPROACHES AND PERCEPTUAL EVALUATIONS

Oriol Nieto

Program in Music Technology
Department of Music and Performing Arts Professions

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in the
Steinhardt School of Culture, Education, and Human Development
New York University
2015

To Amalia, Ana, Antonio and Juan.

ACKNOWLEDGEMENTS

This epic adventure would have been impossible to undertake without the help of many wonderful individuals who unconditionally shared their seemingly endless wisdom with me. Here, and in chronological order, I give them my eternal gratitude.

First of all, to family. Especially: my parents, who never gave up on me; my sister, who is probably the strongest person I know (I am so proud of you); and my grandparents, to whom I dedicate this work.

To my half brother Daniel Bolsa Madrid, for teaching me Life since I was eight. To my second family, also known as La Caverna, or The Old School, or Can Paco's Crew, including, in alphabetical order: Santi Bonillo, Daniel Fàbregas, Daniel González, Bernat Maspons, Albert Mesas, Guillem and Arnau Mayoral, and Adrià Montanya. Special mention to Eduard and Roger Piqueras (Chewbacca!), Anna Mercader, and Sergi Mansilla, for showing me video games (and peach *juice*), how to deal with our parents' divorce, and GNU/Linux, respectively.

To Maria Josep, my math teacher in high school, who made me learn (and enjoy!) Calculus. To Lluís, my music teacher in high school, who unraveled the world of The Beatles to me.

To Jordi Bosch, Pau Farell, and Gerson Gelabert for making me sing in Darkgeon, my first metal band. To Carles Ferreiro, Jordi Llobet, Marc Prim, Felip Sánchez (miss you my friend), David Alarcón and Albert Comerma for

all the Sargon years, which I look back with pride, nostalgia, and happiness (I strongly believe that Vida is the best metal album in Catalan to date). To Dani "de Aquitania," for inspiring me in all possible ways, and for the never ending nights with Madee, Buses, Trains, Chairs, Volls, Cazallas, and Marsellas.

To Marc Alier for being the example of the teacher I would like to become and the ideal advisor for my Computer Science undergraduate thesis. To Miquel Barceló, for publishing Neal Stephenson in Spain.

To all my BEST friends, especially: João Terra (that train in Siberia will not ever be forgotten), Cemil Ozan, Páll Jens, the rest of the participants and organizers of the amazing course in Ekaterinburg; Marc Velasco, Alba Gil, Elena Guasch, Beta Foix, Erik Abner, and the rest of the LBG Barcelona. Europe got much smaller, my English got much better, and my desire to study abroad got much stronger with you around.

To all the wonderful people I met the years of living in Barcelona, particularly: Amanda "Amandarina" Fernández, Leslie Cristobal, Sandra Maya, Ivan Rivero, Donato Lorenzo, Anna Basagaña, and the rest of València 477. Also thanks to all the guys at Madee, for giving me the opportunity to tour with them across Spain in several occasions.

To Justin Salamon (to whom, at the end of my very first day at the Music Technology Group as a master's student, I asked: "what is a Fourier Transform?"), for being my colleague at the MTG, roommate in Barcelona, and friend in life. To Vassilis Pantazis, for showing me Meshuggah. To Elena Martínez, for teaching me the basics of signal processing. To Jordi Janer, for making me use Python. To Xavier Serra, for showing me the fascinating world

of music technology, and inspiring me to pursue a Ph.D. To Jordi Bonada, for being the best advisor I could ever had during my master's thesis.

To La Caixa Fellowship, for giving me this once-in-a-lifetime opportunity to cross the Atlantic to keep studying my two passions —computers and music— in the best schools in the world. Thanks to my friends who also got this scholarship in 2008, especially: Ferran Masip, Franc Camps, Carlos Fernández, Daniel Climent, Jordi Graupera, Marta Martínez, Juan Astasio, Marta Fenollosa, Almudena Toral, Sara Cabal, Sergi Casanelles, Tomàs Peire, Juan Argote, and Pau Guinart. After having taken a little peek to your wonderfully bright minds, I still wonder why I obtained this fellowship.

To Professor Dimitar Deliysky and his beautiful family, who hosted when I visited the University of South Carolina, and later gave me the opportunity to "scream" in front of the world experts on voice production in the gala dinner of the AQL conference.

To my professors at Stanford, particularly: Jonathan Abel, Julius O. Smith, Ge Wang, and Jonathan Berger. You have inspired me in so many ways that I still feel I have to somehow pay it back. Also, to my fellow students and friends: Nick Bryan, Roy Fejgin, Blair Kaneshiro, Nick Kruge, Puja Kumar, June Oh, Colin Raffel, Adam Sheppard, Adam Somers, and Sean Zhang. What a year we had together in California!

Special thanks to Jordan Rudess, an inspiration since I was sixteen, who I now have the privilege to have as a friend (thanks for coming all the way to Stanford to play a couple of shows with us!). Thanks to him, I have met two of the best musicians I know: Eyal Amir and Eren Başbuğ. Collaborating with these titans has been such a powerful inspiration, I feel so lucky and thankful for having had this opportunity.

To my advisor at NYU, Mary Farbood, for all the many good hours encouraging and advising me in the pursuit of this Ph.D. This road would have been much more painful without you. To my "second" advisor, Juan P. Bello, for challenging me in every possible way, and making me not only a much better researcher, but a much better human. To Dennis Shasha, for his strenuous class of heuristic problem solving, and all the pleasant and stimulating lunches together. To the rest of the professors at MARL, especially: Agnieszka Roginska, Tae Hong Park, Alex Ruthman, Tom Beyer, and Panayotis Mavromatis. To my fantastic C Programming students, who really teach me how to teach.

To my fellow students and friends at NYU: Areti Andreopoulou, Rachel Bittner, Braxton Boren, Taemin Cho, Jon Forsyth, Aron Glennon, Brian McFee, Michael Musick, Andrew Telichan, and Finn Upham. Very special thanks to Eric Humphrey. I doubt I would have finished this without him, it has been a pleasure to share the office and to have learned so much by his titanic side. Also big thanks to those of you who reviewed the "Catalanglish" of this document, I owe you so much (or so many beers, at the very least).

To the Caja Madrid scholarship, since I would not have been able to continue my Ph.D without their financial help.

To the people at The Echo Nest, who treated me like one of them since the very first day of my internship. Especially Tristan Jehan (who gave me the opportunity in the first place, and from whom I learned so much as an engineer, researcher, and friend), Ruofeng Chen, Brian Whitman, Nicola Montechio, Hunter McCurry, Noura Howell, and Amanda Bulger. Thanks to Ava Vitali for the good and geeky times in Boston.

To Nathalie Alegre for making me a better human being in all imaginable aspects. I am so lucky and thankful to have such a great partner in life. Special

thanks to her family, particularly to Isabel, Pablo, and El Nono, for making me feel like home for the two and a half months I was in Lima, where this document originated in June of 2014.

To my ISMIR colleagues, especially: Amélie Anglade, Eric Battenberg, Sebastian Böck, Michael Casey, Oscar Celma, Tom Collins, Emanuele Coviello, Sander Dieleman, Frederic Font, Masataka Goto, Philippe Hamel, Katie Kinnaird, Matthias Mauch, Matt McVicar, Geoffroy Peeters, Jan Schlüter, Erik Schmidt, Jeff Scott, Joan Serrà, Siddharth Sigtia, Moha Sordo, Jessica Thompson, and Aäron van den Oord. Especial thanks to the "music structure analysis" reading group: Jordan Smith, Nanzhu Jiang, and Meinard Müller. ISMIR, in general, would not make sense without you.

To the guys at my current metal band, Midnight Blue: Seth, Jason, Giovani, and Jan. Now that I am about to become a doctor, we should have no problems on saving the world with our music.

Finally, thanks to Thiru Kumar ("The Dosas Man") for making the best food in New York City. Thanks to Bare Burger for the wonderful beers and, of course, –now veggie– burgers. And thanks to Vim and LaTeX for letting me create this document without suffering as much as I would have without their existence.

You —and all of those who I left out due to my terrible memory, sorry!— are all titans.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

"I'm reaching for the random
or what ever will bewilder me.
And following our will and wind
we may just go where no one's been."

-Tool, *Lateralus* (Climax before the Outro).

CHAPTER I

INTRODUCTION

Structure is everywhere. From tiny atoms to massive galaxy clusters, we keep discovering structural relationships that make us better apprehend our surroundings. When chaos is perceived, the curious and the patient will eventually develop the necessary processes and tools to identify the structural patterns that give meaning to this apparent mayhem. We might need specific triggers to come to the realization of the structure of the world we live in (e.g. some might need to experience an apple falling from a tree, others to fly a kite in the middle of a lightning storm), but that will not stop our curiosity. Such is the way we have reached the current state of human knowledge, which will likely to continue to expand as long as humanity exists.

Additionally, and as part of our *inherent structure*, some of us also have the urge to express ourselves, to communicate feelings, to share emotions. In fact, we could argue that this trait in our species is what makes us really "human." And to that end, music has been the preferred means of expression by many, which might explain why it is one of the oldest, most fascinating, and pervasive art forms that exist today.

This dissertation aims at shedding more light on the understanding of structure in music, both from an algorithmic and perceptual points of view. The analysis of musical structure will be investigated from different angles by proposing novel methods for accomplishing multiple tasks such as the auto-

matic identification of large-scale musical segments or the generation of music summaries. Whether or not some representation of music perception can actually improve the current techniques that aim at quantifying these tasks will also be examined. Thus, the work presented here falls somewhere in between the fields of music information retrieval (MIR) and music perception and cognition (MPC). More specifically, by benchmarking MIR systems using perceptually-based evaluations (designed employing standard MPC tools), researchers would obtain solutions that are more perceptually relevant, and thus closer to user preferences.

In this introductory chapter the scope of the study, its motivation, and its main contributions will be discussed in order to give a high-level overview of the entire dissertation.

1 Scope of this Study

The primary questions that this dissertation addresses are the following: how can we better *teach* computers to interpret the structure of music? And, given the inherent ambiguity of music, how can the methodologies that automatically discover this structure reflect or make use of the differences in perception? Consequently, and as the title of this work suggests, this study can be divided in two main blocks, whose respective goals are (i) to propose novel frameworks and algorithms for multiple MIR tasks of musical structure analysis, and (ii) to explore the perceptual disagreement of the structure of music from a more MPC point of view in order to introduce new evaluations that better align with human preference.

As for the first part (i.e., the automatic approaches), the aim is to present

novel techniques that push the state-of-the-art not necessarily in specific evaluation numbers (since these can yield misleading results, as it will be discussed), but in terms of exploring the usage of mathematical and machine learning tools that never before were employed to identify structure in music. These novel approaches should shed light on the understanding of the present and future challenges when implementing algorithms that aim at discovering this structure.

Questioning our current evaluation metrics and proposing perceptually enhanced ones is the second main goal of this work. In order to do so — following a more MPC-oriented approach— multiple experiments were conducted in order to quantify the amount of disagreement between listeners when perceiving the musical structure. It is beyond the scope of this dissertation to design a general model of these perceptual differences, however the limitations of these metrics (mostly originated due to the high degree of subjectivity in this task) are exposed, which I aim to overcome by proposing novel evaluations that should provide a better comprehension of the structure of music towards a general model. By using these evaluations to optimize music structure algorithms, MIR practitioners could still follow the same standard methodology of developing new solutions (i.e., produce results and compare them against human annotations using an evaluation that yields a certain score), and produce systems that better align with user expectation.

Even though, theoretically, these novel methods should be able to identify and evaluate structure for any type of music, there is a strong bias towards analyzing Western popular music, as is the case in general for common methods in MIR, but not MPC. This is in part because structure is usually perceived with less ambiguity in this genre, as some of the experiments results

presented in this work suggest. Therefore, some of the techniques presented in this dissertation will focus primarily on this type of music.

2 Motivation

The analysis of the structure of the organized sounds from which music is constructed can help us better understand how humans generate and process musical information. This information could then be used by machines in order to provide humans with useful knowledge about their music collections, artists, music genres, etc. Examples of such methods are:

- Better music recommendation systems: segment-level recommendation (e.g., users may want to listen to tracks that are similar to a specific segment of a song).
- Smarter music players: improved intra piece navigation; creating automatic *mash-ups* based on segment similarity.
- Optimal music summarization: Short preview excerpts that successfully summarize all the different sections of a given track.

A wide variety of MIR techniques that automatically extract from audio the structure of a musical piece have been proposed in the past decade, with some degree of success (Dannenberg and Goto, 2008; Paulus et al., 2010; Bello et al., 2011), and more recently, these techniques have been surpassed by applying a set of new structural features (Serrà et al., 2014). However, these approaches share a common denominator: the lack of cognitive principles on which to base their mathematical, statistical, or engineering procedures. This should raise obvious concerns about the limitations of these methods (Casey

and Slaney, 2006; Wiggins, 2009; Casey et al., 2008b; Karydis et al., 2010): can we in fact solve this particular task —i.e., analysis of music structure— without considering any cognitive representation of the understanding of musical information? This opens up apparent paths of exploration that are investigated in the current work.

The problem of human subjectivity has already been discussed for the tasks of chord recognition (Ni et al., 2013), music similarity (Flexer, 2014), and beat detection (Davies and Böck, 2014). Moreover, the problem of perceptual agreement in the context of music structure has been published in a relevant dissertation (Smith, 2014) that supports previous findings concerning strong subjectivity when evaluating the structure in music (Bruderer et al., 2009). This relates to the fact that MIR methods tend to lack multiple human annotations in their datasets, and it provides motivation for including these perceptual disagreements into the process of developing automatic techniques for music structure analysis, which is one of the central topics of the current work.

Traditionally, the standard methodology of developing MIR systems can be divided into the following steps:

i. Design and implement algorithms that produce an estimated output (e.g., time points of music segment boundaries).

ii. Have access to one or several datasets that contain human annotations on which to compare the estimated outputs (e.g., boundary time points of the SALAMI dataset, which it will be thoroughly employed in chapters III, IV, and V).

iii. Evaluate the algorithms designed in step i by using a specific evaluation

that yields one or more scores representing how closely the estimated outputs from the algorithms are to the human annotations of the datasets.

iv. Use these scores in order to redefine and optimize the implementation of the algorithms.

The perceptual evaluations that will be presented in this work, which are designed by making use of standard tools in MPC, could potentially replace the standard evaluations typically employed in step iii. By doing so, the MIR methodology would not be altered, but rather the scores that are used to optimize algorithms would change, which would result in more perceptually meaningful MIR systems.

This approach brings the fields of MIR and MPC closer together, something that has been suggested as one of the main goals to break current "glass-ceilings" (Casey et al., 2008b; Downie et al., 2009). By bridging the distance between methodologies in these two fields, it is the hope of the author to generate beneficial knowledge that would be valuable in both disciplines.

3 Dissertation Outline

The principal contributions of this dissertation are organized in two parts: the first one (Chapters III and IV) presents novel MIR methods that automatically extract the structure of a musical piece. The second part (Chapters V and VI) explores the problem of musical structure analysis from an MPC perspective, where investigations for overcoming the subjectivity effect by having novel evaluation metrics are presented.

A more detailed outline of the work is described next.

Chapter II: A review of the problem of musical structure analysis is presented from both MIR and MPC perspectives. Furthermore, the standard existing techniques to automatically discover structure are discussed, including the basics of feature extraction from audio. Additionally, a review of the common evaluation metrics is also presented, opening up the discussion about the limitations of these metrics due to the high degree of subjectivity present in the perception of the structure in music. Finally, a new open source project to evaluate the most common tasks in MIR, which includes implementations of all the evaluations reviewed in this chapter, is also introduced.

Chapter III: Two novel methods that could be classified under the MIR tasks of music summarization and pattern discovery are presented. These tasks, introduced in the previous chapter, aim at discovering specific aspects of structure of a given piece: the former employs these structures in order to generate audible summaries capturing the most representative parts of the track, while the latter looks for repetition in the piece to extract all the possible musical patterns (including motives, riffs, phrases, and long-scale sections).

Chapter IV: Two additional algorithms are introduced, in this case for the MIR task of music segmentation. The first algorithm makes use of a constrained version of the unsupervised machine learning technique of non-negative matrix factorization to segment a given piece into all its large-scale sections and then dissects the same factorization to label all the extracted segments. The second and last algorithm presented focuses on the structural grouping (or labeling) subproblem of music seg-

mentation by using the two-dimensional Fourier magnitude coefficients to cluster the previously extracted segments based on their harmonic similarity.

Chapter V: In this chapter a new framework to analyze the structure of music is presented, building on some of the most relevant existing algorithms. This framework not only simplifies the evaluation and comparison of multiple algorithms, but also reduces the complexity of the analysis of the agreement between multiple human annotators. To achieve this, the framework includes a novel file format based on JSON that is designed to store multiple human annotations and multiple tasks in a single file. Moreover, a large dataset with more than 2,000 human annotations of musical structure and on which our framework operates is also discussed. Lastly, a description of the acquisition of several human annotations per track as an experiment to explore the subjectivity effect of the perception of structure is included. The tracks that are annotated in the study are automatically selected using the analysis framework presented here.

Chapter VI: This is the final chapter describing the main contributions and encompasses the analysis of the previously collected human annotations in which the problem of subjectivity is most apparent. As an attempt to overcome this issue, four novel types of merging the multiple annotations into flat and hierarchical segments are presented. In order to evaluate these new merged annotations, two metrics are introduced for the flat and hierarchical segments, respectively. Furthermore, a comparison between the robustness of the standard annotations and the merged ones is also presented, in which the superiority of the merging techniques

is demonstrated. Finally, considering the scenario in which additional human annotations are not available, multiple studies on one of the existing metrics for music segmentation are discussed, concluding that the precision value of the F-measure is more perceptually relevant than the recall. These evaluations could be used in MIR in order to produce music structure analysis systems that align better with user preferences.

Chapter VII: This chapter concludes the dissertation. Discussions about the main findings and implications are included, along with perspectives on the future of automatic discovery of musical structure.

4 Contributions

The primary contributions of this dissertation and related references to relevant published papers by the author are listed below:

- A transparent open-source project to evaluate the most common MIR techniques (Raffel et al., 2014).
- An algorithm to generate audio summaries from recordings using two criteria, compression and disjoint information (Nieto et al., 2012).
- A method that constitutes the state-of-the-art in discovering musical patterns from audio by using standard music segmentation techniques (Nieto and Farbood, 2014a).
- The use of a convex constraint applied to non-negative matrix factorization to identify the homogeneous sections of a musical piece (Nieto and Jehan, 2013).
- The utilization of two-dimensional Fourier magnitude coefficients to capture the similarity between musical segments, yielding state-of-the-art results for the most recently published metrics (Nieto and Bello, 2014).

- A framework to compare, evaluate, and analyze musical structure algorithms that contains implementations of the standard and most competitive ones.
- A JSON-based format to store multiple annotations of various MIR tasks in a single file representing a music piece (Humphrey et al., 2014).
- Analysis of the robustness of multiple segment boundary annotations, exposing the problems of using ground-truths containing a single set of annotations per file.
- Four methods of merging multiple segment boundary annotations to alleviate the problem of subjectivity.
- Two metrics to evaluate flat and hierarchical merged boundaries.
- A perceptual evaluation of the F-measure (of the hit rate metric) to evaluate music boundaries that shows precision is more perceptually relevant than recall (Nieto et al., 2014).

5 Associated Publications by the Author

This thesis covers much of the work presented in the publications listed below.

5.1 Peer-Reviewed Articles

- Nieto, O., Humphrey, E. J., & Bello, J. P. (2012). Compressing Audio Recordings into Music Summaries. In Proc. of the 13th International Society for Music Information Retrieval Conference (pp. 313–318). Porto, Portugal.
- Nieto, O., & Jehan, T. (2013). Convex Non-Negative Matrix Factorization For Automatic Music Structure Identification. In Proc. of the 38th IEEE International Conference on Acoustics Speech and Signal Processing (pp. 236–240). Vancouver, Canada.

- Nieto, O., & Bello, J. P. (2014). Music Segment Similarity Using 2D-Fourier Magnitude Coefficients. In Proc. of the 39th IEEE International Conference on Acoustics Speech and Signal Processing (pp. 664–668). Florence, Italy. DOI.1109/ICASSP.2014.6853679.

- Nieto, O., & Farbood, M. M. (2014). Identifying Polyphonic Patterns From Audio Recordings Using Music Segmentation Techniques. In Proc. of the 15th International Society for Music Information Retrieval Conference (pp. 411–416). Taipei, Taiwan.

- Nieto, O., Farbood, M. M., Jehan, T., & Bello, J. P. (2014). Perceptual Analysis of the F-measure for Evaluating Section Boundaries in Music. In Proc. of the 15th International Society for Music Information Retrieval Conference (pp. 265–270). Taipei, Taiwan.

- Humphrey, E. J., Salamon, J., Nieto, O., Forsyth, J., Bittner, R. M., & Bello, J. P. (2014). JAMS: A JSON Annotated Music Specification for Reproducible MIR Research. In Proc. of the 15th International Society for Music Information Retrieval Conference (pp. 591–596). Taipei, Taiwan.

- Raffel, C., Mcfee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., & Ellis, D. P. W. (2014). mir_eval: A Transparent Implementation of Common MIR Metrics. In Proc. of the 15th International Society for Music Information Retrieval Conference (pp. 367–372). Taipei, Taiwan.

5.2 Algorithms Submitted to MIREX

- Nieto, O., & Farbood, M. (2013). MIREX 2013: Discovering Musical Patterns Using Audio Structural Segmentation Techniques. In Music Information Retrieval Evaluation eXchange. Curitiba, Brazil.

- Nieto, O., & Bello, J. P. (2014). MIREX 2014 Entry: 2D Fourier Magnitude Coefficients. In Music Information Retrieval Evaluation eXchange. Taipei, Taiwan.

- Nieto, O., & Jehan, T. (2014). MIREX 2014 Entry: Convex Nonnegative Matrix Factorization. In Music Information Retrieval Evaluation eXchange. Taipei, Taiwan.

- Nieto, O., & Farbood, M. M. (2014). MIREX 2014 Entry: Music Segmentation Techniques and Greedy Path Finder Algorithm to Discover Musical Patterns. In Music Information Retrieval Evaluation eXchange. Taipei, Taiwan.

CHAPTER II

REVIEW OF CURRENT APPROACHES AND EVALUATIONS

In this chapter the problem of Music Structure Analysis is reviewed by (i) framing it under the fields of Music Information Retrieval (MIR) and Music Perception and Cognition (MPC), (ii) providing a description of the standard processes to automatically capture the structure of a given musical piece from audio, and (iii) discussing existing evaluation metrics and approaches to establish reference *ground-truth* datasets that are as objective as possible (both in terms of metrics and references).

Despite the obvious overlap between MIR and MPC (i.e., they both explore music related questions), these research fields aim at solving different problems (Aucouturier and Bigand, 2012), and this becomes apparent when investigating the analysis of music structure from these two different perspectives. This chapter presents a discussion about the main differences between them in the form of a state-of-the-art review, including the most standard techniques to automatically discover structure in music. Background of common methods to extract musically meaningful features from audio (e.g., pitch class profiles, mel-frequency cepstral coefficients) is also reviewed, along with the basic tools to extract structure from audio signals. Finally, evaluation metrics to assess these algorithms are also included in this chapter, organized based on the music structure analysis task that they aim to assess: segment boundary identification, structural grouping, or pattern discovery.

1 Music Structure Analysis Review

Music is regarded as *hierarchically* organized sound, where the multiple layers of this hierarchy define the different qualities and aspects of music (Lerdahl and Jackendoff, 1983). Specific note events with a certain timbre quality, duration, and pitch are found at the bottom of this hierarchy, whilst as we climb up chords, rhythmic patterns, motives, phrases and large scale sections appear. These hierarchical aspects will define the overall structure of a specific piece.

Musicologists have been studying the structure of music much before computers could help in that regard (in fact, evidence of such analyses date back to the middle ages (Bent and Drabkin, 1987)). Nevertheless, in the early 20th century, Heinrich Schenker proposed a formal method to analyze tonal music hierarchically. This approach, known as the Schenkerian analysis, exposes the layered relationships between the pitches of music excerpts and draws structural conclusions based on these hierarchies. More recently, a theory that employs cognitive principles such as the Gestalt rules of psychology was introduced in (Lerdahl and Jackendoff, 1983). This publication, the Generative Theory of Tonal Music (GTTM), established the fundamentals for a deeper analysis of the structure of music, especially for its upcoming automatic approaches. Thanks to the drastic advances in computation, the field of MIR developed through the 90's with a sustained interest in music structure analysis, some of which influenced by the work of Lerdahl and Jackendoff. While MIR researchers tend to work on *how* to improve and implement algorithms in order to automatically discover the structure of a musical piece (e.g., many attempts to implement the GTTM in (Cambouropoulos, 2001; Hamanaka et al.,

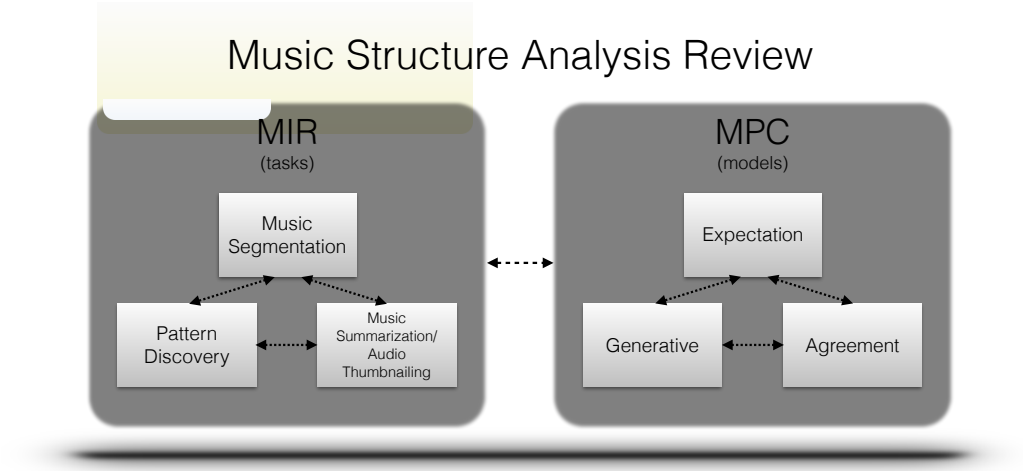


Figure 1: Diagram of the different topics covered in our review of music structure analysis.

2004)), MPC practitioners are more interested on *what* the specific parts of a given stimulus are, and *why* they trigger the correspondent behavior that allows us to perceive the structure of the piece. Therefore, work separates MIR into its different structural *tasks*, and MPC into its different *models* of musical structure.

As mentioned before, overlap between these fields exists, and sometimes a project could fall in many different areas of both MIR and MPC. Nevertheless, in the following subsections music structure analysis is explored from these two different perspectives in order to give an overview of this problem in its broadest sense. In Figure 1 a diagram shows the different topics that will be covered in this review.

1.1 Music Information Retrieval

The research community of MIR (also known as Music Informatics Research) usually treats the problem of music structure analysis from an engineering

perspective, where the ultimate goal is to be able to replicate algorithmically how humans analyze the structure of a given piece, rather than understanding how they are able to do it. Relatively large datasets (e.g., over 500 tracks) are used in order to develop algorithms that are not only as generic as possible, but also scalable to substantial amounts of data.

Given the easy access to massive digital music collections, this automatic identification of structure in music is a topic that has been widely investigated in MIR (Paulus et al., 2010). Even though its primary focus has been mainly the extraction of large scale sections, such as verse, chorus, bridge (a task that is also known as *music segmentation*), in this dissertation it will be discussed that the discovery of patterns and the generation of music summaries can also be interpreted as problems derived from music structure analysis.

1.1.1 Music Segmentation

This review begins with the most standard approach of the automatic analysis of the structure of music: music segmentation. Its main goal is to segment an audio signal representing a music piece in its different contiguous, non-overlapping sections (or segments), and then label these based on their acoustic similarity (e.g., ABAC). Thus, this task is usually divided into two different subproblems: segment boundary identification and structural grouping. Music segmentation has been evaluated in the MIR Evaluation eXchange (MIREX) competition since 2009, and a comprehensive analysis of this MIREX task can be found in (Smith and Chew, 2013).

Music segmentation is often regarded as an ill-defined problem, since it depends on many different aspects, some of which originate due to the subjective perception of the structure of music (Bruderer et al., 2009). Nevertheless,

and especially significant when attempting to narrow the ambiguity of this task, three types of principles have been identified to extract the different segments of a piece: *novelty*, *homogeneity*, and *repetition* (Paulus et al., 2010). The segments that can be identified using the novelty principle are those that start or end at a point in a given piece where one or more music dimensions (e.g., harmony, timbre) change drastically. The homogeneity segments are those that contain a musical aspect that remains constant across the whole segment (this can be seen as a different side of the same coin when compared with the novelty principle). Finally, the repetitive segments are those that can be identified due to their re-occurrences in a given piece, regardless of how novel/homogeneous they are on a smaller time scale. These principles will help to classify the music segmentation algorithms reviewed below.

The classic approach to identify boundaries is to apply a "checkerboard" kernel over the main diagonal of a self similarity matrix (SSM, which will be reviewed in depth in the next section) of certain music features, thus obtaining a novelty curve from which to extract the boundaries by identifying its more prominent peaks (Foote, 2000; Shiu et al., 2006; Mauch et al., 2009b). The size of this kernel defines the amount of previous and future features being taken into account, and it should be tuned depending on the music piece to be analyzed. This approximation uses both the novelty and homogeneous principles, depending on the preprocessing stage of the SSM computation. Other approaches of novelty and homogeneous-based algorithms include the usage of supervised learning (Turnbull et al., 2007) or variants of SSM also known as lag matrices (Goto, 2003), which indicate the amount of time elapsed (lag) between two time positions of the music piece. These lag matrices, depending on the design of the SSM, will also capture the repetitive segments. One of the

best performing techniques, which combines the three different segmentation principles, makes use of a custom representation called structural features, obtained by a simple rotation of the lag matrix, in order to produce a novelty curve from which to extract the segments (Serrà et al., 2014). Lately, an approach that uses linear discriminate analysis on top of these structural features has shown to obtain more precise results (McFee and Ellis, 2014b), while it has also been discussed that combining the structural features with the lag matrix techniques makes the results even better (Peeters and Bisot, 2014). Finally, deep learning using convolutional neural networks has been used with success in order to identify segment boundaries, even though this approach is limited to the extraction of the novel and homogeneous segments exclusively (Ullrich et al., 2014).

As for the structural grouping subtask, which can be viewed as an audio similarity problem, different methods have also been proposed: using Gaussian mixture models (Wang et al., 2011), a variant of nearest neighbor search (Schnitzer et al., 2011), and non-negative matrix factorization (NMF) (Kaiser and Sikora, 2010). Finally, other methods combine both tasks into one sole algorithm, e.g., using hidden Markov models (Levy and Sandler, 2008; Abdallah et al., 2005; Rhodes et al., 2006), a probabilistic fitness measure with a greedy search algorithm (Paulus and Klapuri, 2009), an NMF method to obtain both repetitive and homogeneous segments (Kaiser and Peeters, 2013), a probabilistic version of convolutive NMF (Weiss and Bello, 2011), k-means clustering (Peeters et al., 2002), more generic graphical models (Panagakis et al., 2011), or spectral clustering (McFee and Ellis, 2014a). These approaches tend to favor the identification of homogeneous segments, since it is common to aggregate the audio features for a given segment in order to capture their similarity. It is

worth noting that recent techniques such as (McFee and Ellis, 2014b,a) are also capable of discovering smaller segments such as riffs and motives, and therefore producing hierarchical outputs that may look more similar to the ones of the pattern discovery task that will be described in the next subsection. In this dissertation, a method to evaluate these hierarchies is proposed (see Chapter VI), however methods to visualize these more complex structures already exist, such as Paul Lamere's Infinite Jukebox*, Martin Wattenber's The Shape of Song†, or (Müller and Jiang, 2012; Nikrang et al., 2014).

To conclude with the music segmentation review, it might be interesting to see examples in which this task has been successfully applied to other problems. Given a music collection, the structure of a piece has been shown to be useful in order to identify and group all of its performances within the collection (Bello, 2009). The structure of a piece can be seen as a *fingerprint* so that it can be used as a query to identify similar performances in a given music collection (Grosche et al., 2012) (a further discussion about audio fingerprinting will be presented in subsection 1.1.3). Additionally, thanks to techniques in music segmentation, the automatic analysis of the sonata form can be significantly improved (Jiang and Müller, 2013), and also the identification of lyrics can be better performed (McVicar et al., 2014). Furthermore, the field of MIR could potentially help the field of MPC regarding the perception of the structure of a piece, and a good example is (Bimbot et al., 2012), where the authors propose a consistent communication language to describe the similarities and internal relationships within a music piece. Finally, variations of the

* `http://infinitejuke.com`

† `http://www.turbulence.org/Works/song`

methods described above can yield techniques to capture the most repetitive part(s) of an audio signal, which is related to the tasks of pattern discovery, music summarization and *audio thumbnailing*. These tasks will be reviewed next.

1.1.2 Pattern Discovery

As opposed to music segmentation, the task of discovering repetitive musical patterns (of which motives, themes, and repeated sections are all examples) consists of retrieving the most relevant musical ideas that repeat at least once within a specific piece (Janssen et al., 2013; Collins, 2013). Therefore, this task can be seen as an extension of music segmentation, with the following main differences: (i) a more granular level of detail is desired, (ii) the patterns found do not have to cover the entire piece, and (iii) overlap between these small segments is accepted. Even though this task has been explored in MIR for as many years as music segmentation, the first time it was evaluated in MIREX was in 2013, likely due to the difficulty of assessing it and having human annotated references that are accurate and rich enough.

Besides the relevant role this task plays in musicological studies, especially with regard to intra-opus analysis, it can also yield a better understanding of how composers write and how listeners interpret the underlying structure of music. Computational approaches to this task can dramatically simplify not only the analysis of a specific piece, but of an entire corpus, potentially offering interesting explorations and relations of patterns across works.

While in music segmentation digital audio is the usual form of input, typically the task of automatically discovering musical patterns uses symbolic representations of music (Collins et al., 2014b). Methods that assume a mono-

phonic representation have been proposed, and operate on various musical dimensions such as chromatic/diatonic pitch, rhythm, or contour (Lemström, 2000; Conklin and Anagnostopoulou, 2001; Lartillot, 2005, 2014). Other methods focusing on polyphonic music as input have also been presented, mostly using geometric representations in Euclidean space, with a different axis assigned to each musical dimension (Meredith, 2006; Forth, 2012; Collins et al., 2014a). The Hausdorff distance (Rockafellar et al., 1998), has been shown to be more aligned with human perception (Lorenzo and Maio, 2006), and interesting approaches have also been proposed using this distance (Typke, 2007; Romming and Selfridge-Field, 2007). Similar techniques that attempt to arrive at a compressed representation of an input, multidimensional point set have also been explored (Meredith, 2006; Forth and Wiggins, 2009; Meredith, 2013). Other methods using cognitively inspired rules with symbolic representations of music have also been proposed in (Forth, 2012; Nieto and Farbood, 2012). Working with the score of a musical piece instead of its audio representation can indeed reduce the complexity of the problem, however this also significantly narrows the applicability of the algorithm, since it is not necessarily common to have access to symbolic representations of music, particularly when working with genres such as jazz, rock, or Western popular music.

Methods using audio recordings as input have also been explored. A good recent example is (Collins et al., 2014b), where the authors first estimate the fundamental frequency from the audio in order to obtain the patterns using a symbolic-based approach. Another one uses a probabilistic approach to matrix factorization in order to learn the different parts of a western popular track in an unsupervised manner (Weiss and Bello, 2011). The algorithm can be tuned in order to obtain shorter repeated sections, which in turn become the

most salient riffs or patterns. The only algorithms working at the audio level submitted to MIREX have been presented by the author (Nieto and Farbood, 2013a, 2014b), and they will be discussed in Chapter III.

1.1.3 Music Summarization and Audio Fingerprinting

Music summarization is a much less explored task in MIR that aims at automatically generating summaries of pieces such that, for example, users navigating large collections are able to obtain a better understanding of the most representative parts of a given track. Therefore, audio, and not symbolic representations, is used as input to this type of task. Traditionally, the solution is to represent a full track with a single, identifiable excerpt. Known as audio thumbnailing, much effort has been invested into the development of automatic systems to this end; using structural analysis (Cooper and Foote, 2003; Shao et al., 2005; Müller et al., 2011), key phrases (Logan and Chu, 2000), segmentation by clustering (Peeters et al., 2002), timbral features (Meintanis and Shipman, 2008), or tempo tracking (Kim et al., 2006). Generally, it is agreed that music segmentation plays a big role in music summarization and audio thumbnailing, especially when focusing on the most repetitive segments of the piece.

Regardless, representing a full track with a single excerpt presents one unavoidable deficiency: the defining characteristics of a track are rarely concentrated in one specific section. In this dissertation an approach to music summarization that combines elements from music segmentation and pattern discovery is presented in the first part of Chapter III.

To conclude, there is no MIREX task available to evaluate music summarization, given the lack of actual metrics and, more importantly, datasets with

human references on which to compare the results. It is still unclear how these datasets should be designed for this type of task, one of the reasons being the important role of subjectivity. These perceptual problems are usually treated in the MPC field, which will be reviewed next under the framework of music structure analysis.

1.2 Music Perception and Cognition

In the field of MPC, the general focus is to develop theories regarding the mental and cognitive processes of musical events (Clarke, 1989), rather than automatizing these processes as reviewed in the field of MIR. Typically, practitioners of MPC design subject studies in order to formalize and evaluate cognitive models that would be helpful to understand the relations between aspects of musical stimuli and listeners or performers. Since these experiments are costly, it is preferred, as opposed to MIR, to work with less amounts of –sometimes unrealistic– audio data to keep the time of the experiment low while having as much control of the stimuli and the environment as possible. On the other hand, MPC usually deals with numerous subjects in order to test the generalization of these theories across some population. This contrasts with MIR, where reference datasets are commonly annotated by only one person per track, which becomes problematic when the task to evaluate is subjective enough as it will be discussed in Chapter VI.

Based on the cognitive models designed to better comprehend how humans understand music structure, the rest of this review is divided in three subsections: expectation, generative, and agreement models.

1.2.1 Models of Expectation

The most influential model of expectation in terms of cognitively grouping the structure of music is the implication-realization (IR) theory by Narmour (Narmour, 1992). This model defines the mechanisms of expectation by identifying two main principles: when a repetition in melody occurs, another repetition is expected; and when a drastic change in music appears, another one is expected. In order to identify the segment boundaries, the IR proposes to focus on the idea of *closure*, where the resolution to the tonic, the size of the intervals in the main melody, and the length of the notes play an important role.

The IR theory has been successfully tested (Krumhansl, 1996), even though more recently, and thanks to other listener studies, extensions have been proposed by adding external parameters (e.g., age), since they seem to also have a strong effect in the perception of structure and melody (Royal, 1995). These factors are apparently hard to be captured by an automated algorithm, therefore the MIR field tends to overlook this type of findings that suggest important differences in perception when assessing the structure of music, since they are not necessarily included on the audio signal.

A further model of expectation, which can be considered a descendant of IR, is Pearce's information dynamics of music (IDyOM) (Pearce, 2005). This probabilistic model aims at giving more emphasis to priors such as listeners' musical preferences, and it has been compared to rule-based generative models, where it performs similarly (Pearce et al., 2010).

Another important work on expectation, including theories about how humans comprehend the structure of music, is described by Huron (Huron, 2006). In his work the goals of expectation are discussed in relation to affect,

such as tension or resolution, reinforcing the fact that closure is relevant in the context of structural analysis. Huron also argues that the segments of a given track of popular music might contain multiple phrases that reoccur throughout the piece, which plays an important role in informing expectations while listening to this type of music.

Moving the focus on the identification of segment boundaries, human studies show that boundaries are not only perceived locally, but they are also expected when a specific change occurs in the music stimulus (Tillmann and Bigand, 2001). Moreover, it has been recently shown that it is possible to automatically learn the most salient perceptual features when segmenting a given melody, which yields state-of-the-art results when using techniques that are more MIR-oriented (Rodríguez-López et al., 2014). In the next subsection a review of models that use heuristic rules in order to define music structure-related theories will be presented. These models could potentially generate new valid music according to human perception.

Finally, a model of melodic expectation that employs both elements from IR and GTTM (discussed below) has also been presented (Margulis, 2005). In this case, the tonal pitch space used in IR, and the rule-based, bottom-up approach of GTTM are present in this model, which explicitly describes how expectation is connected to affect and tension, paying especial attention to the repeated notes and the hierarchy they form. Five main rules are defined in this model, some of them similar to those of GTTM: stability, proximity, direction, mobility, and hierarchy.

1.2.2 Generative Models

These models tend to be designed from a constructive (i.e., generative or bottom-up) perspective: how humans perceive each individual music event, and how, when grouped, larger hierarchical layers such as musical phrases or sections can be perceived. As mentioned in the beginning of this review, the GTTM publication, which uses this generative approach, is one of the most significant in terms of developing methods that would not only provide a better insight on how humans perceive and group structure (using Gestalt rules of perception), but also presents guidelines on how machines could potentially recognize them.

The main preference rules for grouping structure defined in GTTM are the following:

i **Proximity**: A segment boundary should be placed if the end of a slur or a rest occurs within a specific range of notes or if the time interval in the middle of the notes to be analyzed is large enough compared to the rest of the notes.

ii **Change**: Based on the degree of change of the register, dynamics, articulation, and/or length of the notes analyzed, this rule proposes to add a boundary between the two notes in which the degree of change is greatest. This rule is analogous to the novelty principle of the MIR task of music segmentation.

iii **Intensification**: If the rules i and ii are pronounced enough, a new hierarchical layer should be added on top of the respective notes.

iv **Symmetry**: Equal-length segments should have higher preference when segmenting music.

v **Parallelism**: If there are at least two segments that could be interpreted in a parallel manner, it is advised to form parallel parts of the groups they are forming. This is similar to the repetitive principle used in the MIR task of music segmentation previously described in this review.

vi **Time-span and Prolongational Stability**: Stable time-spans and/or *prolongational* reductions* are preferred, therefore grouping segments having this stability in mind is suggested.

Although previous efforts using Gestalt rules in music also exist (Tenney and Polansky, 1980), GTTM formalized some of these earlier theories into a single model that it is still considered valid today. Studies to corroborate GTTM are also available: it has been shown that it aligns well with the perception of music in general (Deliège et al., 1996), perception of temporal changes in music (Clarke and Krumhansl, 1990), perception of segment boundaries (Bruderer et al., 2006b), and the perception of musical patterns (Forth, 2012).

Algorithms stemming from GTTM have also been proposed, mostly based on the first two grouping rules of GTTM, which are the proximity and the change ones. The most relevant methods are the Local Boundary Detection Model (Cambouropoulos, 2001), which is a simplification of GTTM that focuses on the novelty principle of segmentation; the Grouper (Temperley, 2001), which also uses the parallelism preference rule and a more probabilistic

*The reader is referred to the original GTTM publication (Lerdahl and Jackendoff, 1983) to further investigate the concepts of time-span and prolongational reductions.

approach; and the Musicat (Nichols, 2012), which sophisticatedly uses GTTM rules with the aim of not only to analyze but also to write music automatically.

It remains to be seen how humans tend to agree/disagree on the perception of music structure, a problem that is usually approached from an MPC perspective. In the next and final section of this review an outline of the most relevant models of agreement is discussed.

1.2.3 Models of Agreement

It is commonly assumed in the field of MPC that music is generally ambiguous. This is sometimes the most fundamental problem regarding the implementation of algorithms that attempt to approximate the behavior of human perception when listening to music: people will have different subjective experiences when exposed to the same stimulus. This is generally not only due to previous exposition and familiarity of music, but also because of the *context* in which the music is heard (Krumhansl and Castellano, 1983). It is therefore a challenging task to capture these differences in perception, however various models have been proposed in order to better understand and address the problem of subjectivity.

An agreement exploration in terms of music similarity has been recently proposed (Flexer, 2014). They raise awareness in the difficulty of assessing the MIR task of "Audio Music Similarity and Retrieval," by performing a human experiment. An upper-bound of the performance of such systems is proposed, rather than a method to aggregate multiple annotations into a single one.

The perception of beats has also been assessed under the MIR task of "Beat Prediction" (Davies and Böck, 2014). In this case, and by the means of another human study, the authors demonstrate that most of the common

metrics to evaluate this task are non-informative. Moreover, they discuss that beats are perceived under a rather short time window, and therefore the actual metrics should use much shorter windows than the standard ones, since these ones may lead to misleading results.

Subjectivity plays also an important role in music emotion, where six main psychological mechanisms seem to be the base of the cognitive processes that occur when an emotion is perceived/evoked (Juslin and Västfjäll, 2008). Some known mechanisms are: brain stem reflex, evaluative conditioning, emotional contagion, visual imagery, episodic memory, and musical expectancy. Even though agreeing on emotion might be an unreachable —and probably an ill-posed— goal, the exploration of these mechanisms should yield a better form of unification of human responses on this task.

Finally, and focusing on the actual identification of segment boundaries, it has already been discussed in this review that humans do not tend to perceive them similarly (Bruderer et al., 2006b, 2009). Regardless, studies suggest that humans are able to identify the most *salient* boundaries in a given piece, therefore they tend to agree upon the degree of ambiguity of each boundary (Bruderer et al., 2006a). Additionally, these segment boundaries seem to be potentially defined by multiple listener responses (Livingstone et al., 2012). This model of salience agreement and the suggestion of producing more robust segment boundaries from subjects will be exploited in the merging of multiple boundaries annotations that will be performed in Chapter VI.

2 Current Approaches

Standard techniques to automatically discover structure from music are presented here. These methods will be used in the subsequent chapters, and will become especially relevant to successfully follow the description of the novel MIR approaches in Chapters III and IV. To start with, a review of the extraction of audio features from an audio signal is presented. These features become the initial input to the approaches described afterwards.

2.1 Feature Extraction

In this subsection the description of the audio features that will be used throughout this dissertation are presented. More specifically, two types of harmonic features (pitch class profiles and tonal centroids), and the standard features to capture timbre (Mel-frequency cepstral coefficients) are reviewed. Additionally, and by using previously computed (or annotated) beat information, these features can be aggregated at a beat level, thus reducing the amount of data and obtaining tempo invariant features, as it will be discussed at the end of this subsection.

To start with, we introduce the definition of the discrete short-time Fourier transform (STFT), as widely discussed in (Smith, 2010):

$$X_m(k) = \sum_{n=0}^{M-1} x(n + mH)w(n)e^{-j2\pi kn/N_{DFT}} \quad (1)$$

where x is the input signal, w is the analysis window, H is the hop size, M is the window size, N_{DFT} is the size of the discrete Fourier transform (DFT), $k \in [0 : N_{DFT}/2 - 1]$ is the frequency index, and $m \in [0 : N -$

1] is the time frame index. By taking the absolute value of the complex frequency domain representation $|X_m(k)|$ the magnitude spectrum of the signal is obtained. The features described below require this type of initial analysis for their computation.

2.1.1 Pitch Class Profiles

The standard set of features to capture the harmony of an audio signal are called pitch class profiles (PCPs, also known as Chroma features, or Chromagrams), and were initially introduced in (Fujishima, 1999) under the context of the MIR task of chord recognition. The main idea is to capture the amount of energy contribution of each of the twelve notes of the western scale (i.e., pitch classes) across a specific number of octaves of the magnitude spectrum of a windowed signal. This results in a twelve-dimensional profile for each frame of the STFT, and it is typically visualized as a $12 \times N$ matrix. Each of these feature vectors can be normalized such that its maximum is one, therefore removing any bias introduced by the loudness and/or noise of the signal. In Figure 2 an example is plotted.

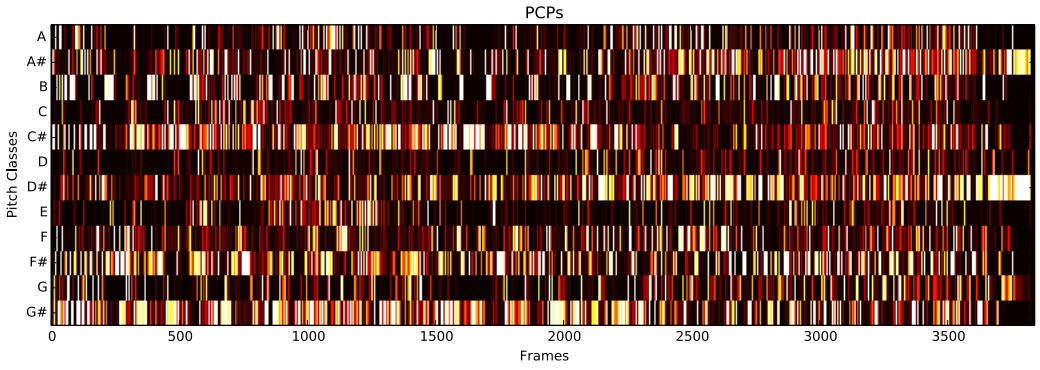


Figure 2: Example of the normalized PCP features on the track "Sweet Child O' Mine" by Guns N' Roses.

Formally:

$$PCP'_m(p) = \sum_{k=f_o}^{f_e} (\varphi(k) = p)|X_m(k)|^2 \tag{2}$$

where f_o and f_e are the frequency bins from which to start and end computing the PCPs, respectively, $p \in [0:11]$ is the pitch class index, and φ is a frequency mapping function defined as:

$$\varphi(k) = \left[12 \log_2 \left(\frac{f_s}{f_{ref}} \frac{k}{N_{DFT}} \right) \right] \mod 12 \tag{3}$$

where f_s is the sampling frequency, f_{ref} is the reference frequency for the first pitch class index $p = 0$, and $[\cdot]$ represents the *round* operator.

PCPs are usually normalized such that the maximum value for each vector is one. Formally:

$$PCP_m(p) = \frac{PCP'_m(p)}{\arg\max_\rho PCP'_m(\rho)} \tag{4}$$

These features are commonly used in the MIR task of music structure analysis (Paulus et al., 2010; Foote, 2000; Weiss and Bello, 2011; Kaiser and Sikora, 2010), and in this work the normalized version of these PCPs will be used. Additionally, harmonic PCPs (HPCPs), a variant of PCP that only consider the most relevant harmonic peaks of the spectrum (Gómez, 2006), will also be explored.

2.1.2 Tonal Centroids

Initially introduced by Euler (Euler, 1739), tonal centroids (or Tonnetz) are a geometric representation that describe the tonal space under the shape of

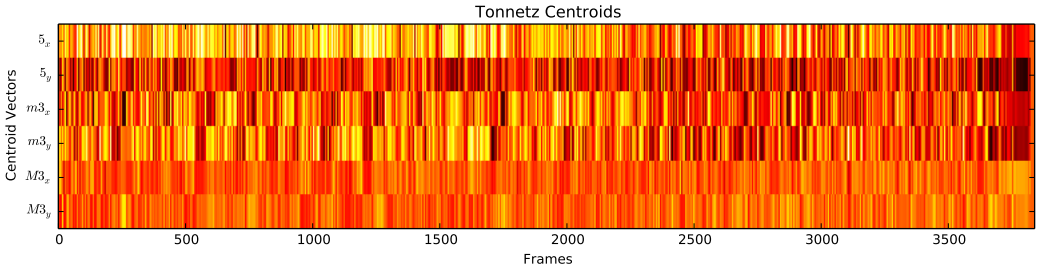


Figure 3: Example of the Tonnetz Centroids on the track "Sweet Child O' Mine" by Guns N' Roses.

a *torus**. From a given point of the torus representing a specific pitch, six possible surrounding relative pitches are found: perfect pitch, minor third, major third, and their three respective intervals on the other direction.

Recently, Tonnetz have been (re-)introduced in MIR by applying a series of geometric transformations on the PCPs to obtain six-dimensional vectors that can be used as harmonic features (Harte et al., 2006). These dimensions represent the three circles that the torus contains: the circle of fifths, the circle of minor thirds, and the circle of major thirds. An example of these features is plotted in Figure 3. Formally, the tonal centroids can be defined as:

$$\zeta_m(d) = \frac{1}{||PCP_m||_1} \sum_{p=0}^{11} \Phi(d,p) PCP_m(p) \tag{5}$$

where $d \in [0:5]$ represents the specific dimension of the three tonal circles (i.e., 5_x and 5_y for the two dimensions of the circle of fifths, $m3_x$ and $m3_y$ for the two dimensions of the circle of minor thirds, and $M3_x$ and $M3_y$ for the two dimensions of the circle of major thirds), $||\cdot||_1$ represents the $L1$ distance of a specific vector, and $\Phi \in \mathbb{R}^{6\times 12} = [\phi_0, \ldots, \phi_{11}]$ is the geometric transformation matrix defined as follows:

* A toroid in which the revolved figure is a circle.

$$\phi_p = \begin{bmatrix} \sin d\frac{7\pi}{6} \\ \cos d\frac{7\pi}{6} \\ \sin d\frac{3\pi}{2} \\ \cos d\frac{3\pi}{2} \\ 0.5 \sin d\frac{2\pi}{3} \\ 0.5 \cos d\frac{2\pi}{3} \end{bmatrix} \tag{6}$$

This representation has the advantage of being geometric, and therefore, and as opposed to PCPs, an interpolation between frames should be meaningful (Humphrey et al., 2012a). In this work an exploration of these tonal centroids will be presented in the context of musical structure analysis.

2.1.3 Mel-Frequency Cepstral Coefficients

Initially designed for speech recognition in (Mermelstein, 1976), and used later for music in (Logan, 2000), the Mel-frequency cepstral coefficients (MFCCs) are a reliable set of features to efficiently encode the spectral shape, which can therefore capture the timbre of a musical piece*. The Mel scale in which these features rely is designed to better align with human psychology (Stevens et al., 1937), which motivates its use to capture timbre in a more perceptually enhanced manner. For a visual example of MFCCs, the reader is referred to Figure 4.

MFCCs are obtained by following these steps:

1. Map frequencies of magnitude spectrum ($|X_m(k)|$) to the Mel scale using a filterbank design (e.g., overlapping triangular windows).

2. Sum the energy of each window and take its logarithm.

* Up to a point, given the apparent impossibility to define what timbre really *is*.

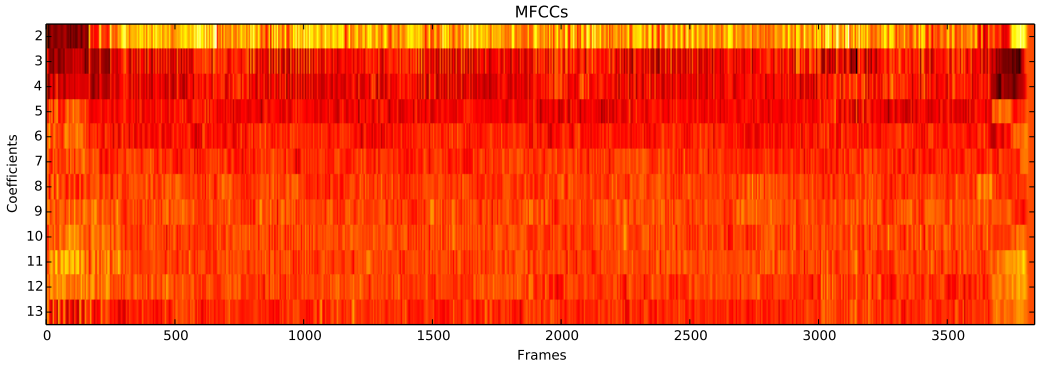


Figure 4: Example of the MFCC features on the track "Sweet Child O' Mine" by Guns N' Roses.

3. Apply a discrete cosine transform (DCT) to the log energies of the filter-bank.

4. Obtain the desired coefficients (typically the 2nd to the 13th) and discard the rest.

The Mel frequencies f_m can be obtained by mapping Hertz frequencies f_h following this formula:

$$f_m = 2595 \log_{10} \left(1 + \frac{f_h}{700} \right) \tag{7}$$

MFCCs are widely used in MIR, and they have been effectively applied to music segmentation (Levy and Sandler, 2008; Kaiser and Peeters, 2013). In this work MFCCs will also be investigated when presenting the automatic approaches to discover structure in music.

2.1.4 Beat-Synchronous Features

Given that music segmentation aims at identifying large-scale segments that usually start and end at a beat level, it is common to resample the frames of any

of the audio features presented above to specified beats. This greatly reduces the number of frames, leading to *beat-synchronous* features, which become the input to algorithms that can discover musical structure more efficiently in terms of computation time. Additionally, beat-synchronous representations are considered to be tempo agnostic, such that audio sequences can be compared in a normalized time scale. In Figure 5 an example of beat-synchronous PCPs is plotted.

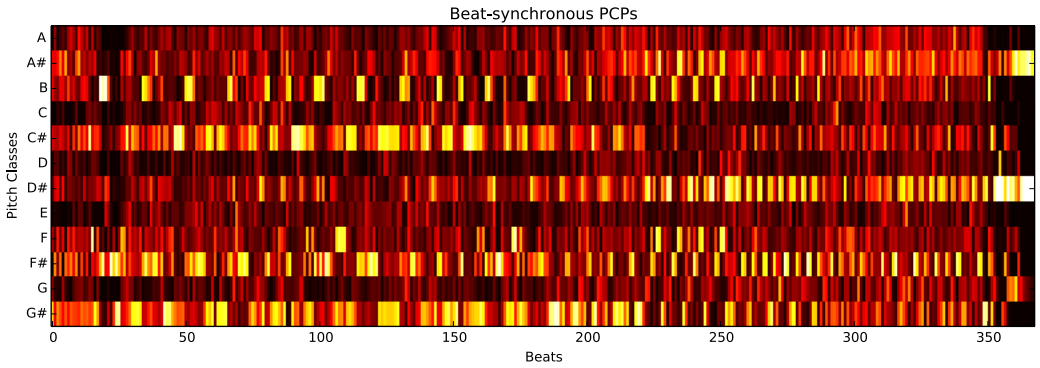


Figure 5: Example of the normalized beat-synchronous PCP features on the track "Sweet Child O' Mine" by Guns N' Roses.

To obtain the series of beats, automatic beat-trackers are usually employed (Zapata et al., 2013), and then, to resample the features, the different frames are averaged across the estimated beats (Ellis and Poliner, 2007). Depending on the application, resampling at a sub-beat level (e.g., at a fourth of a beat) can also be helpful as it will be discussed in the method to discover musical patterns in the next chapter.

Beat trackers, however, are far from being perfect (Holzapfel et al., 2012), and this can yield substantial problems when using this type of synchronous features. For example, if only half of the beats are tracked (which is a common beat-tracker mistake), the structure to be discovered might have noticeable

miss-alignments, with segments starting one or two beats after the expected time, or completely missing some of the segments.

Nevertheless, the improvement in terms of speed is significant, and after preliminary investigations of the impact of using estimated beats, and given the considerable number of algorithms to discover structure that were run for this dissertation, beat-synchronous features will be consistently used from now on.

2.2 Tools for Discovering Structure

2.2.1 Pre-Processing Features

It is common to enhance the audio features prior to the analysis of the structure of a given piece. To do so, two standard methods are employed: (i) enhancing the contrast of the features and (ii) aggregating multiple contiguous feature vectors. The first point is usually implemented as a power-law expansion of the features (e.g., taking the point-wise square operation on the whole feature matrix), such that the most prominent features are intensified and the parts with less energy are diminished, thus enhancing the features by removing possible noisy parts (Bertin-Mahieux and Ellis, 2012). The second point is achieved by running a filter across the time dimension of the audio features in order to aggregate the most similar parts that are contiguous which will produce more stable structural results, and also remove potential additional noise. These filters are usually standard mean or median filters (Cho and Bello, 2014). In the context of one of the algorithms presented in Chapter IV, it will be discussed that the median filter is a better candidate than a regular mean one.

2.2.2 Self-similarity Matrix

A standard tool to discover the structure of musical pieces is the self-similarity matrix (SSM). An SSM contains pair-wise comparisons of a given set of features using a specific distance measure d and stores the results in an $N \times N$ symmetric matrix S, such that $S(i, j)$ holds the amount of similarity between the features of the time indices i and j (which can represent beats, sub-beats, or frames, as discussed in subsection 2.1.4). In practice, an SSM can be useful to obtain an overview of the parts of a piece that recur at least once (at least for the features used for its computation). As an example, an SSM can be seen in Figure 6, where different segments of the song (plotted as vertical and horizontal lines from the human reference data) can already be visualized in the SSM.

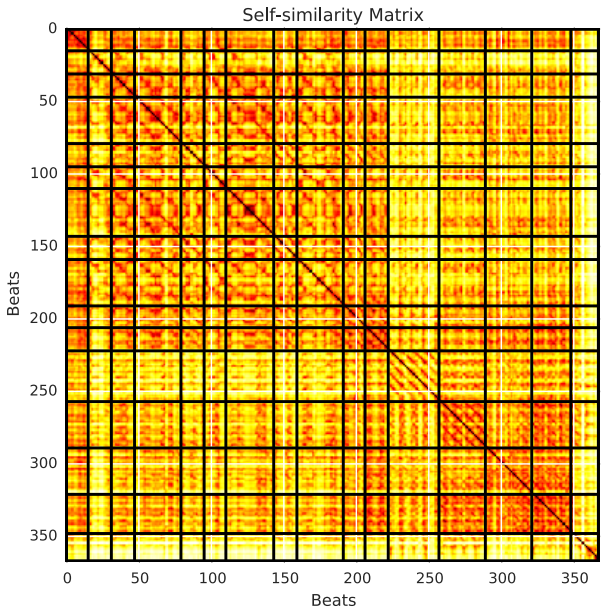


Figure 6: Example of an SSM computed from the normalized beat-synchronous PCP features of the track "Sweet Child O' Mine" by Guns N' Roses. The black lines represent the annotated large-scale segment boundaries found on the reference dataset.

An SSM is typically normalized such that its maximum value is 1, which, along with its symmetrical properties, can be characterized as $S(i,i) = 1$ and $S(i,j) = S(j,i)$, $\forall i, j \in [1:N]$. Formally, an SSM can be defined as:

$$S(i,j) = 1 - d(C_i, C_j) \tag{8}$$

where d is the distance metric and C is the feature vector used (e.g., PCPs, MFCCs). The choice of the distance metric d depends on the usage of the SSM, but the Euclidean distance is typically employed in music segmentation (Foote, 2000). Additionally, the correlation distance will be used in this dissertation, since it empirically yields good results for the algorithms presented in Chapter IV. The correlation distance is defined as follows:

$$d(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mu_{\mathbf{x}}) \cdot (\mathbf{y} - \mu_{\mathbf{y}})}{||\mathbf{x} - \mu_{\mathbf{x}}||_2 ||\mathbf{y} - \mu_{\mathbf{y}}||_2} \tag{9}$$

where $\mu_{\mathbf{x}}$ represents the mean of the vector $\mathbf{x}$, and $||\cdot||_2$ represents the L_2-norm. Essentially, an SSM is a specific instance of the more generic recurrence plots (Marwan et al., 2007), but using distances (or similarities) instead of binary values.

Furthermore, in the context of music segmentation, it is common to filter the matrix across its diagonal, such that repetitive structures become more apparent in the form of stripes or paths (Müller, 2007). These stripes will be perfectly diagonal as long as the features are tempo-invariant (e.g., beat-synchronous), otherwise, if tempo variations occur, these paths might have a *wobbly* form. A novel method to extract these repetitions under the context of musical pattern discovery will be presented in the next chapter.

Finally, a modified version of SSM called lag matrices, can be sometimes

used to extract the structure of a musical piece, as it was presented in (Goto, 2003).

2.2.3 Transposition-Invariant SSM

One of the main problems that arise when using harmonic features to compute the SSM is the potential challenge to capture repeated parts that are key transposed. Key transposition is a fairly common practice in western music, and a standard method to algorithmically capture this is the computation of the transposition-invariant SSM $\mathcal{S}$ (Müller, 2007). This technique can be described in two steps:

i. Compute twelve different SSMs from harmonic representations (e.g., PCPs), each corresponding to a transposition of the twelve pitches of the Western chromatic scale.

ii. Obtain the transposition-invariant SSM $\mathcal{S}$ by keeping the maximum similarity across the twelve matrices for all the $N \times N$ similarity coefficients in the output matrix.

Formally:

$$\mathcal{S}(i,j) = \underset{k\in[0:11]}{\arg\max}\{S_k(i,j)\}, \forall i,j \in [1:N] \tag{10}$$

where $\mathcal{S}$ is the transposition-invariant SSM, and S_k is the k-th transposition of the matrix S.

This method might introduce noise to the final matrix, and hence it is recommended to pre-process and diagonally filter the initial matrix S before

computing $\mathcal{S}$ (Müller, 2007). This technique will be further employed in this dissertation in the next chapter under the context of musical pattern discovery.

3 Current Evaluations

Once techniques that aim at automatically analyzing musical pieces have been designed, it is important to have rigorous scientific evaluations to assess and compare them as objectively as possible, which has always been a prevalent goal in MIR (Urbano et al., 2013). A significant effort towards having a unified framework to measure the effectiveness of these algorithms is the aforementioned evaluation exchange MIREX (Downie, 2008). Since 2005, multiple MIR tasks have been evaluated in this public platform that keeps growing every year, and that implements the most relevant evaluation metrics for each of the tasks*.

It is common to evaluate these tasks (including music segmentation or pattern discovery) with various human annotated datasets as reference (also known as *ground truth* data). These data usually contain one set of annotations for each of the tracks included in the dataset, and they are typically collected by multiple music experts under a controlled environment (Smith et al., 2011), even though other more engaging scenarios like games can also be used for certain tasks (Barrington et al., 2012). As previously mentioned, in Chapter VI it is shown that having only one reference for subjective tasks like music segmentation can become problematic due to the ambiguous choice of boundaries between subjects.

Regardless, in this last section of this chapter the most standard tech-

* As of 2014, there are 20 available tasks in MIREX.

niques to evaluate the identification of segment boundaries, the structural grouping of the segments, and the pattern discovery task are reviewed. The F-measure (or F_1 measure) will be presented first, which is a standard statistical tool that is typically used to compare an estimated result with an annotated dataset. As it will become apparent, the F-measure is used in all the evaluations reviewed in this section.

3.1 F-measure

When comparing an automatic technique with a reference dataset three properties are typically desired:

- Number of data points that have been correctly found (true values or "hits").
- Number of data points that have not been found (false negatives).
- Number of data points that have been incorrectly found (false positives).

The F-measure tries to quantify these three properties into one single real number between 0 and 1. The F-measure is the combination of two other real values: precision P and recall R. See Figure 7 for a visual interpretation of P and R.

P is the fraction between the amount of true values over the amount of values that the algorithm estimated (i.e., it is a real number between 0 and 1). As an example, imagine an algorithm that extracts only one boundary, and this boundary is correct. It will have a precision of 100%, but, since a song usually has more than one segment boundary, this does not mean that it

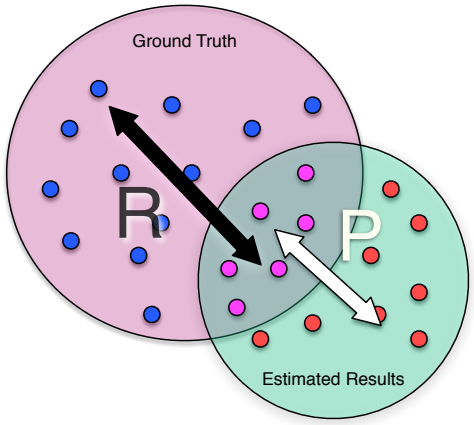


Figure 7: Visual interpretation of the Precision (P) and Recall (R) values. The "hits" are the small circles (pink) in the intersection between the ground-truth and the estimated results. The false negatives are marked in blue, while the false positives are marked in red. P is computed by dividing the number of hits over the number of estimated elements, and R is computed by dividing the number hits over the number of ground truth elements.

is a favorable algorithm (in this case there would probably be too many false negatives).

R is the fraction between the amount of true values over the amount of values in the ground truth (i.e., it is also a number between 0 and 1). To exemplify this, picture an algorithm that returns one boundary every second. This algorithm will likely have a recall value of 100% (it will depend on the time window size, which will be described in the boundary identification evaluation below), but that does not mean that it is a desired algorithm (it may have a lot of false positives, and these are not captured by the R value).

The F-measure combines these two values using the harmonic mean:

$$F = 2\frac{R \cdot P}{R + P} \tag{11}$$

The harmonic mean will make the F-measure penalize small values of R or P (e.g. if $R = 90$ and $P = 5$, then $F = 9$), thus the higher the F-measure, the higher both P and R will be, meaning that the more similar the estimated and the ground truth annotations will be (e.g. if $F = 100$, then $R = 100$ and $P = 100$, so the ground truth data is exactly the same as the estimated one).

In the next subsection it is reviewed how the F-measure can be used to automatically evaluate the tasks of boundary identification, structural grouping, and pattern discovery. In each of these scenarios, a "hit" (or a true value) will represent a different aspect of the specific task to evaluate. The metrics presented here are the ones used in MIREX, which are considered standard, even though some of them have often received criticism as it will also be discussed in this section. Typically, the evaluation of full datasets only include the average of each of the metrics across all the tracks of the set. Finally, `mir_eval` will be introduced, an open source package that contains the implementations for all of the MIREX metrics (Raffel et al., 2014),

3.2 Boundaries Evaluation

Two different metrics are usually employed to evaluate automatically estimated boundaries: the hit rate, and the median distances.

3.2.1 Hit Rate

The hit rate (also known as the F-measure for music boundaries) is the most standard metric for the evaluation of segment boundaries, which is performed by checking whether each estimated boundary falls within a specific time window from a ground truth boundary. This time window is usually 3 or 0.5 seconds long (these are the values used in MIREX). If the estimated bound-

ary falls within this time window of the ground truth, a hit is found. In this dissertation, the F-measure, precision, and recall values of this metric when using a T seconds window will be represented as $\mathbf{F}_T, \mathbf{P}_T$, and $\mathbf{R}_T$ respectively.

Additionally, the importance of removing the first and last boundaries has been discussed when performing the evaluation, since these boundaries should be trivial to retrieve. Therefore, they might bias the results (Nieto and Smith, 2013). This process is called *trimming*, and to indicate that a specific value of this metric has been computed using a trimmed version of the boundaries the t symbol will be added (e.g. $\mathbf{P}_{0.5t}$ represents the precision value of the hit rate metric using a 0.5 seconds time window and trimming).

3.2.2 Median Distances

The time deviations existing between the boundaries may also reflect the quality of the estimated boundaries, and they were first used in (Turnbull et al., 2007). They are calculated by obtaining the median distance in seconds from the two sets of boundaries to be compared. Consequently, two values are used to report this metric: the median distance from the estimated to the annotated boundaries $\mathbf{D}_{E2A}$, and the median distance from the annotated to the estimated ones $\mathbf{D}_{A2E}$.

This metric has been shown to strongly overlook outliers, which becomes problematic when evaluating tracks that contain a relatively high number of boundaries (Smith and Chew, 2013). Therefore, in this dissertation, and as it is common when reporting scores for new boundary algorithms, the evaluation of boundaries will be mostly based on the hit rate measure.

3.3 Structure Evaluation

The evaluation of structural grouping aims at quantifying the labels given to the segments based on their acoustic similarity. There are three standard metrics to assess this task: Pair-wise frame clustering, random clustering index, and normalized entropy scores.

3.3.1 Pair-wise Frame Clustering

Pair-wise frame clustering evaluation is a standard technique to assess clustering algorithms (the problem of structural grouping is, essentially, a clustering one). The idea, which was first used for this task in (Levy and Sandler, 2008), is to subdivide the segments at a certain level into frames (e.g., using a specific constant framerate or, alternatively, using beats). This is performed both for the estimated results and for the ground truth data (note that since the same framerate is used for both cases, the same amount of frames is considered for each case). Then, a pair-wise comparison is done for all the possible pairs. This results in two sets:

- P_e: The set of similarly labeled paired frames in a song according to the estimated results.
- P_a: The set of similarly labeled paired frames in a song according to the annotated results.

The intersection of these two sets (let us call it $P_{ea} = P_e \cap P_a$), will represent the correctly extracted segment similarities or hits. In this case, the precision value $\mathbf{Pw}_p$ will be the cardinal of the set P_{ea}, divided over the cardinal of the set P_a (therefore, a number between 0 and 1):

$$\mathbf{Pw}_p = \frac{|P_{ea}|}{|P_a|} \tag{12}$$

The recall value $\mathbf{Pw}_r$ will be the cardinal of the set P_{ea}, divided over the cardinal of the set P_e (also a number between 0 and 1).

$$\mathbf{Pw}_r = \frac{|P_{ea}|}{|P_e|} \tag{13}$$

In the rest of the dissertation, the pair-wise frame clustering F-measure, precision, and recall will be denoted by $\mathbf{Pw}_f, \mathbf{Pw}_p$ and $\mathbf{Pw}_r$, respectively.

3.3.2 Random Clustering Index

This index, also known as the Rand Index or RIC and originally introduced in (Hubert and Arabie, 1985), was reported for structural grouping of music in (Ehmann et al., 2011). It is similar to pair-wise frame clustering but in this case the *dissimilarities* are also considered. Using the same notation as in the pair-wise clustering, the RIC is computed as follows:

$$RIC = \frac{|P_{ea}| + |P_d|}{\binom{n}{2}} \tag{14}$$

where P_d is the set of all the dissimilar pairs of frames between the estimation and the annotation, and n is the number of frames in both the estimation and the annotation. Note that the coefficient $\binom{n}{2}$ (often read as "n choose 2") is the total number of pairs to be considered.

RIC is not usually reported in publications of structural grouping algorithms given its skewness with large tracks with many different labels (Smith and Chew, 2013; Lukashevich, 2008). In these cases, pair-wise frame clustering

is more reliable. Consequently, in this work this evaluation will not generally be reported.

3.3.3 Conditional Entropies

The last standard evaluation of the task of identifying audio similarity between segments of a song is the more recently proposed method based on the information-theoretic conditional entropy (Lukashevich, 2008). This aims at overcoming the problem of other methods (such as the pair-wise frame clustering, or the random clustering index described above) that tend to yield higher scores to tracks that contain fewer number of section types. For example, if a track has only two parts —e.g., "verse" and "chorus"— of equal duration, a random segmentation with only two section types could easily lead to a 50% of F-measure using pair-wise frame clustering. This score would decrease as the number of section types increases. This newer method based on conditional entropies is invariant to the number of section types, therefore it could be considered as superior in terms of comparing results across datasets of different number of section types.

The two different types of conditional entropies used in this method are defined as follows:

- $H(A|E)$: Amount of ground-truth (or annotated —A) information that is *missing* in the estimated result (E).
- $H(E|A)$: Amount of *incorrect* information found in the estimated result compared to the annotated one.

Note that these measures will be zero in the best case scenario: when the annotated results are the same as the estimated ones.

These measures are calculated by using the marginal distributions for the annotated and estimated structural segmentations along with their conditional distributions (for more technical information, formulations, and examples see the original publication (Lukashevich, 2008)). Once these conditional entropies have been computed, the final evaluation will be given by the over-segmentation score ($\mathbf{S}_o$) and the under-segmentation score ($\mathbf{S}_u$), computed as follows:

$$\mathbf{S}_o = 1 - \frac{H(E|A)}{log_2 N_e} \qquad \mathbf{S}_u = 1 - \frac{H(A|E)}{log_2 N_a}$$

These scores, like the F-measure, the precision and the recall values, are within the range of 0 and 1. The higher they are, the more similar the estimated results will be compared to the annotated ones. The lower they are, the more randomly chosen the estimated labels will be. Additionally, and to be consistent with the rest of the metrics, the F-measure between $\mathbf{S}_o$ and $\mathbf{S}_u$ will also be reported in this work, which will be denoted as $\mathbf{S}_f$.

3.4 Music Segmentation Evaluation Criticism

These presented methods for evaluating music segmentation are far from being perfect. The hit rate metric evaluation can be misleading, as Serrà et al. illustrated in (Serrà et al., 2012). In this publication, they compare their method for extracting boundaries with three different baselines using the F-measure on a standard dataset to compare estimated music segmentation results. These baselines are computed as follows:

- Baseline 1: Place a boundary every 17 seconds, which is the average time length of the boundaries in the dataset.

- Baseline 2: Place 8 boundaries in each song in a random way. 8 is the average number of boundaries per song in the dataset.

- Baseline 3: Place a boundary every 3 seconds, which is the amount of time to check around the annotations for a correctly placed boundary (therefore the recall value should be close to 100%).

The results are shown in Table 1. As it can be seen, an F-measure of more than 50% is reached with Baseline 3, which is very high considering that most of the publications report results for boundaries on the same dataset on a range of 50% to 75%, as it will become apparent in Chapter IV. This illustrates how these numbers can lead to the interpretation of bad results (e.g., baseline 3) as relatively good ones.

Method	$\mathbf{F}_3$	$\mathbf{P}_3$	$\mathbf{R}_3$
Baseline 1	40.3	38.7	43.8
Baseline 2	41.0	40.1	44.1
Baseline 3	50.5	34.7	99.8

Table 1: Boundary results for different baselines for The Beatles dataset as reported in (Serrà et al., 2012)

Another criticism of the boundary evaluation method is the time threshold that is traditionally used (±3 seconds). Even though the 0.5 second threshold is also used in MIREX, it is not common to find publications using this smaller threshold. It might be the case that 3 seconds is simply too long to capture high quality boundaries, and in this dissertation both 3 and 0.5 results are presented. In the task of beat-tracking, where a similar metric is also used, it has been shown that smaller time windows better align with perception (Davies and Böck, 2014). In Chapter VI the perception of hit measure

at 3 seconds will be evaluated, where it will become apparent that precision seems to play a more important role than recall.

On the other hand, the pair-wise frame clustering method used to evaluate audio similarity can also be misleading, but other methods like the conditional entropies make the results more robust, especially if both pair-wise frame clustering and conditional entropy results are reported (as in (Weiss and Bello, 2011)).

Finally, and as it was discussed when reviewing the MPC aspect of music structure analysis, one of the major problems when evaluating these tasks is the ambiguity in perception of the structure of music. Given that generally only one person per track annotates the reference ground-truth, biased results might be obtained due to the subjectivity problem. In Chapter VI a series of methods to merge multiple annotations per track will be presented in order to alleviate this apparent problem for the task of music segmentation.

3.5 Pattern Discovery Evaluation

As mentioned before, the task of pattern discovery appeared in MIREX in 2013 for the first time. Given its novelty, the metrics presented here are still at an early adoption stage, and only very recent publications include them. Nevertheless, and interestingly in contrast with the other two evaluations described above, the dataset to which this task is compared in MIREX contains multiple annotations from different music experts for each track. This should make it more robust to perceptual changes, even though further investigations should be carried out to assess the alignment of these metrics with our perception. Unfortunately, identifying the musical patterns by hand can be a much more daunting task than identifying the segments and label them accordingly,

and that is one of the reasons why only a small dataset of five musical pieces exists in order to evaluate pattern discovery algorithms*. It is likely that in the future this number will grow as the task is becoming increasingly popular.

Two main aspects of this task are evaluated: the patterns discovered and the occurrences of the identified patterns across the piece. Collins and Meredith proposed metrics to quantify these two aspects, which are detailed in (Collins, 2013); all of these metrics use the standard F_1 accuracy score, defined in Equation 11.

Establishment $\mathbf{F}_1$ Score (F_{est}): Determines how the annotated patterns are *established* by the estimated output. This measure returns a score of 1 if at least one occurrence of each pattern is discovered by the algorithm to be evaluated.

Occurrence $\mathbf{F}_1$ Score ($F_{\text{O}(c)}$): For all the patterns found, the goal is to estimate the ability of the algorithm to capture all of the occurrences of these patterns within the piece independently of how many different patterns the algorithm has identified. Therefore, this score would be 1 if the algorithm has only found one pattern with all the correct occurrences. A parameter c controls when a pattern is considered to have been discovered, and therefore whether it counts toward the occurrence scores. The higher the c, the smaller the tolerance. In this dissertation, as in MIREX, $c = .75$ and $c = .5$ are used.

Three-Layer $\mathbf{F}_1$ Score (F_3): This measure combines both the patterns established and the quality of their occurrences into a single score. It is computed using a three-step process that yields a score of 1 if a correct pattern has been found and all its occurrences have been correctly identified.

---

*The JKU Development dataset:
`https://dl.dropbox.com/u/11997856/JKU/JKUPDD-Aug2013.zip`

3.6 mir_eval

To conclude with the evaluation section, a new software package that was designed with the aim of simplifying the evaluation of any MIR task is discussed. MIREX is part of a larger infrastructure called Networked Environment for Music Analysis (NEMA) (West et al., 2010). The NEMA source contains many dependency to proprietary software (e.g., MATLAB), which makes it problematic when trying to install it in other machines by often requiring a time-consuming custom implementation of the metrics. This is one of the main motivations behind the new open source project called `mir_eval`.

`mir_eval`, which is a joint effort between Columbia University and New York University (Raffel et al., 2014), includes open source implementations in Python* for all of the available tasks in MIREX, including those described in this section. MIR and MPC researchers do not need to know the Python programming language to use `mir_eval`, since it comes with a series of scripts to easily produce the desired evaluations. The author of this dissertation contributed to `mir_eval` by implementing all the pattern discovery metrics and supervising the implementation of the music segmentation ones. Testing has demonstrated that this implementation of MIR task evaluations yields results comparable to the current MIREX evaluations†. All the metrics presented in this dissertation have been computed using `mir_eval`.

---

* The software can be downloaded from `https://github.com/craffel/mir_eval`.

† Some of these results are slightly different due to implementation and conceptual difficulties. The reader is referred to the original publication in order to see a discussion about it (Raffel et al., 2014).

4 Summary

The goal of this chapter was not only to set the necessary grounds for a successful reading of the rest of the dissertation, but also to motivate the work to be presented in the upcoming chapters. Starting with a review of the state-of-the-art, the fields of MIR and MPC were used to classify the various tasks and models, respectively, of the problem of music structure analysis. Important connections between the two fields were highlighted, which this dissertation aims to reinforce by centering the discussion of the next two chapters (III and IV) in the MIR field, and connecting it to the MPC field in the subsequent two (V and VI). Furthermore, the basics of feature extraction from audio were discussed in order to provide enough background to external readers, especially for the upcoming two chapters, where PCPs, MFCCs and Tonnetz will be treated as inputs for the proposed MIR algorithms. Finally, the standard evaluation metrics for the tasks of music segmentation and pattern discovery have also been reviewed, along with their main limitations and a novel transparent open source package that implements them. These metrics will be used throughout this work, and more perceptually inspired evaluations motivated by the limitations described above will be put forward in the last chapter of the main contributions.

Let us thus begin with new methods of identifying structure from an MIR perspective: two methods for music summarization and pattern discovery in Chapter III, and two methods for music segmentation in Chapter IV.

CHAPTER III

MIR METHODS: MUSIC SUMMARIES AND PATTERNS

1 Introduction

In this chapter the description of the main contributions of this dissertation begins under the context of MIR by focusing on two tasks that strive for the identification of the most prominent parts of a given audio signal: music summarization and pattern discovery. Ideally, music summaries contain a concatenation of non-overlapping segments of a music recording that best describe it, while the identification of repeated patterns yields the exact time points in which the most re-occurring (and possibly overlapping) sequences take place across the piece. Specifically, the proposed music summary defines a criterion that produces an audio summary which captures the most representative and less overlapping parts of an audio recording (Nieto et al., 2012). On the other hand, the second algorithm, which discovers repeated musical patterns in audio representations, uses a series of music segmentation principles and a novel greedy approach that yields state-of-the-art results (Nieto and Farbood, 2014a).

2 Audio Representation

These two algorithms use the same type of audio features as inputs. As is common in MIR research (see Chapter II) beat-synchronous PCPs are used to

capture the harmony of tracks in a tempo-agnostic manner. Additionally, tonal centroids (or Tonnetz features) are explored as an alternative set of harmonic features, which can be computed applying a set of geometric transformations to the PCP, as detailed in (Harte et al., 2006). Tonnetz can be useful when comparing distances between harmonic features, since they produce a geometric space where, as opposed to the probabilistic mass functions defined by the PCP, the euclidean distances become meaningful, as it has been reviewed in Chapter II.

For the methods presented in this chapter the author does not have access to open source implementations to other published algorithms (moreover, there is no standard method of evaluating music summaries, hence the uncertainty on how to compare our algorithm against similar ones), therefore a custom implementation is used to obtain the harmonic features. Specifically, a constant-Q transform is applied to an audio frame over the range of 110–1760 Hz with 12 bins per octave, producing a pitch vector X. The longest filter, which is set to 0.45 seconds, determines the length of the analysis window. A modified pitch vector Y is produced by standardizing the log-coefficients $log(\lambda X)$ and half-wave rectifying the result, as it is detailed in (Mauch and Dixon, 2010). The λ scale factor is heuristically set to 1000, but values within an order of magnitude in either direction produce similar results. By wrapping Y onto a single octave and scaling by the L_2 norm, the PCP features are derived. The Tonnetz are computed as described in (Harte et al., 2006).

2.1 Tracking the Beats

In order to compute the beats to synchronize them to the features described above, the recording is analyzed by a beat tracker adapted from (Grosche and

Müller, 2011). Constraints on the range of possible tempi that the system can track are imposed in the interest of mitigating double/half errors and producing consistent feature sequences across a variety of content. To do so, the periodicity analysis of the novelty function Δ_n is computed at N log_2 spaced frequencies per octave over a range of 1 to 8 Hz, producing the tempogram $\mathcal{T}$ as defined in (Grosche and Müller, 2011). This time-frequency representation is then wrapped to a single tempo octave of N bins and the most likely tempo path is extracted via the Viterbi decoder. In lieu of static transition probabilities, the transition probability matrix p_{trans} is defined as an identity matrix I of rank N convolved with a 1-D, 0-mean Gaussian window $\mathcal{N}$, where the standard deviation σ_n is parameterized by the relative amplitude of the maximum tempogram value as a function of time n, as follows:

$$p_{trans}[n] = I_N * \mathcal{N}\left(\mu = 0, \sigma_n = \frac{max(|\mathcal{T}[n]|)}{\mu_{|\mathcal{T}[n]|}}\right) \tag{15}$$

With this it is achieved the desirable effect of allowing the tempo estimator to adapt when the pulse strength is high, but resist change when the tempo becomes ambiguous. A histogram of the chord durations contained in publicly available chord annotations* is analyzed in order to find the best tempo octave to unwrap the path into. Having found that approximately 95% of the chord durations are greater than 0.5 seconds in duration, 2Hz is selected as a natural upper bound and map the optimal path through the single octave tempogram into the range of 60-120 BPM. Once this modification has been applied, the remainder of the implementation follows the reference algorithm.

*`https://github.com/tmc323/Chord-Annotations`

3 Summarizing Music Using a Criterion

One of the most direct applications of the automatic discovery of music structure is to generate an audible summary of a given track by concatenating the most relevant parts of the piece. This would significantly help, for example, when navigating massive music collections, since the user would not need to listen to an entire track in order to validate search results, but to a summarized version. Much effort has been invested towards this goal, mostly framed under the *audio thumbnailing* MIR task (Cooper and Foote, 2003; Shao et al., 2005; Peeters et al., 2002; Bartsch and Wakefield, 2005), which traditionally attempts to solve this problem by identifying the single excerpt that best represents the entire track (i.e., usually the chorus or the most repeated segment). These methods tend to perform well in identifying potential thumbnails for popular music that are particularly repetitive in nature. Regardless, there is an unavoidable deficiency when representing a full track with a single excerpt: the defining characteristics of a track are rarely concentrated in one specific segment, since this segment will hardly capture the most salient parts of the whole track.

In this section an alternative approach to classical audio thumbnailing is presented, where a short, audible summary, capturing representative parts of a track, as well as the most unique, is generated. More specifically, this section introduces a novel audio summary criterion and an efficient method of automatically generating these summaries from real music recordings. The criterion enforces that the chosen segments are maximally representative while having minimal overlap between them. Via examples and an experimental study it is shown how this measure yields successful audio summaries. Fur-

thermore, it is discussed that it is possible to automatically select the optimal number and length of the selected subsequences specific to a given recording, thus capturing the inherent structure of the audio track.

3.1 Feature Quantization

This approach is computationally demanding, which is why the features are quantized into a space of a finite number of discrete values, significantly reducing the computation time. The feature space is clustered using k-means and then performing vector quantization replacing each feature vector by its cluster's centroid. With computational efficiency in mind, the pairwise distances between centroids are pre-computed, thus accelerating the distance calculations between symbolic feature sequences as they will be needed in the algorithm. Even though larger values of k more faithfully reproduce the original features, the added computational load risks making the process intractable. By increasing k, distortion slowly decreases as a negative exponential while the size of the pairwise distance matrix grows quadratically. In the experiments, k is set to 50, 100, and 200.

3.2 Defining an Audio Summary Criterion

The main idea when producing an audio summary of a music track is to retain the minimum number of distinct parts that best describe it, thus exploiting the fundamental characteristics of structure and repetition inherent in any musical work. Consequently, a good summary criterion actually synthesizes two opposing notions: to keep as much information as possible, while avoiding overlap between chosen parts. A summary is defined as the set $\Gamma = [\gamma_1^N, \ldots, \gamma_P^N]$ of P,

N-length subsequences that maximizes a function Θ over a feature sequence $\mathbf{S}$ of length M, where $\exists m$ s.t. $s_m^N = \gamma_i^N, m \in [1:M]$, $s_m^N \in \mathbf{S}$, and $i \in [1:P]$.

3.2.1 Compression Measure

Since our goal is to describe a sequence in terms of itself with a minimal loss of information, this can be framed as a data compression problem. Building upon this idea, a compression measure $\mathcal{C}(\Gamma|\mathbf{S})$ that quantifies the extent to which Γ explains a given $\mathbf{S}$ is defined as follows:

$$\mathcal{C}(\Gamma|\mathbf{S}) = 1 - \frac{1}{PJ}\sum_{i=1}^{P}\sum_{m=1}^{J} ||\gamma_i^N, s_m^N||_2 \tag{16}$$

This measure can be interpreted as a normalized, convolutive Euclidean distance, such that there are $J = M-N+1$ element-wise comparisons between a given N-length subsequence γ_i^N and all J N-length subsequences $s_m^N \in \mathbf{S}$. All distances, taken directly from the precomputed pairwise matrix discussed in subsection 3.1, are then averaged over the J rotations and P subsequences in Γ. Intuitively, the compression measure equals 1 when $\Gamma = \mathbf{S}$ and 0 when $\Gamma \not\subseteq \mathbf{S}$.

3.2.2 Disjoint Information Measure

Besides determining how well Γ describes $\mathbf{S}$, it is needed to measure the amount of information shared between each pair of subsequences in a set. Conversely, a disjoint information measure $\mathcal{I}(\Gamma)$ that seeks to quantify the uniqueness of each subsequence in Γ relative to the rest is introduced as follows:

$$\mathcal{I}(\Gamma) = \left(\prod_{i=1}^{P}\prod_{j=i+1}^{P} D_{min}(\phi(\gamma_i^N), \phi(\gamma_j^N))\right)^{\frac{2}{P(P-1)}} \tag{17}$$

Shift-invariance in time is achieved by mapping a sequence of features γ_i^N to a sequence of *shingles* ρ_i^K with length $K = N - L + 1$ where a shingle is defined as the stacking of L adjacent feature frames into a single feature vector (for a more detailed discussion about shingles, the reader is referred to (Casey et al., 2008a)). The function ϕ returns the *shingled* version of a subsequence. A modified Euclidean distance function D_{min} then measures the intersection between sequences of shingles, returning the average minimum distance between the u^{th} shingle in ρ_i^K and all v shingles in a different subsequence ρ_j^K, defined as follows:

$$D_{min}(\rho_i^K, \rho_j^K) = \sqrt{\sum_{u=1}^{K} \arg\min_{v} (\rho_i[u] - \rho_j[v])^2} \tag{18}$$

Two important subtleties should be observed when calculating this measure. First, distances between shingles are defined by the element-wise L_2 norm based on the same pairwise distance matrix as before. Additionally, $\mathcal{I}(\Gamma)$ is a geometric mean and only produces large values when all pairwise distances are also large; any small distance in the product forces the overall measure toward zero.

3.2.3 Criterion Definition and Calculation

Inspired by the F_1-measure (described in Equation 11 of Chapter II), and having established measures of compression and disjoint information for some Γ, both of these traits are captured by defining a single criterion Θ as follows:

$$\Theta(\mathcal{C}, \mathcal{I}) = \frac{2\mathcal{C}\mathcal{I}}{\mathcal{C} + \mathcal{I}} \tag{19}$$

It is important to note that $\mathcal{C}$ and $\mathcal{I}$ are constrained on the interval [0,1] and converge to one when optimal, therefore computing the criterion as a harmonic mean enforces the behavior that its value is only large when both measures are as well.

At this point it is worthwhile to make the observation that, from a theoretical point of view, this criterion can be evaluated at every unique combination of subsequences Γ over an entire sequence $\mathbf{S}$. The output of this exhaustive calculation is a P dimensional tensor where each axis is of length J, and the best summary is given simply by the $argmax$ of the resulting data structure. From here onward, the term *optimal criterion* Θ_{max} is used to refer to the absolute maximum of this tensor, as would be found through a naive, exhaustive search of the space. Note that for large J and P however, evaluating every cell in this tensor becomes computationally intractable and efficient approximations are necessary. A heuristic approximation will be presented in section 3.3.

3.2.4 Case Example

In this subsection the behavior of the audio summary criterion is illustrated by analyzing the first half of Frédéric Chopin's Mazurka Op. 30 No. 2, which exhibits a well-defined AB structure. For the sake of demonstration, a subsequence length of $N = 8$ is selected and $P = 2$ is defined such that an exhaustive evaluation of Θ produces an easy to visualize $J \times J$ matrix. In Figure 8, the result of computing $\mathcal{C}$, $\mathcal{I}$ and Θ over all pairs of subsequences is shown.

In the left-most matrix of Figure 8 the compression measure $\mathcal{C}$ is shown. This measure quantifies the extent to which a set Γ explains the overall track independent of any correlation between subsequences. The optimal $\mathcal{C}$ in this

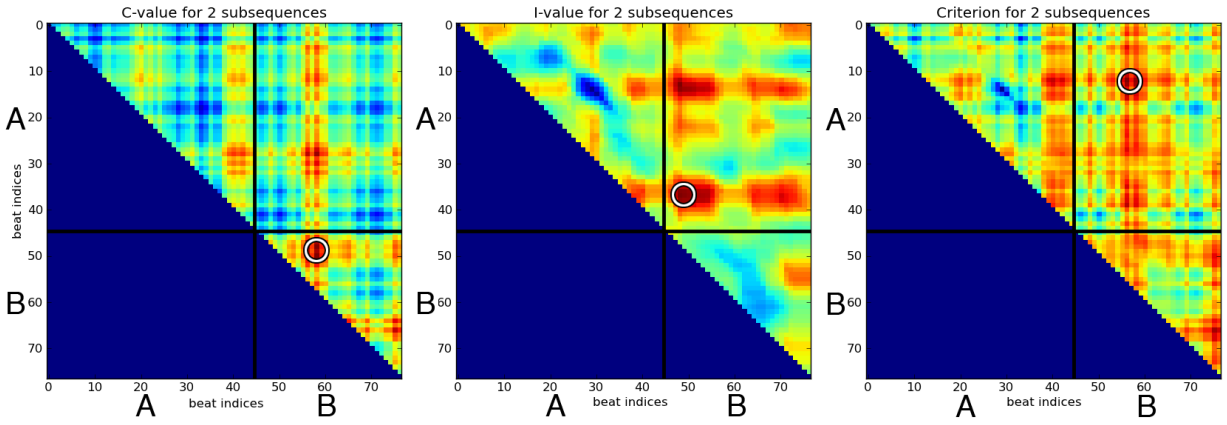


Figure 8: Search space for $\mathcal{C}$, $\mathcal{I}$ and Θ (left, middle, and right respectively) for $P = 2$ subsequences in the first half of a performance of the Mazurka Op. 30 No. 2. Black lines split part A and B. Circles mark the maximum value. Each position in the matrices correspond to a 8-beat subsequence.

matrix corresponds to the two subsequences at beat indices (48, 59) in the B-B quadrant. These subsequences correspond to repetitions of the same part, making the information in Γ redundant.

The disjoint information measure $\mathcal{I}$ is depicted in the center matrix of Figure 8. This measure captures the degree of uniqueness between subsequences in Γ. As it can be seen, the measure behaves as expected: repeated subsequences in the same section (in quadrants A-A or B-B) produce significantly lower values of $\mathcal{I}$ than subsequence pairs in A-B, where the highest $\mathcal{I}$ is found.

Lastly, the criterion Θ is obtained by combining the previous two matrices, and is depicted in the last matrix of Figure 8. In the example the maximum value of $\mathcal{C}$ corresponds to repetitions of the same part, thus making $\mathcal{I}$ to be small and forcing the overall Θ to also be small. Similarly, the position of the maximum value of $\mathcal{I}$ at the boundary between A and B results in a low $\mathcal{C}$ value, again producing a smaller Θ. In this example it becomes apparent that

the expected is obtained: Θ is maximized by the combination of subsequences in A,B that best balance the two criteria by capturing the middle sections of each part.

3.3 Heuristically Approximating the Optimal Solution

In certain scenarios, a naive calculation of the optimal criterion can become computationally inefficient, impractical, or even impossible with today's technology. To be concise, an exhaustive evaluation and parallel search of the full Θ tensor of size $(J/2)^P$ would result in an algorithm of exponential complexity $\mathcal{O}((JN \log J)^P)$. A much faster implementation based in heuristic principles that approximates the optimal solution is presented in this subsection.

The basic idea behind the fast approach is to assume that the most relevant parts of a song will most likely be uniformly spread across time. Here this heuristic approach is described with the support of the pseudocode found in Algorithm 1. The method *EquallySpaced()* initializes all P subsequences into equally spaced time indices and stores them in the array Υ. Then the algorithm iterates over the P subsequences, fixing all of them except the P_i being processed. A sliding window is used, operating over the region between the endpoint of the previous subsequence and the start of the next one, to find the best local music criterion θ by calling the function *ComputeCriterion()*. The sliding window must be within the correct bounds, and this is checked with the method *CheckBounds()* at every iteration, and if it is, the best index υ in Υ is updated. Finally, the summary Γ is obtained by concatenating the subsequences at the time indices in Υ. This operation is done inside the method *GetSubseqsFromTimeIdxs()*.

This results in a linear algorithm with respect to P, with a time com-

Algorithm 1 Heuristic Approach

Require: $\mathbf{S} = \{s_1, \ldots, s_M\}, P, N$
Ensure: $\Gamma = \{\gamma_1^N, \ldots, \gamma_P^N\}$
  $\Upsilon \leftarrow \text{EquallySpaced}(\mathbf{S}, P, N)$
  for $i = 1 \rightarrow P$ **do**
    $\theta \leftarrow 0$
    for $j = 1 \rightarrow M$ **do**
      if CheckBounds(Υ) **then**
        $\Theta \leftarrow \text{ComputeCriterion}(\mathbf{S}, \Upsilon, N, P)$
        if $\Theta > \theta$ **then**
          $\theta \leftarrow \Theta$; $\upsilon \leftarrow j$
        end if
        $\Upsilon[i] \leftarrow j$
      end if
    end for
    $\Upsilon[i] \leftarrow \upsilon$
  end for
  $\Gamma \leftarrow \text{GetSubseqsFromTimeIdxs}(\mathbf{S}, \Upsilon)$
  return Γ

plexity of $\mathcal{O}(PMJ)$. This approach dramatically improves efficiency, allowing the exploration of different hyperparameter values of P and N, as it will be described in subsection 3.5.1.

3.4 Evaluation

In this section, the following evaluations of the audio summary criterion are performed:

i Efficiency of the beat-tracker, quality of the harmonic features, and exploration of different dictionary sizes.

ii Analysis of the heuristic approach, comparing it with the exhaustive search and random selection.

iii Discovery of the optimal combination of the hyper-parameters P and N by using the heuristic approach.

Before describing the three different experiments in depth, the dataset used for evaluation is described.

3.4.1 Methodology

A collection of solo piano music compiled by the Mazurka Project* is employed, comprised of 2,914 tracks corresponding to different recorded performances of 49 Mazurkas. In order to avoid confusion, the terms *piece* or *work* are used when referring to a Mazurka, and *track* or *performance* are reserved to describe an instance of the work as audio. Thanks to this dataset the several performances of a single work can be leveraged to measure the consistency of our criterion, since many performances for each musical work are available. Additionally, this collection contains 301 tracks with human-annotated, ground-truth beat times, which allows the evaluation of the impact of beat tracking on various dimensions of performance. It also provides the added benefit that Chopin's Mazurkas are notoriously difficult to beat-track via automatic approaches (Grosche and Müller, 2011). Therefore, if there is marginal discrepancy between summaries using ground truth beat annotations and summaries built upon estimated beat times, then the work presented here does not wholly depend on the accuracy of the beat tracking algorithm.

*`http://www.mazurka.org.uk`

3.4.2 Parameter Sweep and Selection

An experiment is designed to sweep across the range of free parameters in order to select a feature space with which to proceed, aiming to identify the optimal configuration. There are three questions to address: (i) is automatic beat tracking sufficient? (ii) Do PCP and Tonnetz features perform equivalently? (iii) Does performance vary significantly as a function of codebook size?

These three decisions can be resolved by observing how the optimal criterion behaves across various performances of the same work, comparing between ground truth and estimated beat annotations. A satisfactory audio summary of the same piece would persist across multiple recorded versions —at least intuitively—, so the summaries themselves should be substantially similar.

In order to objectively validate this approach, the tracks of the 301 recordings with ground truth beat annotations are stratified into five folds for cross validation such that all but one are used to train the quantizer and the remaining hold-out is reserved as a test set. Sweeping across the two beat annotation sources (ground truth, automatic), the type of harmonic features (PCP and Tonnetz), and three codebook sizes (50, 100, 200) produces 12 possible feature space configurations (see Table 2). Summary sets Γ are identified by exhaustively computing Θ_{max} over all possible combinations of subsequences, where segment length N and number P are fixed at 16 and 4, respectively. Moreover, a stride parameter of $N/2$, analogous to a hop size in frame based audio processing, is applied to make the exhaustive search more computationally tractable.

The intra-class distance measures the degree to which summaries of the same work are close together, while the inter-class distance captures the dis-

K	Beats	PCP-F_{ratio}	Tonnetz-F_{ratio}
50	Ground-Truth	3.64	3.97
100	Ground-Truth	3.84	4.29
200	Ground-Truth	4.09	**4.74**
50	Estimated	2.71	3.89
100	Estimated	2.68	4.20
200	Estimated	2.87	**4.45**

Table 2: Parameter Sweep, where K is the codebook size.

tance between dissimilar works. Pairwise distances between summaries of tracks in each fold are computed and the values are treated as empirical distributions of these two classes. The Fisher ratio, defined below, provides an estimate of the separation between intra- and inter-class summary distances.

$$F_{ratio} = \frac{\mu_{intra} - \mu_{inter}}{\sigma^2_{intra} + \sigma^2_{inter}} \tag{20}$$

Note that higher values of F_{ratio} indicate distinct, well-localized distributions where 'similar' items cluster together, and translates to more consistency across performances. Table 2 shows the results of sweeping free parameters in the feature space. There are a few important observations to make about these results. First, a Tonnetz representation produces consistently better results than chroma features. Additionally, the estimated beats strongly influences the PCP results, making them considerably worse than those using human annotated beats. Nevertheless, Tonnetz features computed from automatically extracted beat times only marginally trail their ground truth equivalent. Moreover, the codebook size k has a non-trivial impact on performance and

is positively correlated. Therefore, we can conclude that Tonnetz-features computed with a beat tracking front-end are the best choice going forward, and that the parameter k should be large and ultimately based on practical limitations of the implementation.

3.5 Evaluation of the Heuristic Approximation

The performance of the heuristic approach is evaluated by comparing the summaries it produces with the optimal solution obtained through exhaustive computation. A second comparison is made with the expected performance of a random algorithm, obtained by averaging across all results observed in the course of computing Θ_{max}. Therefore, the upper (max) and lower (random) bounds of performance can be established, allowing to determine where on this continuum our heuristic solution lives. The discrepancy is measured between the optimal Θ_{max}, random Θ_{rand}, and heuristic Θ_{heur} solutions by computing the averaged Mean-Squared Error (MSE) across all tracks in the full dataset. To account for local variance resulting for a given track, the range of $\mathbf{\Theta}$ is normalized such that $\Theta_{max} = 1$ and $\Theta_{min} = 0$. The normalized MSE can be expressed formally as follows:

$$\text{MSE}(\mathbf{\Theta}) = \frac{1}{\mathcal{S}} \sum_{i}^{\mathcal{S}} (1 - \mathbf{\Theta}_i)^2 \tag{21}$$

In this context, a normalized Θ_{max} always equals 1, $\mathbf{\Theta}$ represents a vector of normalized criteria obtained by some search strategy, and $\mathcal{S}$ is the number of songs in the Mazurka data set.

In Table 3 the results of the MSE setting the hyper-parameters to $P = 4$ and $N = 16$ are shown. The MSE of the random baseline is approximately

$\boldsymbol{\Theta}$	MSE($\boldsymbol{\Theta}$) (in %)
Θ_{max}	0.00
Θ_{heur}	1.12
Θ_{rand}	21.01

Table 3: Evaluating the heuristic approach Θ_{heur} using the Mean-Squared Error to compare it against the brute force approach (Θ_{max}) and random (Θ_{rand}).

21%, whereas our heuristic approximation is nearly two orders of magnitude better, achieving a MSE of slightly over 1%. It is evident from this contrast that the heuristic search very closely approximates the results of exhaustive computation, significantly outperforming the random baseline. Therefore the preliminary conclusion that the heuristic approach is a sufficient approximation can be claimed. This allows a more thorough exploration over the space of hyper-parameters.

3.5.1 Automatically Selecting Hyper-parameters

In lieu of automatically discovering the inherent large-scale structure of the musical piece (i.e., learn P and N), the maximum result of the heuristic approach Θ_{heur} can be used for different combinations of P and N now that the efficiency to perform a search across these hyper-parameters has been gained. In this experiment 9 pairs of $P \in [2:5]$ and $N \in [16:64]$ are explored (constraining N to powers of two), avoiding (P, N) combinations such as $(5, 64)$ or $(2, 16)$ that would produce summaries that are too long or short, respectively. These ranges incorporate prior musical knowledge, as there are typically a small number of distinct parts in a work and meter is predominantly binary.

Note that, since the best choice of P and N is signal-dependent, in reality there is no universally optimal combination for all music.

Theoretically, if high values of P and short values of N were used, motivic elements would likely be identified in the music piece. However, this would result in a computationally expensive method that would favor non-overlapping motives or repeated patterns. The other algorithm detailed in this chapter aims at identifying possibly overlapping repeated patterns following a much more efficient approach, as it will be seen in Section 4.

Since the structure and meter are generally invariant to interpretation, the combination of P and N that yields Θ_{heur} for a given track provides another statistic that should persist across multiple performances of the same work. The criterion is further evaluated by measuring consistency of the optimal (P, N) pair using the entire Mazurka dataset, and providing qualitative examples of the observed behavior.

3.5.2 Quantitative Evaluation

In Figure 9 a consistency distribution resulting from a sweep across combinations of P and N is given. The proportion of performances for a given Mazurka that produces the most frequent (P, N) pair at Θ_{heur}, where a value of 1 indicates complete agreement and 0 complete disagreement, is represented by the x-axis. Moreover, the y-axis represents the number of works that produce a given consistency value, and there are 49 in total.

As illustrated by the plot, there is high consistency ($\geq 90\%$) for more than half of the data set, resulting in an average consistency of 87%. This shows that the criterion is able to capture high-level information about the structure of a work across various performances, validating its capacity to

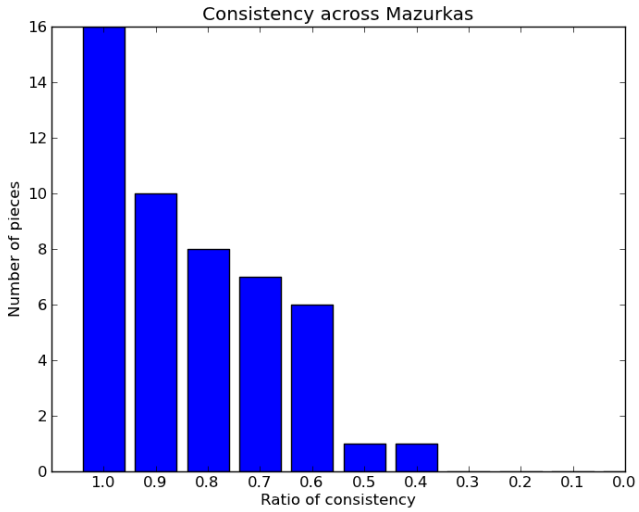


Figure 9: Evaluating consistency across different performances of the same song for the entire Mazurka data-set

produce informative audio summaries. Despite a high average overall, it is of special interest to qualitatively analyze the Mazurkas that yield different optimal configurations of the hyper-parameters.

3.5.3 Qualitative Evaluation

Note that Figure 9 fails to capture the degree of contrast between Θ_{heur} and values for other combinations of P and N. Some Mazurkas have an ambiguous form: depending on how deep the analysis of the piece is, it could result into having different subparts that one might want the audio summary to capture. Therefore, it is sometimes interesting to fix P depending on the depth of the analysis: high P and low N to obtain more parts of a small length; or low P and high N to obtain fewer parts of the track but with a longer subsequence length. Upon closer inspection, the structure of some works might not be clearly defined leading to multiple, equally reasonable interpretations, i.e.,

more than one (P, N) with large Θ values. One such instance of multiple interpretations occurs for Op. 7 No. 2. The form of this work is $ABCA$, but —depending on performance— parts B and C can be interpreted as one longer part, resulting in an ABA structure. Consequently, 62% of these performances produced a Θ_{heur} for $P = 2$, while 31% of performances occurred at $P = 3$.

The other primary cause of inconsistency seems to be due to tempo modulations and the resulting errors and artifacts caused by the beat tracker. An example of this is Op. 41 No. 1, producing the lowest consistency ratio of 49%. In this track a lack of well-defined onsets and expressive rhythmic interpretations are observed, both within and between performances. This causes the beat tracker to behave erratically, producing misaligned feature sequences that ultimately yield Θ_{heur} values for different pairs of (P, N).

On the other hand, Op. 24 No. 3, which exhibits a clear ABA structure and a more stable tempo, achieves 100% consistency for $P = 2$ and $N = 32$. The more noteworthy observation though is that this particular piece is in a ternary meter. Therefore better summaries would likely be obtained with N being a power of 3, and exploring other values of N could potentially improve consistency.

3.6 Discussion on Tonnetz

In these experiments Tonnetz features yielded significantly better results than PCPs, particularly in the absence of good beat information. This finding, by itself, warrants discussion. One possible explanation is that, as Tonnetz features live in a continuous-valued geometric space, any beat estimation errors result in a smooth interpolation of the feature space. PCP features, which act as a time-varying probability distribution, cannot resolve timing errors in the

same way. As a result, a beat tracker does not need to be perfect if given a suitable feature representation.

Tonnetz will also be explored in the next method as an alternative harmonic representation, and see if it has the same impact when applied to a different MIR structure-related task, in which tracking beats becomes trivial.

4 Identifying Repeated Musical Patterns

In this section a method for discovering patterns of note collections that repeatedly occur in a piece of music is presented, as it was originally published in (Nieto and Farbood, 2014a). The task of discovering repetitive musical patterns (of which motives, themes, and repeated sections are all examples) consists of retrieving the most relevant musical ideas that repeat at least once within a specific piece (Janssen et al., 2013; Collins, 2013). It not only plays a relevant role in musicological studies —especially with regard to intra-opus analysis—, but it can also yield a better understanding of how composers write and how listeners interpret the underlying structure of music by having a deeper understanding of the music motives of a piece and how they relate with each other. This differs from the previous task of music summarization, where a fixed sized audio file (e.g., around 30 seconds) containing a concatenation of the most repeated parts of a music peace, with as little overlap as possible, was desired.

Computational approaches to the task of pattern discovery can dramatically simplify not only the analysis of a specific piece, but of an entire corpus, potentially offering interesting explorations and relations of patterns across works. Other potential applications include the improved navigation across

both large music collections and stand-alone pieces, or the development of computer-aided composition tools.

Occurrences of these patterns should appear at least twice across a musical work and they may contain slight differences in harmony, timbre, or rhythm. In this section an algorithm that makes use of techniques from the task of music segmentation is described, which exploits repetitive features in order to automatically identify polyphonic musical patterns from audio recordings. This method uses audio recordings as input in an attempt to broaden the applicability of pattern discovery algorithms. Tools that are commonly employed in the music information retrieval task of *music segmentation* combined with a novel score-based greedy algorithm are used in order to identify the most repeated parts of a given audio signal. The input is transformed into a harmonic representation, where the key-invariant SSM (Müller and Clausen, 2007) is computed, and the shortest repeated sequences are found in this new feature space. Finally, the results are evaluated using the JKU Patterns Development Dataset and the metrics described in Chapter II, which are the same used in MIReX (Collins, 2013).

4.1 Rhythmic-Synchronous Harmonic Features

As in the previously presented music summary algorithm, this method also takes tempo-agnostic harmonic representations as input. The same features, PCP and Tonnetz, are used as inputs, as described in Section 2. In this case, the PCP and Tonnetz are normalized such that the maximum energy for a given time frame is 1. Note that, when using these harmonic representations, it is no longer possible to differentiate between octaves, but their compactness

and the energy of each pitch class or tonal mode will become convenient when identifying harmonic repetitions within a piece.

This task will be evaluated against a dataset that only includes deadpan audio versions of the musical pieces (i.e., audio synthesized from a symbolic representation of the piece, such as MIDI). Since this deadpan audio version does not have tempo fluctuations and since the exact BPM for each piece is known (as specified in the MIReX task), it is trivial to extract the exact beats of the tracks. Alternatively, if real music recordings were used as input, the same beat tracker as the one presented in Section 2.1 could be used. Instead of making use of the traditional beat-synchronous approach, which is typically employed in a segmentation task, each beat duration is divided by 4 and aggregated accordingly, thus having $N = 4B$ time frames, where B is the number of beats detected in the piece. The motivation behind this is that some patterns (especially short motives) may not start at the beat level, as opposed to the case for long sections. Furthermore, adding a finer level of granularity (i.e., analyzing the piece at a sixteenth-note level instead of every fourth note or at the beat level) should yield more accurate results in our evaluations. Finally, the same custom implementation is used to compute the audio features as in the previous section, since, as in the music summary task, no other open source methods is available to discover repeated patterns from audio signals.

4.2 Identifying Musical Patterns

The discovery of patterns and their various occurrences involves retrieving actual note collections (which may nest and/or overlap), and so this task can be seen as more complex than structural segmentation, which involves labeling

a single, temporal partition of an audio signal. A repeating musical pattern is defined to be a short idea that is repeated at least once across the entire piece, even though this repetition may be transposed or contain various time shifts. Therefore, each pattern is associated with a set of occurrences that will not necessarily be exact. The patterns and their occurrences may overlap with each other, and this is perfectly acceptable in the context of pattern discovery. An optimal algorithm for this task would (i) find all the patterns contained in a piece and (ii) identify all the occurrences across the piece for each pattern found. In this subsection the algorithm that finds polyphonic patterns as well as a list of all the discovered occurrences for each of the patterns is described.

4.2.1 Finding Repeated Segments

The transposition-invariant SSM $\mathcal{S}$ described in Chapter II is used by this algorithm. It is computed from the selected harmonic features (either PCP or Tonnetz) of a given audio signal using the Euclidean distance, in order to identify repeated segments. As opposed to the task of segmentation, the goal here is to find *all* possible repeated segments in $\mathcal{S}$, independent of how short they are or the amount of overlap present. The other major difference is that the aim is not to find all of the segments of the piece, but rather identify all of the repeated ones. Repeated segments appear in $\mathcal{S}$ as diagonal "stripes," also known as *paths*. Since deadpan audio is used as input (and therefore there will not be any tempo variation), these stripes will be perfectly diagonal.

A score-based greedy algorithm is proposed to efficiently identify the most prominent paths in $\mathcal{S}$. Starting from $\mathcal{S} \in \mathbb{R}^{N \times N}$, half of its diagonals is set to zero, including the main one, due to its symmetrical properties, resulting in $\hat{\mathcal{S}}$, s.t. $\hat{\mathcal{S}}(n,m) = 0$ if $n \leq m$ and $\hat{\mathcal{S}}(n,m) = \mathcal{S}(n,m)$ if $n > m, \forall n, m \in [1:N]$.

A score function σ is computed for each possible path in all the non-zero diagonals of $\hat{\mathcal{S}}$, resulting in a search space of $N(N-1)/2$ possible positions in which paths can start.

Before introducing the score function σ, a trace function is defined given a square matrix $X \in \mathbb{R}^{N_x \times N_x}$ with an offset parameter ω:

$$\mathrm{tr}(X, \omega) = \sum_{i=1}^{N_x - \omega} X(i, i+\omega), \omega \in \mathbb{Z} \tag{22}$$

As can be seen from this equation, when $\omega = 0$ it results in the standard trace function definition.

The score function σ uses various traces of the matrix that comprise a possible path in order to quantify the degree of repetition of the path. If a possible path starts at indices n, m and has a duration of M time frames, then the matrix that the path defines is $P \in \mathbb{R}^{M \times M}$, s.t. $P(i,j) = \hat{\mathcal{S}}(n+i-1, m+j-1), \forall i, j \in [1:M]$. The score σ can be defined as the sum of the closest traces to the diagonal of P (i.e., those with a small ω) and subtract the traces that are farther apart from the diagonal (i.e., where ω is greater). This is normalized in order to obtain a score independent from the duration M of the possible path:

$$\sigma(\rho) = \frac{\left(\sum_{\omega=-(\rho-1)}^{\rho-1} \mathrm{tr}(P, \omega)\right) - \mathrm{tr}(P, \pm\rho)}{M + \sum_{i=1}^{\rho-1} 2(M-i)} \tag{23}$$

where $\rho \in \mathbb{N}$ is the maximum offset to be taken into account when computing the traces of P. The greater the ρ, the greater the σ for segments that contain substantial energy around their main diagonal (e.g., paths that con-

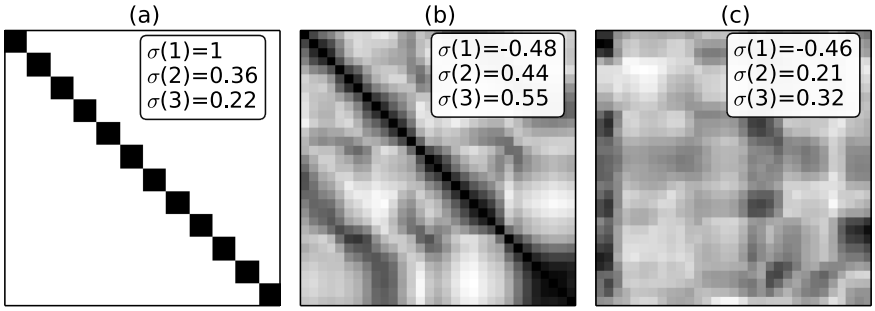


Figure 10: Three examples showing the behavior of the path score $\sigma(\rho)$. (a) shows a synthetic example of a perfect path. (b) contains a real example of a path in which there is some noise around the diagonal of the matrix. In (c), a matrix with no paths is shown.

tain significant rhythmic variations), even though the precision decreases as ρ increases.

Examples for various $\sigma(\rho)$ can be seen in Figure 10. For a perfectly clean path (a), $\rho = 1$ gives the maximum score of 1. However, the score decreases as ρ increases, since there is zero energy in the diagonals right next to the main diagonal. On the other hand, for matrices extracted from audio signals (b and c), the scores $\sigma(1)$ are low, indicating that the diagonals next to the main diagonal contain amounts of energy similar to the main diagonal. However, when $\rho > 1$, the score is substantially different from a matrix with a path (b) and a matrix without one (c).

For all $N(N-1)/2$ positions in which paths can potentially start in $\hat{\mathcal{S}}$, the goal is to extract the most prominent ones (i.e., the ones that have a high σ). At the same time, the paths should be extracted from beginning to end in the most accurate way possible. The proposed algorithm assigns a certain σ to an initial possible path $\hat{z}$ of a minimum length of ν time frames, which reduces the search space to $(N-\nu+1)(N-\nu)/2$. If the score σ is greater than a certain threshold θ, the possible path is increased by one time frame, and

recomputed σ until $\sigma \leq \theta$. By then, the path $\hat{z}$ can be stored as a segment in the set of segments $\mathcal{Z}$. In order to avoid incorrectly identifying possible paths that are too close to the found path, this path from $\hat{\mathcal{S}}$ is set to zero, including all the ρ closest diagonals, and the search continues starting from the end of the recently found path.

The pseudocode for this process can be seen in Algorithm 2, where $|\mathbf{x}|$ returns the length of the path $\mathbf{x}$, $\{x\}$ returns the path in which all elements equal x, the function ComputeScore computes the $\sigma(\rho)$ as described in equation 23, OutOfBounds$(\mathbf{x}, X)$ checks whether the path $\mathbf{x}$ is out of bounds with respect to X, IncreasePath$(\mathbf{x})$ increases the path $\mathbf{x}$ by one (analogously as DecreasePath), and ZeroOutPath$(X, \mathbf{x}, \rho)$ assigns zeros to the path $\mathbf{x}$ found in X, including all the closest ρ diagonals.

Algorithm 2 Find Repeated Segments

Require: $\hat{\mathcal{S}}, \rho, \theta, \nu$
Ensure: $\mathcal{Z} = \{z_1, \ldots, z_k\}$
  for $\hat{z} \in \hat{\mathcal{S}} \wedge |\hat{z}| = \nu \wedge \hat{z} \neq \{0\}$ **do**
    $b \leftarrow$ False
    $\sigma \leftarrow$ ComputeScore$(\hat{z}, \rho)$
    while $\sigma > \theta \wedge \neg$OutOfBounds$(\hat{z}, \hat{\mathcal{S}})$ **do**
      $b \leftarrow$ True
      $\hat{z} \leftarrow$ IncreasePath$(\hat{z})$
      $\sigma \leftarrow$ ComputeScore$(\hat{z}, \rho)$
    end while
    if b **then**
      $\mathcal{Z}$.add(DecreasePath$(\hat{z})$)
      ZeroOutPath$(\hat{\mathcal{S}}, \hat{z}, \rho)$
    end if
  end for
  return $\mathcal{Z}$

An example of the paths found by the algorithm when using PCP features is shown in Figure 11. Parts of some segments are repeated as standalone

segments (i.e., segments within segments), therefore allowing overlap across patterns as expected in this task. Observe how some of the segments repeat almost exactly across the piece —there is a set of patterns at the top of the matrix that appears to repeat at least three times. The next step of the algorithm is to cluster these segments together so that they represent a single pattern with various occurrences.

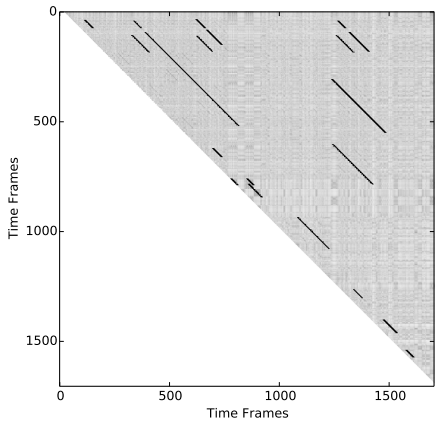


Figure 11: Paths found (marked in white) using the proposed algorithm for Chopin's Op. 24 No. 4., with $\theta = 0.33, \rho = 2$ and PCP features.

4.2.2 Clustering the Segments

Each segment $z \in \mathcal{Z}$, defined by the two indices in which it starts (s_i, s_j) and ends (e_i, e_j) in $\hat{\mathcal{S}}$, contains two occurrences of a pattern: the one that starts in s_i and ends in e_i and the one that occurs between the time indices s_j and e_j. In order to cluster the repeated occurrences of a single pattern, an occurrence for each segment $z \in \mathcal{Z}$ is found if one of the other segments in $\mathcal{Z}$ starts and ends in similar locations with respect to the second dimension of $\hat{\mathcal{S}}$. Note that the bottom left triangle of the matrix is set to zero as explained in subsection 4.2.1, so first dimension to cluster the occurrences can not be used. Formally, a segment $\hat{z}$ is an occurrence of z if

$$(s_j^z - \Theta \leq s_j^{\hat{z}} \leq s_j^z + \Theta) \wedge (e_j^z - \Theta \leq e_j^{\hat{z}} \leq e_j^z + \Theta) \tag{24}$$

where s_j^z represents the starting point of the segment z in the second dimension of $\hat{\mathcal{S}}$ and analogously e_j^z is the ending point, and Θ is a tolerance parameter.

4.2.3 Final Output

At this point, a set of patterns with their respective occurrences represented by their starting and ending time frame indices is available. Even though the algorithm is not able to distinguish the different musical lines within the patterns, the annotated score can be used to output the exact notes that occur during the identified time indices, as suggested in the MIReX task (Collins, 2013). If no score is provided, only the time points will be presented. In order to overcome this limitation in future work, the audio should be source-separated to identify the different lines and perform an F0 estimation to correctly identify the exact melody that defines the pattern (and not just the time points at which it occurs). Progress towards this goal has been presented in (Collins et al., 2014a).

4.2.4 Time Complexity Analysis

Once the rhythm-synchronous features are computed, the process of calculating the transposition-invariant SSM is $\mathcal{O}(13N^2)$ for the PCP and $\mathcal{O}(7N^2)$ for the Tonnetz, which asymptotically converges to $\mathcal{O}(N^2)$, where N is the number of time frames of the harmonic features used. The procedure to compute the score given a path has a time complexity of $\mathcal{O}(2\rho M) = \mathcal{O}(\rho M)$, where ρ is the required parameter for the computation of the score, and M is the length

of the path from which to compute the score. The total process of identifying segments is $\mathcal{O}\left(\frac{(N-\nu+1)(N-\nu)}{2}\rho M\right) = \mathcal{O}((N-\nu)^2\rho M)$, where ν is the minimum number of time frames that a pattern can have. Asymptotically, the clustering of the segments can be neglected, since the length of $\mathfrak{Z}$ will be much less than N. Therefore, the total time complexity of the proposed algorithm is $\mathcal{O}(N^2 + (N-\nu)^2\rho M)$.

4.3 Evaluation

The JKU Patterns Development Dataset* is used to evaluate the algorithm. This dataset is comprised of five classical pieces annotated by various musicologists and researchers (Collins, 2013). Moreover, this dataset is the public subset of the one employed to evaluate the Pattern Discovery task at MIReX.

4.3.1 Results

This algorithm is evaluated using the standard metrics described in Chapter II. The results of the proposed algorithm, computed using the open source evaluation package `mir_eval` (Raffel et al., 2014), are shown in Table 4 both when using PCPs and Tonnetz, averaged for the entire JKU Dataset, along with an earlier version of the algorithm submitted to MIReX (Nieto and Farbood, 2013b), another recent algorithm called SIARCT-CFP (Collins et al., 2014a) that is assessed using both audio and symbolic representations as input in (Collins et al., 2014b), and "COSIATEC Segment," a method that only uses symbolic inputs (Meredith, 2013). This latter method is used for comparison because it is the only symbolic method in which the author has access to all

*`https://dl.dropbox.com/u/11997856/JKU/JKUPDD-Aug2013.zip`

Alg	P_{est}	R_{est}	F_{est}	$P_{\text{O(.75)}}$	$R_{\text{O(.75)}}$	$F_{\text{O(.75)}}$	P_3	R_3	F_3	$P_{\text{O(.5)}}$	$R_{\text{O(.5)}}$	$F_{\text{O(.5)}}$	Time (s)
PCP	54.96	51.73	**49.80**	37.58	27.61	**31.79**	35.12	35.28	**32.01**	45.17	34.98	38.73	454
Tonnetz	53.37	51.97	48.20	37.72	24.84.61	29.63	31.92	35.92	29.43	41.27	31.47	34.72	420
(Collins et al., 2014b)	14.9	60.9	23.94	–	–	–	–	–	–	62.9	51.9	**56.87**	–
(Nieto and Farbood, 2013b)	40.83	46.43	41.43	32.08	21.24	24.87	30.43	31.92	28.23	26.60	20.94	23.18	**196**
(Collins et al., 2014b)	21.5	78.0	33.7	–	–	–	–	–	–	78.3	74.7	76.5	–
(Meredith, 2013)	43.60	63.80	50.20	65.40	76.40	68.40	40.40	54.40	44.20	57.00	71.60	63.20	7297

Table 4: Results of various algorithms using the JKU Patterns Development Dataset, averaged across pieces. The top rows of the table contain algorithms that use deadpan audio as input. The bottom rows correspond to algorithms that use symbolic representations as input.

of the resulting metrics, and SIARCT-CFP since it is the most recent method that uses audio as input. The parameter values used to compute these results, $\nu = 8, \theta = 0.33, \rho = 2$, and $\Theta = 4$, were found empirically. Moreover, it can be seen how the proposed algorithm is better than (Nieto and Farbood, 2013b) in all the metrics except running time; it also finds more correct patterns than (Collins et al., 2014b) (the current state-of-the-art when using audio as input).

It is clear from the table that PCP is slightly better than Tonnetz in all of the metrics, except for running time. It could be argued that, since beat tracking the deadpan audio files of the JKU dataset is trivial —as opposed to tracking the Mazurkas—, Tonnetz do not have a strong positive result in the final output. Furthermore, in this task the SSM is computed on the actual audio features, not on a quantized version of them. This quantization might have been disadvantageous to PCP, that reside in a higher dimensional space than Tonnetz (12 vs 6 dimensions), thus k should have been higher to compete with the Tonnetz in the music summary method evaluation. Working on the features themselves instead of a quantized representation seems to make PCP a better candidate than Tonnetz, at least when used in this pattern

discovery algorithm. On the other hand, due to the reduced dimensionality of the Tonnetz, a decrease of 34 seconds of running time is obtained when using Tonnetz compared to PCP.

This algorithm obtains state-of-the-art results when extracting patterns from audio, obtaining an F_{est} of 49.80%. This is better than the symbolic version of (Collins et al., 2014a) and almost as good as the algorithm described in (Meredith, 2013). It could be claimed that this result is actually better, since P_{est} and R_{est} are closer to each other, which is always more desirable, as it will be seen in Chapter VI. The fact that these results are superior or comparable to the two other algorithms using symbolic representations indicates the potential of the presented method.

When evaluating the occurrences of the patterns, it becomes clear that the proposed algorithm is still better than (Nieto and Farbood, 2013b), but worse than (Collins et al., 2014a) (at least for $c = .5$, which is the only reported result). Nevertheless, the numbers are much lower than (Meredith, 2013). In this case, working with symbolic representations (or estimating the F0 in order to apply a symbolic algorithm as in (Collins et al., 2014a)) yields significantly better results. It is interesting to note that when the tolerance increases (i.e., $c = .5$), the results of the presented method improve as opposed to the other algorithms, therefore it remains to be seen if this algorithm is actually comparable with the audio version of (Collins et al., 2014a) when using $c = .75$. This might be due to the fact that some of the occurrences found in the SSM were actually very similar (therefore they were found in the matrix) but were slightly different in the annotated dataset. A good example of this would be an occurrence that contains only one melodic voice. The proposed algorithm only finds points in time in which an occurrence might be included, it does

not perform any type of source separation in order to identify the different voices. If the tolerance decreases sufficiently, a polyphonic occurrence would be accepted as similar to a monophonic one corresponding to the same points in time.

Our three layer score F_3 is the best result when using audio recordings and PCP, with an F-measure of 31.74% (unfortunately this metric was not reported in (Collins et al., 2014a)). This metric aims to evaluate the quality of the algorithm with a single score, including both pattern establishment and occurrence retrieval. The results of the method described here are still far from perfect (32.01%), but when compared to an algorithm that uses symbolic representations (Meredith, 2013) (44.21%), it appears that this difference, though significant, is not as large as one might expect given it is the state-of-the-art for symbolic representations.

Finally, this algorithm takes more than twice as long as (Nieto and Farbood, 2013b). However, it is over 16 times faster than (Meredith, 2013), indicating its efficiency in terms of computation time, specially when compared to these symbolic-based methods. This algorithm is implemented in Python and available for public download*.

5 Summary

In this chapter two novel MIR methods that aim at discovering different aspects of music structure were presented. The first one is an audio summary criterion that produces audible summaries from music recordings, which were assessed through data-driven evaluation and qualitative inspection. This crite-

*`https://github.com/urinieto/MotivesExtractor`

rion consistently produces informative summaries that capture both meaningful harmonic and high-level structural information. Additionally, a heuristic approach capable of producing audio summaries that closely approximates the absolute maximum was also introduced. Furthermore, several example summaries are made available online*.

This criterion, when tunning its hyper-parameters accordingly, could potentially extract music motives from audio, as argued in this chapter. Following this idea, the second presented algorithm is a novel method to discover repeating polyphonic patterns using audio recordings as input. The method makes use of various standard techniques typically used for music segmentation, and it is much more efficient than the summary criterion. This method was evaluated using the JKU Pattern Development Dataset and it was shown how it obtains competent results when retrieving all the occurrences of the patterns and state-of-the-art results when finding patterns. When the algorithm is compared to others that use symbolic representations, it is comparable or superior in terms of the correct patterns found. Since this method is much simpler to those time consuming approaches that employ convolutive processes (e.g., (Lartillot, 2014) can take weeks to estimate the different patterns), while still obtaining competitive results, the author hopes that this work should motivate and inspire future simpler and yet more effective algorithms. Also in future work, source separation might be needed to successfully identify patterns that only comprise a subset of the different musical lines.

In the next chapter two other novel algorithms that can be categorized under the *music segmentation* MIR task will be described. As seen in Chap-

* `https://files.nyu.edu/onc202/public/ismir2012`

ter II, this task is considerably more popular in the MIR literature than music summarization or pattern discovery, therefore the following methods will be assessed against many other existing ones and it will be shown how the proposed methods are competent in comparison.

CHAPTER IV

MIR METHODS: MUSIC SEGMENTATION

1 Introduction

The description of MIR methods continues here with the introduction of two algorithms to discover structure in music categorized under the MIR task of *music segmentation*. The first one, convex non-negative matrix factorization (C-NMF) (Nieto and Jehan, 2013), aims to both identify musical boundaries and label segments based on their acoustic similarity; the second, 2D Fourier Magnitude Coefficients (2D-FMC) (Nieto and Bello, 2014), deals with the labeling problem exclusively.

In the MIR community the task of music segmentation has been more widely discussed than music summarization or pattern discovery*, and therefore multiple segmentation methods are available on-line as open source projects. These will be used to compare the quality of the methods presented here, and in order to avoid differences that might arise due to the feature computation, the same audio features will be employed for all the techniques evaluated in this chapter. To compute these audio features the open source package Essentia is used (Bogdanov et al., 2013), which provides transparent implementations for the most standard features in MIR. The output of these methods will then be

* The music segmentation task was first run in MIReX in 2009, while the pattern discovery task started on 2013, and no standard task for music summarization exists yet.

evaluated against various human annotated datasets that are common in the music segmentation task in order to assess the accuracy of their results.

2 Convex Non-negative Matrix Factorization

This method aims at identifying large scale non-overlapping music segments by using a specific type of matrix factorization that adds a convex constraint (Ding et al., 2010) to obtain a decomposition that captures the different section prototypes of a musical piece (e.g., verse, chorus) in a more consistent and efficient way than classic non-negative matrix factorization, which has been previously used to label the different segments of a musical piece (Kaiser and Sikora, 2010). The technique presented here is capable of both identifying the boundaries of the sections and grouping them based on their acoustic similarity. Most segments found by this method fall under the *homogeneity* principle of music structure, which identifies the passages that contain relatively uniform musical aspects. Additionally, this method is evaluated on two different datasets and it is shown that it is competitive compared to other music segmentation techniques, outperforming other methods that also aim at identifying homogeneous segments.

2.1 Pre-Processing and Enhancing Audio Features

As it is common in this task, the segmentation will be based on the underlying harmony of the musical piece, therefore harmonic features will be used as input to this method. More specifically, PCPs and Tonnetz discussed in Chapter II, are the harmonic representations that this method will exploit. In order to obtain beat-synchronous representations, the beats are estimated

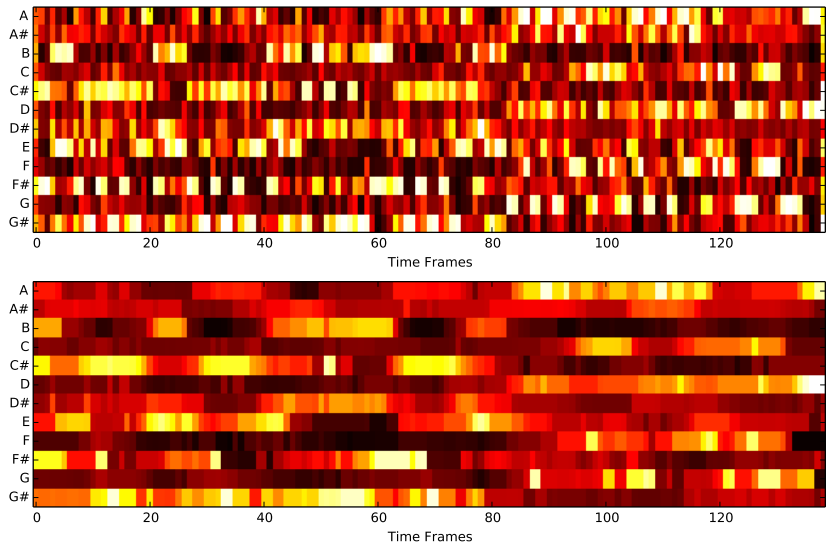


Figure 12: Example of PCPs (top) and filtereds PCP with $h = 9$ (bottom), of the song *And I Love Her* by The Beatles.

using Essentia's beat tracker (Zapata et al., 2013) and the technique described in (Ellis and Poliner, 2007) is employed to average the features across frames.

A series of transformations are applied to these beat-synchronous harmonic features in order to better capture the different parts of a song (as it is common in this type of problems (Paulus et al., 2010)). First, a sliding median filter of size h is run against each of the beat-synchronous vectors. The median filter gives sharper edges than a regular mean filter (Cho and Bello, 2014), which yields higher precision when identifying section boundaries. By filtering features across time, the most prominent feature vectors are retained within the h-size window and smaller artifacts are removed, since they can be considered irrelevant in this context. This results in a representation that is more suitable when retrieving homogeneous segments. In Figure 12 examples of non-filtered PCPs and their corresponding pre-preprocessed PCPs are shown.

2.2 Convex NMF in Music Segmentation

2.2.1 Convex NMF Description

The factorization of an input feature matrix $X \in \mathbb{R}^{p \times N}$, composed of $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, which has N column observations $\mathbf{x}_i$ of p features, can be described as $X \approx FG$, where $F \in \mathbb{R}^{p \times r}$ can be interpreted as a *centroid* column matrix, $G \in \mathbb{R}^{r \times N}$ is composed of rows with the activations of these centroids across the observations, and r is the rank of decomposition (i.e., the number of clusters). In NMF, both F and G are enforced to be positive (and thus X must also be positive). A column vector is denoted by $\mathbf{z}$ and a row one by $\mathbf{z}^T$.

C-NMF adds a constraint to $F = (\mathbf{f}_1, \ldots, \mathbf{f}_r)$ such that its columns $\mathbf{f}_j$ become *convex combinations* of the observations of X:

$$\mathbf{f}_j = \mathbf{x}_1 w_{1j} + \cdots + \mathbf{x}_N w_{Nj} = X\mathbf{w}_j \quad j \in [1 : r] \tag{25}$$

For a linear combination to be convex, all coefficients w_{ij} must be positive and the sum of each set of coefficients $\mathbf{w}_j$ must be 1. Formally: $w_{ij} \geq 0, ||w_i||_1 = 1$.

This results in $F = XW$, where $W \in \mathbb{R}^{N \times r}$, which makes the columns $\mathbf{f}_j$ interpretable as weighted cluster centroids, representing, in this case, better section prototypes of the musical piece and sharper activations as it will become apparent in subsection 2.2.3. Finally, C-NMF can be formally characterized as: $X \approx XWG$.

For a more detailed description of C-NMF with an algorithm explanation and sparsity discussion the reader is referred to (Ding et al., 2010). A good review of algorithms for NMF can be found in (Lee and Seung, 2000). Lastly,

a good example of C-NMF in computer vision can be found in (Thurau et al., 2009).

2.2.2 C-NMF vs NMF

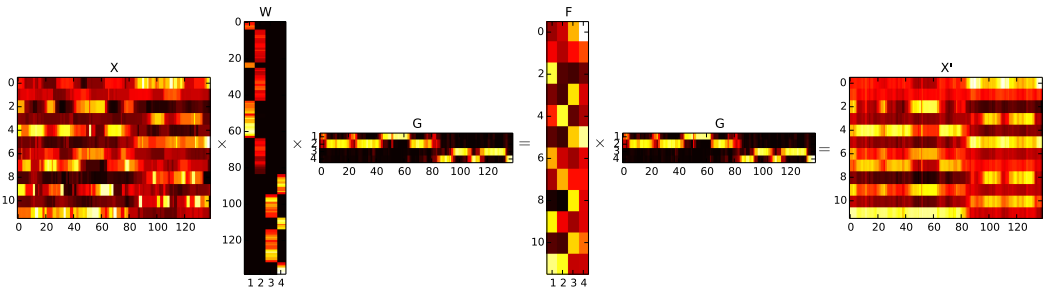
As opposed to NMF, in C-NMF the matrix F is a set of convex combinations of the columns of the input matrix X (see Equation 25). The matrices W and G are naturally sparse when adding this convex constraint, as opposed to traditional NMF (where G is not necessarily sparse, and W does not exist).

This convex constraint can be seen as an extra layer if framing this problem using deep learning. In this case, the constraint would act as a *regularizer*, where only certain combinations (i.e., convex ones) are allowed, therefore creating activations that are much more sparse, with centroids that are more informative than in standard NMF. Sparsification schemes have been explored on top of NMF (Hoyer, 2004; Li et al., 2001), however they yield centroids that are not as meaningful as the ones produced by C-NMF (Ding et al., 2010).

To illustrate this principal difference the song *And I Love Her* by The Beatles is analyzed using its PCPs as the input X to both matrix factorization methods, as can be seen in Figure 13. From the figure it becomes clear how much sparser G is in C-NMF as opposed to standard NMF. Moreover, the matrix W is even sparser than G in C-NMF, since the sum of its columns must sum 1 due to its convex linear combination property. This produces a matrix F that, in the case of C-NMF, better captures the pitch distribution of the segment prototypes, which is desired in the case of music segmentation.

The benefits of the sparsity of G are two-fold: (i) this matrix yields preferred results when segmenting music as it will be seen in the evaluation section, and (ii) it generates more consistent factorizations, since C-NMF is less

(a) Convex Non-negative Matrix Factorization



(b) Non-negative Matrix Factorization

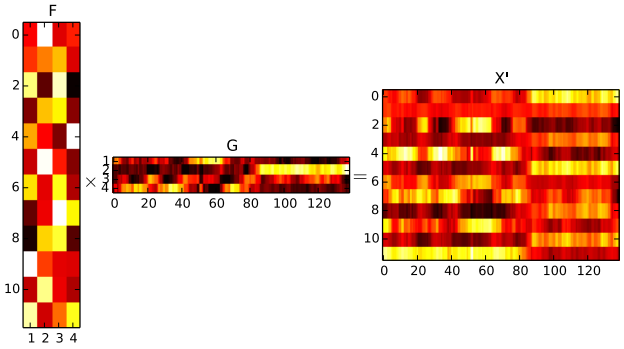


Figure 13: Comparison of C-NMF and NMF when decomposing the PCPs representing the song *And I Love Her* by The Beatles.

sensitive to its initialization due to the added convex constraint. To explore and compare the levels of consistency, both C-NMF and NMF are computed $T = 100$ times for the song *Help!* by The Beatles with $r = 2$, and their decomposition matrices are explored. A decomposition matrix R_k is the product between one cluster $\mathbf{f}_k$ in F and its respective row activation $\mathbf{g}_k^T$ in G, such that $R_k = \mathbf{f}_k \mathbf{g}_k^T, \forall k \in [1 : r]$. Therefore, there are r different decomposition matrices for each matrix factorization process. The pairwise Euclidean distance $C(\mathcal{R}^i, \mathcal{R}^j), \forall i, j \in [1 : T]$ is computed between their resulting sets of decomposition matrices $\mathcal{R}^n = \{R_1^n, \ldots, R_r^n\}$ (where n is the execution index, $n \in [1 : T]$). Formally:

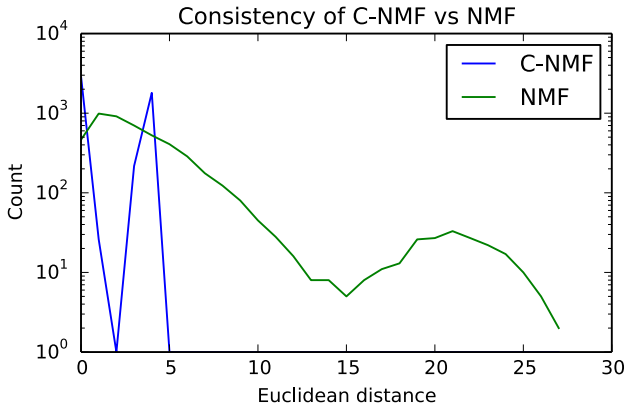


Figure 14: Logarithmic histogram of distances between 100 sets of decomposition matrices obtained with C-NMF (blue) and NMF (green) from the song Help! by The Beatles.

$$C(\mathcal{R}^i, \mathcal{R}^j) = \sum_{m=1}^{r} ||{R_m}^i - {R_m}^j||_2 \quad i, j \in [1:T] \tag{26}$$

In Figure 14 the logarithmic histogram of these differences is plotted for each method, so that the shorter the difference, the more consistent the technique will be. Consistency is desired, since it will be possible to obtain stable solutions with less iterations, thus having a better efficiency in terms of computation time. As it can be seen, C-NMF's greatest difference is smaller than 5, and NMF's greatest difference is almost 45, thus illustrating the preference of C-NMF over NMF in terms of stability and consistency.

2.2.3 Applying C-NMF in Music Segmentation

In this subsection it is described how C-NMF can be useful in the task of music structure analysis. This part is divided into the two main problems of music segmentation: boundaries and labels.

Identifying Boundaries Harmonic features (either PCPs or Tonnetz) are used as input to the C-NMF process, as illustrated in Figure 13a. It is expected that C-NMF learns the inherent harmonic structure by yielding meaningful cluster centroids in F, which, in this case, would represent the most likely harmonic distribution for each of the segments. The number of clusters when identifying boundaries, r, is a fixed parameter of our system, which should be tuned based on the type of music to be analyzed (as it is common in this type of techniques (Kaiser and Sikora, 2010)). As discussed in section 2.2.2, W and G are both sparse matrices and, while W contains the weights for the convex combination of the time frames of the harmonic features, G can be seen as the *activation* of the harmonic distributions learned in F across the track. This fact is exploited by using G to identify the boundaries of the segments.

Nevertheless, G can become noisy when the track can not be split into r coherent segments due to track complexity or because the used features are not able to capture the similarity between segments (e.g., there are no harmonically different parts). In order to address this limitation a set of transformations on G are applied such that the boundaries between segments can be easily extracted.

First, a simplified, discrete version of the G matrix, $\mathcal{G}$, is obtained where, for each time frame, an identifier of the segment prototype is assigned to the most prominent cluster and 0 for the rest of them. This is analogous to the classical signal processing technique of *vector quantization*. Formally:

$$\mathcal{G}_{i,j} = \begin{cases} i, & \text{if } i = \arg\max_k G_{k,j} \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in [1:r], \forall j \in [1:N]$$

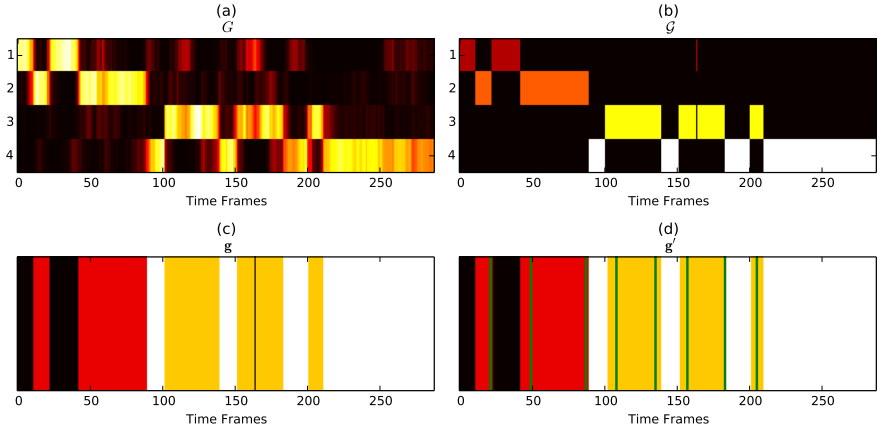


Figure 15: Example of the extraction of boundaries of the song *Strawberry Fields Forever* by The Beatles using $r = 4$. (a) Factorized matrix G obtained using C-NMF. (b) Discrete matrix $\mathcal{G}$. (c) Aggregated array $\mathbf{g}$. (d) filtered array $\mathbf{g}'$ with the ground truth boundaries depicted as green vertical lines.

Thus, at a given time frame, one and only one cluster is activated, removing any possible overlaps between segments. Then $\mathcal{G}$ is collapsed into an array $\mathbf{g}$, and median filter of size θ is applied to remove noise and segments that are too short. This process is analogous to the filtering operation on the features prior to the C-NMF decomposition. From the filtered array $\mathbf{g}'$ the boundaries can be easily extracted by obtaining one boundary for each change of segment identifier. This process is illustrated in Figure 15.

Labeling Segments Once the filtered array $\mathbf{g}'$ has been computed the labels can be obtained by using the identifiers of each of the segments represented in the array. However, the usage of a higher number of unique segments might be needed in order to capture as many unique labels as possible. Due to the properties of matrix factorization, the ranking r to identify the boundaries must remain relatively small to avoid too many noisy segments, whereas the number of unique labels should be as close to the average of the music to be

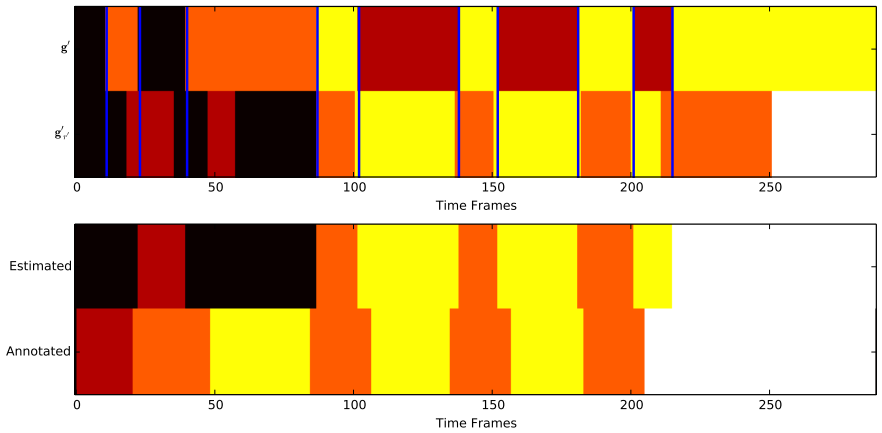


Figure 16: Example of the labeling of the segments of the song *Strawberry Fields Forever* by The Beatles using $r' = 5$. On the top plot the two arrays $\mathbf{g}'$ and $\mathbf{g}'_{r'}$ are plotted with the identified boundaries marked with blue vertical lines. On the bottom plot, the estimated labels are plotted on top of the annotated ones.

analyzed as possible. To do so, C-NMF is run again with a different rank of decomposition r' (where, typically for pop music $r' > r$), and use the previously identified boundaries to smooth the resulting $\mathbf{g}'_{r'}$ for each boundary found. This smoothing process consists of simply taking the most frequent segment identifier in $\mathbf{g}'_{r'}$ within each pair of found boundaries. This produces one specific label identifier for each segment, which is based on the harmonic similarity across segments.

There are two main drawbacks on this labeling process. First, the number of unique labels r' is a fixed parameter, and as aforementioned, it is highly sensitive to the musical style of the dataset to be analyzed. However, it is not uncommon for music segmentation algorithms to use this approach to fix parameters (Kaiser and Sikora, 2010). Second, this method is not capable of capturing similar segments that are key-transposed. An algorithm to overcome this (2D-FMC) is presented in the next section of this chapter. The labeling process is illustrated in Figure 16.

2.3 Evaluation

This algorithm is evaluated with the annotated Beatles dataset published by Isophonics (ISO-Beatles)*. This dataset is composed of 179 songs and is traditionally used to evaluate segmentation techniques (Serrà et al., 2014; Levy and Sandler, 2008; Kaiser and Sikora, 2010; Weiss and Bello, 2011). This method is further evaluated with the SALAMI dataset (Smith et al., 2011), which contains 769 pieces and offers a wider variety of music genres.

The following parameters, found empirically to maximize the results, were used in our evaluation: $r = 3$ for the rank of decomposition for the boundaries, and $r' = 5$ for the number of unique segments per track. For the Beatles, $h = 13$ beats were used for the size of the median-filter window for the features, while $h = 8$ was set for SALAMI. Finally, $\theta = 18$ beats were employed for the size of the median-filter for the activation matrix G in The Beatles dataset, whereas $\theta = 15$ was used in SALAMI.

The results of the algorithm (both using PCPs and Tonnetz) are compared against two other techniques that also tend to identify homogeneous segments: SI-PLCA (Shift Invariant Probabilistic Latent Component Analysis (Weiss and Bello, 2011)) and CC (Constrained Clustering (Levy and Sandler, 2008)). The parameters used for SI-PLCA are the ones proposed for MIREX (see source code†). The parameters used for CC are the ones that come with its open source implementation‡. Additionally, our algorithm is compared to a modified version that uses NMF instead of C-NMF, in order

*`http://isophonics.net/content/reference-annotations-beatles`

†`http://ronw.github.io/siplca-segmentation`

‡`http://code.soundsoftware.ac.uk/projects/qm-dsp`

to see the differences of the convex constraint when used in music segmentation. The same features described in Section 2.1 were used for the three algorithms and the NMF variation. Furthermore, these results are compared with the ones reported for a recent technique that aims to identify the three types of music segments (instead of only the homogeneous) and obtains the highest results for various metrics, called SF (Structural Features (Serrà et al., 2014))*. All the results reported both for the boundaries and for the labels were computed using `mir_eval` (Raffel et al., 2014), the open source package discussed in Chapter II that contains implementations for the evaluation of the most common MIR tasks.

2.3.1 Boundaries Evaluation

The metrics used to evaluate the boundaries are the Hit Rate at 3 seconds (F-measure $\mathbf{F}_3$, precision $\mathbf{P}_3$, recall $\mathbf{R}_3$) and at 0.5 seconds ($\mathbf{F}_{0.5}$, $\mathbf{P}_{0.5}$, $\mathbf{R}_{0.5}$). These two metrics are the standard methods to evaluate boundaries in MIReX (Smith and Chew, 2013). Moreover, the same metrics are also reported with the first and last boundaries trimmed (this is denoted by the t symbol), since these two boundaries can be trivially extracted and they could yield misleading results, as it is discussed in (Nieto and Smith, 2013). On Table 5 the boundary results are reported.

Starting with an analysis of the harmonic features, PCPs (C-NMF$_P$) seem to consistently outperform Tonnetz (C-NMF$_T$) in all conditions. Two reasons might be the cause of this: (i) matrix factorization tends to produce more meaningful centroids when more information is included in the input

*Some of the SF results are reported in (McFee and Ellis, 2014b).

ISO-Beatles												
	$\mathbf{F}_3$	$\mathbf{P}_3$	$\mathbf{R}_3$	$\mathbf{F}_{0.5}$	$\mathbf{P}_{0.5}$	$\mathbf{R}_{0.5}$	$\mathbf{F}_{3t}$	$\mathbf{P}_{3t}$	$\mathbf{R}_{3t}$	$\mathbf{F}_{0.5t}$	$\mathbf{P}_{0.5t}$	$\mathbf{R}_{0.5t}$
C-NMF$_P$	**60.41**	59.84	63.45	24.89	24.52	26.41	**51.66**	51.57	54.93	**8.46**	8.39	9.13
C-NMF$_T$	56.72	56.17	60.55	23.71	23.23	25.78	47.42	47.45	51.40	7.34	7.07	8.50
NMF	54.56	55.57	56.78	22.63	22.94	23.71	45.75	47.24	48.17	7.62	7.67	8.21
SI-PLCA	50.12	70.59	39.97	**28.27**	39.57	22.74	34.62	56.88	26.05	5.97	9.51	4.71
CC	55.06	60.17	52.16	25.06	27.30	23.86	43.75	49.26	41.06	6.30	7.06	4.11
SF	77.4	75.3	81.6	–	–	–	65.8	62.1	72.8	15.3	14.4	16.9
SALAMI												
	$\mathbf{F}_3$	$\mathbf{P}_3$	$\mathbf{R}_3$	$\mathbf{F}_{0.5}$	$\mathbf{P}_{0.5}$	$\mathbf{R}_{0.5}$	$\mathbf{F}_{3t}$	$\mathbf{P}_{3t}$	$\mathbf{R}_{3t}$	$\mathbf{F}_{0.5t}$	$\mathbf{P}_{0.5t}$	$\mathbf{R}_{0.5t}$
C-NMF$_P$	**49.56**	46.97	59.25	21.59	20.38	26.27	**40.65**	38.35	51.31	**7.77**	7.45	9.94
C-NMF$_T$	45.91	45.00	53.16	20.83	20.30	24.46	35.86	35.45	43.36	6.00	5.90	7.27
NMF	46.42	45.73	53.93	21.09	20.68	24.73	36.72	36.46	44.39	6.39	6.47	7.58
SI-PLCA	43.73	60.57	36.73	**28.65**	44.81	23.62	24.97	34.50	21.02	3.94	6.52	3.04
CC	49.41	52.54	50.08	22.19	23.30	22.86	38.51	41.94	39.69	5.39	5.97	5.38

Table 5: Boundary results for the four different algorithms (C-NMF, NMF, SI-PLCA, and CC) applied to two different datasets: ISO-Beatles (top) and SALAMI (bottom). Additionally, the reported results for the SF algorithm are also shown for the ISO-Beatles.

(each Tonnetz frame is composed of a 6 dimensional vector, as opposed to the 12 dimensions included in each PCPs frame); and (ii) PCPs might be more *separable* than Tonnetz given the type of music contained in the datasets and, again, their differences in dimensionality.

This contrasts with the behavior of Tonnetz described in Chapter III, where Tonnetz outperform PCPs when using beat-synchronous features. One explanation might be that beat trackers obtain more accurate results when analyzing the pieces contained in ISO-Beatles and SALAMI than those in the Mazurkas dataset, which is more challenging to analyze rhythmically (Grosche, 2010). Consistently, Tonnetz decrease performance on the other algorithms,

therefore the values reported on the table for the other methods were computed using PCPs features.

As shown in the table, C-NMF outperforms SI-PLCA and CC in the F-measure of the Hit Rate measure with a 3 second tolerance on both datasets, but SI-PLCA obtains a better score with a tolerance of 0.5 seconds. When the precision ($\mathbf{P}_3$ and $\mathbf{P}_{0.5}$) and recall ($\mathbf{R}_3$ and $\mathbf{R}_{0.5}$) values are observed, it can be seen that SI-PLCA has a quite high precision and low recall. This means that SI-PLCA retrieves too few boundaries compared to the ones contained in the ground truth (i.e., it under-segments), but these tend to be correct. However, when the F-measure of the trimmed scores is observed ($\mathbf{F}_{3t}$ and $\mathbf{F}_{0.5t}$), it becomes clear that the method presented here outperforms the other two (both when using Tonnetz or PCPs), conveying that the high precision of SI-PLCA benefited from the correct retrieval of the first and last boundaries. Moreover, and without the convex constraint, this method would obtain lower results (see NMF on the table), therefore demonstrating the value of this constraint in matrix factorization for music segmentation. Since our method not only obtains higher results in the trimmed version both for ISO-Beatles and SALAMI, but also maintains a desired parity between precision and recall, it is possible to conclude that C-NMF is able to retrieve homogeneous boundaries more successfully than the other three methods*. Nevertheless, when comparing C-NMF with an algorithm that aims to retrieve the three types of boundaries (homogeneous, repetitive, and novel) such as SF, it becomes apparent how SF outperforms C-NMF in all the metrics†.

* At least using these standard metrics. In Chapter VI it is discussed how higher precision is more perceptually desired than higher recall.

† The SF results are incomplete since the author does not have access to its original source

ISO-Beatles						
	$\mathbf{Pw}_f$	$\mathbf{Pw}_p$	$\mathbf{Pw}_r$	$\mathbf{S}_f$	$\mathbf{S}_o$	$\mathbf{S}_u$
C-NMF$_P$	**53.53**	58.29	52.65	**57.20**	55.82	60.63
C-NMF$_T$	49.13	54.00	49.02	53.86	53.16	57.65
NMF	48.74	49.15	49.15	54.64	57.43	54.68
CC	49.18	62.91	41.06	56.50	50.36	66.50
SI-PLCA	49.36	42.67	65.17	48.08	62.28	42.67
SF	71.1	78.7	68.1	–	–	–

SALAMI						
	$\mathbf{Pw}_f$	$\mathbf{Pw}_p$	$\mathbf{Pw}_r$	$\mathbf{S}_f$	$\mathbf{S}_o$	$\mathbf{S}_u$
C-NMF$_P$	**50.96**	59.89	51.39	**54.01**	53.07	62.82
C-NMF$_T$	48.67	57.80	48.57	51.84	50.40	60.79
NMF	48.98	58.15	48.26	51.83	50.20	60.92
CC	47.55	64.94	41.53	52.56	46.77	68.68
SI-PLCA	50.17	48.73	62.19	48.08	57.61	48.57

Table 6: Label results for the four different algorithms (C-NMF, NMF, SI-PLCA, and CC) applied to two different datasets: ISO-Beatles (top) and SALAMI (bottom). Additionally, the reported results for the SF algorithm are also shown for the ISO-Beatles.

2.3.2 Structure Evaluation

The labeling task using the Pairwise Frame Clustering (F-measure $\mathbf{Pw}_f$, precision $\mathbf{Pw}_p$, and recall $\mathbf{Pw}_r$) and the Normalized Entropy Scores (F-measure $\mathbf{S}_f$, over-segmentation $\mathbf{S}_o$, and under-segmentation $\mathbf{S}_u$). These are the standard metrics used in the music segmentation task of MIReX (Smith and Chew, 2013), which were introduced in Chapter II. The results are shown on Table 6.

Again, PCPs outperform Tonnetz in the subtask of structural analysis for all the algorithms tested in our experiments. As aforementioned, this

code and the non-trimmed version with 0.5 second tolerance has not been yet reported in any other publication.

method is not capable of labeling two similar segments that are key transposed, while SI-PLCA is specifically designed to solve this problem. These datasets might contain a relatively small number of tracks with key transpositions, due to the fact that C-NMF, which is not key transposition invariant, outperforms SI-PLCA. Regardless, $\mathbf{Pw}_f$ can prefer algorithms like SI-PLCA that tend to undersegment, which is one of the motivations why the normalized entropy scores were proposed (Lukashevich, 2008). Consequently, a trend can be observed in which the entropy scores tend to be higher than the Pairwise Frame Clustering ones in all the algorithms, except in SI-PLCA. Nevertheless, C-NMF obtains higher results when compared to the other algorithms that aim to identify homogeneous regions, always outperforming the NMF version. Finally, it can be seen how SF still outperforms these methods in terms of structural analysis.

2.3.3 Discussion

While the boundary values for the CC algorithm are comparable to the ones reported in its original publication (Levy and Sandler, 2008), the ones obtained for the structural analysis for CC and SI-PLCA are significantly lower (around 10% decrease in both algorithms). Since their open source implementations are used to compute the results, this difference might exist due to the fact that different parameters have been used when computing the audio features (the ones obtained using Essentia). Those values might be optimized to their specific feature extraction process and do not generalize well. This shows the dependency and importance of feature extraction, which is likely to impact not only music segmentation tasks, but the rest of MIR tasks that rely on these features. Another reason might be due to the more accurate method used

to compute the structural evaluations in `mir_eval`, which uses a constant frame rate of 10Hz instead of the beat-synchronous frames. This difference is extensively discussed in the original `mir_eval` publication (Raffel et al., 2014).

C-NMF follows a stochastic process, so it is prone to fall into local minima. A good number of iterations to run, found experimentally, is around 30 for C-NMF and 100 for NMF, since, C-NMF is more consistent as previously discussed in Section 2.2.2. The features used in these experiments are not key transposition invariant, and it should be noted that the key-invariant SSM could be computed as the one used to discover patterns in the previous chapter and used as input to the C-NMF process, therefore potentially increasing the results of the algorithm, but significantly increasing its running time. In the next section a novel method that can efficiently label segments in a key-transposition invariant manner is discussed.

C-NMF is considerably faster than SI-PLCA or the regular NMF because of the fewer number of iterations required. It would be interesting to formally compare the speed of each of these algorithms in the future, but it is already worth mentioning that SI-PLCA takes over 1000 seconds to run on the ISO-Beatles dataset, while it only takes 170 seconds with the C-NMF approach. Computational efficiency is important when running this sort of algorithms over large datasets, and the following algorithm also successfully scales at larger amounts of data.

3 2D Fourier Transform Magnitude Coefficients

In this section a novel approach to label music segments by using 2D-Fourier Magnitude Coefficients (2D-FMCs) is presented. Recently, these coefficients

have proven to be an efficient solution to the task of large-scale cover song identification (Bertin-Mahieux and Ellis, 2012; Humphrey et al., 2013) because of their interesting inherent characteristics: key transposition and phase shift invariance. By aggregating 2D-FMCs into fixed-size patches representing full tracks, the comparison between tracks becomes fast and trivial. Analogously, and as a novel process, various methods are explored to obtain a set of segment-synchronous 2D-FMCs that can be used to characterize the similarity between segments of a given track, and to group those segments using k-means clustering. This results in a simple and computationally inexpensive process (as opposed to (Mauch et al., 2009b) or (Weiss and Bello, 2011)). Methods to estimate the optimal k are also discussed, and main components of this approach are systematically evaluated, resulting in performance similar to current state of the art.

3.1 2D-FMCs in Music Segment Similarity

In Western popular music, segments representing the same music section are likely to have common harmonic or melodic sequences (e.g., phrases, melodic lines, riffs, chord progressions), which are often played at different tempi, instrumentation and dynamics, and are flanked with repetitions and ornaments (that could cause phase shifts in the pattern), or even at different keys. Here it is detailed how beat-synchronous 2D-FMCs are invariant to these changes and therefore can be effective to label the different segments of a given piece.

3.1.1 Audio Features

As in the previous section, and similarly to other works (e.g., (Serrà et al., 2014; Weiss and Bello, 2011)), the proposed algorithm is solely based on har-

monic representations, which have proven to be a relevant musical aspect when segmenting musical pieces, especially for Western popular music (Smith et al., 2013).

The same beat-synchronous PCP features computed using Essentia are used as input to this algorithm. Tonnetz are not considered here since, as opposed to PCPs, a rotation of a Tonnetz feature vector does not directly translate as a key-transposition, and thus the transposition invariance property would be lost. Once the PCP features have been computed, they are segmented using the boundaries that define the sections of a given track. These boundaries can be automatically estimated using existing methods like the one described in the previous section (in this case, a custom implementation of the approach described in (Serrà et al., 2014), which yields better boundary results).

3.1.2 Computing the 2D-FMC Segments

By computing the magnitude 2D-Fourier transform of a sequence of beat-synchronous PCP features, three main characteristics are achieved:

(i) Key transposition invariance: Obtained by computing the Fourier transform on the pitch dimension and discarding the phase, since a key transposition appears on the PCP features as a rotation of the pitch distribution.

(ii) Phase shift invariance: Analogously as the previous point, by taking the Fourier transform, this time over the time dimension, and discarding the phase this time shift invariance of harmonic motives within a segment is obtained.

(iii) Local tempo invariance: This is an inherent property of using beat-synchronous features, as explained in Chapter II.

The 2D-Fourier transform, applied to the 2D signal $x_i \in \mathbb{R}^{M \times N}$, is defined as follows:

$$X_i(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_i(m,n) e^{-2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)} \tag{27}$$

where x_i is the i-th harmonic segment of a given track, M is the dimensionality of the harmonic feature vector (i.e., 12 for PCPs), and N is determined based on one of the strategies described below.

The goal of this stage of the process is to produce *segment-synchronous* feature vectors of the same dimensionality $M \times N$. However, different segments of a given track will have different lengths, requiring some form of segment length normalization in our analysis. Three different strategies are explored:

- **Maximum Window Size**: In this setup, N is set to the maximum length of the set of harmonic segments that constitute a track. Since most of the segments will be less than N, the segments are zero-padded before obtaining the 2D-FMC. The zero-pad operation is performed across the time dimension, resulting in an interpolated version of the patch of length N, which makes the comparison with other patches possible.

- **Minimum Window Size**: Another approach is to set N to the smallest segment size of all the harmonic segments of a given track. The majority of the harmonic segments will be greater than N, so grouping the longer segments into this smaller N is needed. To do so, the segments are

divided into 2D-FMC patches of size N with a hop size of one beat and aggregate them into a single patch of length N. Three different types of aggregation are considered: mean, median, and maximum.

- **Fixed Window Size**: In this case, a specific size for N is chosen, computing as follows: If the harmonic segment size is less than N, then zero-pad as in the maximum segment type. On the other hand, if the harmonic segment size is greater than N, then the longer segment is divided into smaller patches and they are aggregated using the mean, median or maximum as in the minimum segment type.

3.1.3 Clustering the 2D-FMC Segments

Before clustering, the logarithm of the patch is taken such that the weight of the DC component is diminished and the higher frequencies are emphasized, as it empirically yields better results in initial experiments. The symmetry of the 2D-FMC is also exploited by removing half of the coefficients.

k-means clustering with Euclidean distance is used on the segment-synchronous 2D-FMC patches. Further, to validate the quality of each partition, the Bayesian Information Criterion (BIC_k) is employed, which is defined as follows:

$$\mathrm{BIC}_k(S) = L - \frac{p \log(N)}{2} \tag{28}$$

where $S \in \mathbb{R}^{B \times M \times N}$ is the set of B 2D-FMC segment-synchronous patches, p is the number of free parameters of the system (which in our case is the sum of k classes, $N \times k$ centroid coordinates and the variance estimate σ^2 of the

partition), and L is the log-likelihood of the data when using k. Formally:

$$L = \frac{-N \log(2\pi) - NM \log(\sigma^2) - (N - k)}{2} \tag{29}$$

More information on this model can be found in (Pelleg and Moore, 2000). k-means is run with various k and the knee point detection method (Zhao et al., 2008) in BIC_k is used in order to estimate the most optimal k.

3.1.4 Illustrating the Process

In Figure 17 an example of this method is depicted with the song "And I Love Her" by The Beatles. The beat-synchronous PCPs matrix (top-left), the segment-synchronous 2D-FMC patches (bottom-left), and the normalized Euclidean distance between each pair of 2D-FMC patches (right) are shown. The segments S (solo) and V4 (verse 4) are key-modulated versions of segments V1, V2, and V3. This modulation is marked with an arrow in the beat-synchronous PCPs matrix, but disappears in the 2D-FMC representation, which successfully makes these five segments close to each other as shown by the self-distance matrix. The bridge (B) is harmonically different to the rest of the segments, which is also captured in the self-distance matrix, while the intro (I) and outro (O) share harmonic parts and, even though they have different time lengths, are grouped closer to each other.

This method estimates $k = 3$ unique labels for this track (I+O, V+S, and B). However, the ground truth indicates 5 unique labels (I, V, B, S, and O). Harmonically, it makes sense to have only 3 unique labels, but the timbre (e.g., for the guitar solo) and the placement of the segments (e.g., intro and outro) also have a relevant role in music segmentation. This, plus the inherent

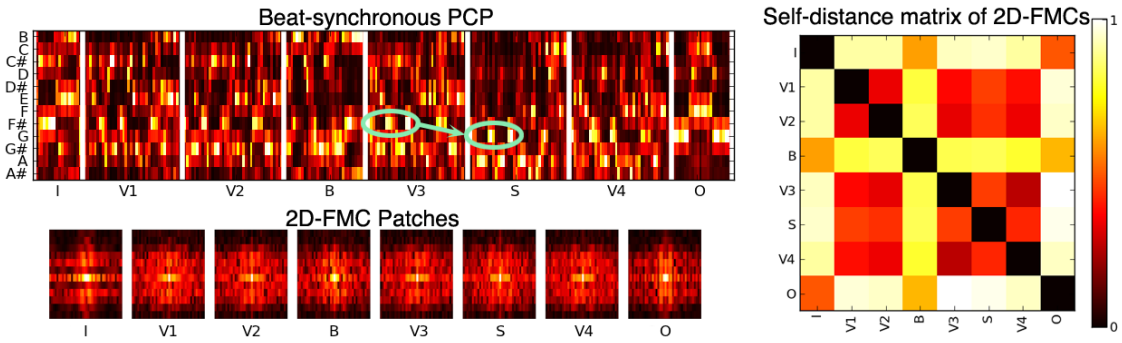


Figure 17: Example of the similarity between 2D-FMC patches representing sections of the song "And I love Her" by The Beatles. The beat-synchronous PCPs features are on the top-left, segmented with the ground truth segments by vertical white lines. The key transposition between V3 and S is marked. On the bottom-left the 2D-FMC patches are shown for each of the segments. On the right, the similarity between 2D-FMC patches is shown using the normalized Euclidean distance.

subjectiveness of the task, makes this problem remarkably difficult. In fact, it has been shown that it is unlikely that two people would manually annotate a specific dataset identically (Serrà et al., 2014; Smith et al., 2011; Bruderer et al., 2006b). Efforts towards improving these annotation issues in music segmentation will be discussed in Chapter VI.

3.2 Experiments

In this section the goal is to find, via experimentation, the optimal parameters of the system: the segment-synchronization strategy, and the number of unique labels k. To do so, and analogously to the previous algorithm, The Beatles dataset published by Isophonics is used. The focus is on this dataset rather than SALAMI because most published algorithms report their numbers on The Beatles dataset, and thus a comparison against these published results will be possible. Additionally, results for the SALAMI dataset are also reported here to facilitate comparisons in the future.

3.2.1 Evaluation

To evaluate the results the same two measures to evaluate structural grouping used in C-NMF —and discussed in Chapter II— are used: the pairwise frame clustering (F-measure $\mathbf{Pw}_f$, precision $\mathbf{Pw}_p$ and recall $\mathbf{Pw}_r$), and the normalized entropy scores (F-measure $\mathbf{S}_F$, over-segmentation $\mathbf{S}_o$ and under-segmentation $\mathbf{S}_u$). The former evaluation is more sensitive to boundary positions, while the latter strongly penalizes randomly labeled clusters, as discussed in (Lukashevich, 2008). Regardless, the former metric is maintained (i.e., pairwise frame clustering) for comparison purposes. Each presented result is the average of 10 different runs, since k-means is sensitive to initialization.

3.2.2 Optimal Segment-Synchronization Strategy

The annotated boundaries and the real number of unique labels k from the ground truth for each track are used here to experimentally determine the best segment-synchronization strategy. The algorithm is run with the three different strategies discussed above: maximum, minimum and fixed. For the minimum and fixed types, three different types of aggregation are also explored: median, mean, and maximum. Finally, for the fixed strategy, a window size of $N = 32$ is used (i.e., 8 bars in $\frac{4}{4}$ time signature, which is common in popular music), since it empirically yielded better results when compared to other multiples of 4.

The results are shown in Table 7. As it can be seen, the best performance is given by the maximum window size type, with a $\mathbf{Pw}_f$ of 81.59% and $\mathbf{S}_f$ of 87.31%, which defines the upper bound of the system's performance. This strategy outperforms all other strategies tested. One possible reason is that, by

bypassing aggregation and including all information within each segment, this strategy captures important low-frequency periodicities that are characteristic of the segments, e.g., sub-sequence repetitions.

Ntype	Aggr.	$\mathbf{Pw}_f$	$\mathbf{Pw}_p$	$\mathbf{Pw}_r$	$\mathbf{S}_f$	$\mathbf{S}_o$	$\mathbf{S}_u$
Max	–	**81.59**	79.90	85.57	**87.31**	90.18	85.48
Min	median	76.35	73.73	81.13	83.10	85.59	81.31
	mean	78.04	75.08	82.71	84.05	86.53	82.15
	max	76.88	75.11	80.23	83.42	84.99	82.47
Fixed	median	77.64	75.70	83.18	84.39	88.14	82.24
	mean	79.17	77.16	84.27	85.20	88.54	83.27
	max	79.55	76.02	86.42	85.75	90.11	82.91

Table 7: Results of the system when using the boundaries and the real k from the ground truth.

3.2.3 Estimating k

In this subsection the goal is to estimate k (number of unique segments per track) in the most optimal way, while still using the ground truth boundary annotations. By examining the ISO-Beatles dataset, it is observable that the median k is 6, with a histogram peak at $k = 5$ and the mean at $k = 5.57$ (see Figure 18, blue bars). The algorithm is run with five different $k = \{3, 4, 5, 6, 7\}$. Unsurprisingly, the results in Table 8, show that best performance is reached when $k = 6$, closely followed by $k = 5$. Note that as k increases, the metrics related to under-segmentation $\mathbf{Pw}_p$ and $\mathbf{S}_u$ increase, and the metrics related to over-segmentation $\mathbf{Pw}_r$ and $\mathbf{S}_o$ decrease, as expected. These results are compared against the C-NMF method described in the previous section, which performs worse in all the metrics compared to 2D-FMC.

Results using the automated approach to estimate k are also reported for the SALAMI dataset on the bottom of Table 8. To put the results in

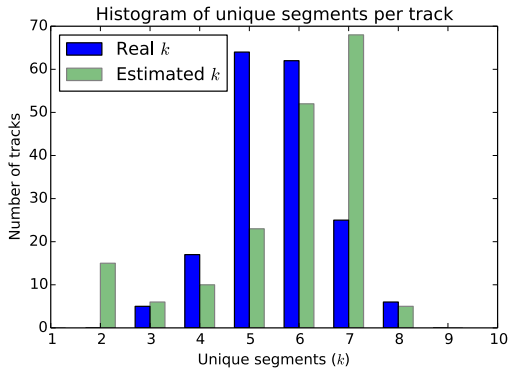


Figure 18: Histogram of the unique number of segments and the estimated ones in The ISO-Beatles dataset.

perspective, C-NMF and the other methods reported in that section are also shown in this table. These numbers were computed using the ground truth boundaries, and it is clear that the 2D-FMC method outperforms the others. Nevertheless, it is less clear whether this method yields state of the art results, since the author does not have access to the results of other algorithms when using human annotated boundaries.

In order to estimate k, the knee point detection method on the BIC, described in subsection 3.1.3, is used. The histogram of estimated k can be seen in Figure 18 in green, with results shown in Table 8. The approximated k tends to find more labels than the ones existing in the dataset. A way of alleviating this deficiency might be by using x-means (Pelleg and Moore, 2000). x-means uses a tree structure for increasingly large partitions, and only increases it if the difference between the BIC value of the new partition (with $k+1$ clusters) and the current one is greater than a certain threshold. This is left as possible future work. The results show how fixing $k = 6$ yields better F-measures, illustrating the difficulty of estimating k. Note that this estimation is made with a small number of 2D-FMC segment-synchronous patches (it

ISO-Beatles						
k	$\mathbf{Pw}_f$	$\mathbf{Pw}_p$	$\mathbf{Pw}_r$	$\mathbf{S}_f$	$\mathbf{S}_o$	$\mathbf{S}_u$
3	60.66	46.33	97.7	61.03	97.14	47.06
4	70.20	59.32	92.42	75.89	93.25	65.83
5	74.66	71.10	83.29	81.90	87.43	78.63
6	**75.38**	80.22	75.05	**83.88**	82.97	86.49
7	73.69	87.97	66.37	84.03	78.36	92.26
auto	73.83	75.39	79.89	80.22	85.36	80.24
C-NMF ($k = 6$)	71.21	75.55	70.47	74.96	74.27	77.25

SALAMI						
Algorithm	$\mathbf{Pw}_f$	$\mathbf{Pw}_p$	$\mathbf{Pw}_r$	$\mathbf{S}_f$	$\mathbf{S}_o$	$\mathbf{S}_u$
2D-FMC (k =auto)	**67.94**	72.65	72.01	**74.72**	76.57	79.49
C-NMF ($k = 6$)	66.63	70.55	70.24	70.42	71.96	75.57
CC (Levy and Sandler, 2008)	63.46	74.11	61.02	66.98	64.17	78.34
SI-PLCA (Weiss and Bello, 2011)	63.94	60.92	76.09	55.39	63.15	60.55

Table 8: Results of the system when using different k (fixed and auto) while using ground truth boundaries.

is uncommon for a track to have more than 15 segments), which likely has a negative effect on clustering. One idea is to obtain 2D-FMC patches for every beat (with a fixed number of beats for each patch and a hop size of one) in order to have a greater number of patches. Even though fixing k can also be interpreted as overfitting the dataset, it is not an uncommon practice (Kaiser and Sikora, 2010; Nieto and Farbood, 2013b), and therefore in the last experiments both fixed and automatic k are used.

3.2.4 Estimated Boundaries

The boundary method described in (Serrà et al., 2014) was implemented here in order to estimate the boundaries that will be used by the 2D-FMC method. This algorithm to extract boundaries yields some of the best results for some of the metrics in this dataset. This is only challenged by (McFee and Ellis, 2014b) when using a 0.5 seconds window instead of a 3 seconds one.

In Table 9, 2D-FMC is compared with a number of state of the art techniques in the literature. Imprecise boundary estimations make the 2D-Fourier transform not to capture the lower frequencies caused by the longer periodicities of the segment, which worsen the results as seen in subsection 3.2.2. Lukashevich showed that poorly estimated boundaries greatly penalize $\mathbf{Pw}_f$ compared to $\mathbf{S}_f$ (Lukashevich, 2008), which may explain why the differences between the two increase for 2D-FMC compared with the previous experiments. This illustrates a drawback of this method: its high sensitivity to good boundary estimation, as clearly illustrated by a lower $\mathbf{Pw}_f$ than those of the other approaches in the table. On the other hand, when observing the entropy scores, it can be seen that 2D-FMC obtains close to state of the art results for $k = 6$ with an $\mathbf{S}_f$ of 68.15% (only improved by Mauch's technique (Mauch et al., 2009b)). In the original publication (Nieto and Bello, 2014), results were reported using audio features computed using a custom implementation instead of Essentia, and state of the art results were obtained (there was a small percentage increase that surpassed Mauch's numbers). However, the emphasis here is intended to be on the technique itself rather than on the bias that the audio feature computation might introduce. Moreover, by using the same exact features than in C-NMF, CC, and SI-PLCA (i.e., those computed using Essentia), this bias can be ignored, thus better comparing and assessing the qualities of these methods.

3.3 Discussion on Efficiency

Having the audio features pre-computed, this method, when using human annotated boundaries, takes approximately 53 seconds when estimating k, and 13 seconds when fixing k to compute all the 179 tracks of the Beatles

k / Others	$\mathbf{Pw}_f$	$\mathbf{Pw}_p$	$\mathbf{Pw}_r$	$\mathbf{S}_f$	$\mathbf{S}_o$	$\mathbf{S}_u$
4	55.73	52.11	64.00	61.61	68.71	57.33
5	56.59	59.52	57.26	64.16	64.91	65.00
6	**57.62**	68.66	51.88	**68.15**	65.84	72.01
7	51.54	69.60	42.56	64.28	57.43	74.49
auto	56.01	70.05	49.91	66.80	62.72	74.14
C-NMF	53.53	58.29	52.65	57.20	55.82	60.63
CC (Levy and Sandler, 2008)	49.18	62.91	41.06	56.50	50.36	66.50
SI-PLCA (Weiss and Bello, 2011)	49.36	42.67	65.17	48.08	62.28	42.67
Grohganz (Grohganz et al., 2013) *	68.0	71.4	68.8	–	–	–
Kaiser (Kaiser and Sikora, 2010) *	60.8	61.5	64.6	–	–	–
Mauch (Mauch et al., 2009b)	66	61	77	**69.48**	76	64
Serrà (Serrà et al., 2014)	**71.1**	68.1	78.7	–	–	–

Table 9: Results of the system when using different k (fixed and auto) and estimated boundaries. *: these results are computed using the ISO-Beatles dataset.

dataset. In comparison, C-NMF takes 63 seconds, CC takes 566 seconds, and SI-PLCA takes 375 seconds under the same circumstances (i.e., pre-computed audio features and using ground truth boundaries)*. This reduced running time compared to these other techniques was expected as the 2D-FMC representation in music was initially introduced to solve large-scale cover song identification (Bertin-Mahieux and Ellis, 2012). Working on the frequency domain, and thanks to the duality of the convolution theorem (Smith, 2007), the computation of the convolution for the key-transpositions and the time-shifts is avoided, which significantly speeds up the process.

4 Summary

In this chapter two novel methods to discover large scale non-overlapping sections of musical pieces were presented: C-NMF and 2D-FMC. These methods

*These running times were computed on a 2013 MacBook Pro, parallelizing the process across its four different cores.

can be classified under the widely discussed MIR task of music segmentation. C-NMF attempts to identify the boundaries and label the segments that fall under the category of *homogeneous* by using a modified version of the common machine learning technique of Non-negative Matrix Factorization. Here it was shown how this technique obtains state-of-the-art results compared to other techniques that also identify homogeneous segments. The 2D-FMC method aims at labeling any type of music segments by using an audio representation that is invariant to key transpositions and phase shifts in time, making similarity computations on chroma features both robust and efficient. When evaluating this technique, it was discussed how it can reach state-of-the-art results on certain metrics, surpassing C-NMF in terms of labeling.

The feature representations have shown to play an important role on these algorithms. This process makes use of many parameters (e.g., frame size, window size, hop size, window type, coefficients for MFCC, number of octaves for the PCP) that significantly impact the final results of the algorithms, as it has been discussed when comparing the scores of open source algorithms such as CC and SI-PLCA against their original publications. It has also been shown that PCPs outperform Tonnetz in C-NMF. This contrast with the findings of the music summarization algorithm of Chapter III, where these features were represented *symbolically* using vector quantization. This quantization process might have a negative impact when using PCPs, since they encode more information than Tonnetz, which therefore should be harder to quantize. On the other hand, only PCPs have been used in the 2D-FMC method, given that its input needs to be a representation in which rotations in both dimensions are meaningful (and desired). Moreover, by using Essentia, it was possible to easily share and reproduce the features across multiple algo-

rithms, which were used here in order to assess C-NMF and 2D-FMCs. It has been shown how the usage of Essentia features tends to yield lower results than the ones reported in original publications in which custom implementations are used for their feature computation. By being able to input the exact features when comparing the quality of various algorithms, this bias introduced by the computation of the features has been overcome.

Open source implementations for both C-NMF and 2D-FMC can be found in the Music Structure Analysis Framework*. Also included in this framework are the CC and SI-PLCA algorithms (forked from their original open source implementations), which should be useful in terms of reproducing the results reported here. An extensive description of this framework will be presented in the following chapter.

It remains to be seen how reliable these standard evaluations used in this chapter are, especially when having in mind the possible perceptual differences when discovering the structure of a musical piece. In the following chapter, an experiment will be designed in order to explore the amount of agreement between listeners when annotating the different sections of a track. This will lead to the proposals in Chapter VI of evaluations that aim at reducing the inherent subjectivity problem of music segmentation.

* `https://github.com/urinieto/msaf`

CHAPTER V

DATA COLLECTION METHODOLOGY

1 Introduction

In the two previous chapters various novel methods to identify different structural aspects of music represented with audio waveforms were presented. More specifically, the last two algorithms were focused on the music segmentation task of MIR, a well-established task and widely discussed topic. Music segmentation algorithms are evaluated using standard metrics that compare the output of the algorithms against human annotations used as reference (a.k.a. *ground-truth*). Following this notion, the results employing the traditional datasets and metrics of this task were discussed. In Chapter II it has been shown, however, that humans do not tend to agree on the perception of music segment boundaries (Bruderer et al., 2009), and therefore relying on annotations that are produced by just one person might yield unreliable results when evaluating boundary algorithms.

In this chapter a methodology to collect multiple human annotations is described in order to obtain a challenging dataset with several segmentation references per track, which will enable the quantification of inconsistency between human annotations. To do so, a diverse set of algorithms are compiled together in a structural framework that facilitates the identification of the easiest and hardest music pieces for automatic segmentation. This framework

uses a novel format called JAMS (JSON Annotated Music Specification) that allows the storing of multiple annotations for several MIR tasks in a single file, which becomes particularly helpful in this work. Finally, a study with human subjects is conducted in the interest of (i) collecting multiple segmentation annotations for the challenging tracks selected by our framework, (ii) analyzing the degree of subjectivity in the task of music segmentation, and (iii) designing more perceptually relevant segmentation evaluation methods that could potentially combine multiple annotations in order to address the problems of subjectivity. The latter two points of this study, which challenge the notion of music segmentation ground-truth produced by a single human annotation, will be detailed in the next chapter. The methodology that will lead us to this final goal is discussed below.

2 Music Structure Analysis Framework

As seen in Chapter II, there are many methods to automatically extract the music structure of a given track. However, in most cases, reproducing these results is not straightforward. Authors of published algorithms usually do not share their source code, and sometimes the limitation of publication lengths of conference proceedings (sometimes no more than 4 pages) results in multiple hyper parameters left unexplained. Moreover, the audio features used might introduce a bias that should not be taken into account when comparing such methods, which could be avoided by inputing the same pre-computed features to all the algorithms.

This motivates the central ideas behind the music structure analysis framework (MSAF), an open source project that aims to facilitate the anal-

ysis, execution, comparison and evaluation of several music segmentation algorithms. In this section we detail the aspects of MSAF in the hopes that it becomes a significant piece of software not only for the MIR community, but also for other researchers (e.g., musicologists, music cognition practitioners) who focus on structure analysis. This framework, implemented in Python, is available for public download*, and is pre-configured to run on a specifically designed dataset that uses a novel file format called JAMS capable of storing multiple annotations for diverse MIR tasks in a single file. In the following subsections the audio features, the segmentation algorithms, the JAMS format, and the datasets used in MSAF are discussed.

2.1 Audio Features in MSAF

The audio features in MSAF are shared by all the algorithms, and, following the work presented in Chapter IV, the open source package Essentia (Bogdanov et al., 2013) is used to compute them. More specifically, Essentia is employed to obtain PCPs and MFCCs (presented in Chapter II). The variant chosen by Essentia for their PCPs is what is known as the harmonic pitch class profiles (HPCPs), which removes noise from the audio signal in order to produce cleaner and therefore more harmonically relevant features. A detailed explanation of this process can be found in (Gómez, 2006). The method that Essentia employs to compute the MFCC features is called MFCC-FB40, which uses a filterbank of 40 bands from 0 to 11000Hz, takes the log value of the spectrum energy in each Mel band, and then performs a discrete cosine transform of the 40 bands to procure 40 Mel coefficients from which only 13

* `https://github.com/urinieto/msaf`

are finally used. This process is detailed and compared with other MFCC implementations in (Ganchev et al., 2005). Additionally, tonal centroids (or Tonnetz), which were discussed in Chapter II, are also available as an alternative harmonic representation, and they are computed using the geometric transformations described in (Harte et al., 2006).

By default, the audio frames are windowed using a Blackman-Harris window of 62dB (Smith, 2010) since it has been shown to yield good results when computing audio features for structural segmentation (Serrà et al., 2014). The window length is fairly large, of approximately 185 milliseconds, and the hop-size for overlapping windows is half this time. This may be advantageous for smoothing out irrelevant local variations, which is often an appropriate property in music structure analysis. Also by default, MSAF accepts audio sampled at 11025Hz, which significantly speeds up the computation of the features without noticeable loss of relevant structural information. All of these parameters (i.e., the number of MFCC coefficients, the type, length and hop-size of the analysis window, and the sampling rate) can be easily tuned in the configuration file of MSAF.

Finally, the multi-feature beat tracker (Zapata et al., 2013) included in Essentia is employed to estimate the beats. This algorithm obtained considerably high results in MIREX 2012 and 2013, and it is based on the idea of identifying the mutual agreement among multiple beat trackers and automatically determining the one that is most likely to yield the best results. The frames are synchronized to the beats by aggregating them using the algorithm proposed in (Ellis and Poliner, 2007) (resulting in beat-synchronous features as described in Chapter II). This synchronization considerably reduces the length of the input features, while maintaining enough resolution for the algo-

Similarity	Boundaries
2D-FMC (see Chapter IV)	C-NMF (see Chapter IV)
C-NMF (see Chapter IV)	Checkerboard (Foote, 2000)
CC (Levy and Sandler, 2008)	CC (Levy and Sandler, 2008)
SI-PLCA (Weiss and Bello, 2011)	OLDA (McFee and Ellis, 2014b)
	SF (Serrà et al., 2014)
	SI-PLCA (Weiss and Bello, 2011)

Table 10: List of algorithms, sorted by type, that are available in MSAF.

rithms to identify and label the segments, since arguably all large scale segment boundaries should start and end at a beat-level. However, beat trackers are far from being perfect, and tracking errors might negatively impact segmentation results. Therefore, MSAF can also accept frame-synchronous features, even if the beat-synchronous ones are the default input. Additionally, human annotated beats (if available) can also be used as input in order to assess the influence of beat-trackers on segmentation results.

2.2 Algorithms in MSAF

MSAF includes implementations of both subtasks required for music structure analysis: boundary finding and segment labeling. The end user can choose any of the possible combinations of these two subtasks to better explore the different results of the algorithms and the impact of the combination of boundary and structural algorithms. The list of the different algorithms that are currently implemented in MSAF is provided in Table 10.

The most popular open source algorithms for music segmentation are included in MSAF, which are, to the best of my knowledge: Constrained Clus-

tering (CC*), Shift Invariance Probabilistic Component Analysis (SI-PLCA†), and Ordinal Linear Discriminative Analysis (OLDA‡). Additionally, MSAF also contains the two structural segmentation methods described in Chapter IV: 2D Fourier Magnitude Coefficients (2D-FMC) and Convex Non-negative Matrix Factorization (C-NMF). Even though this set of algorithms already covers a wide spectrum of segmentation methods, two other relevant techniques that obtain particularly good results for boundary identification are further included: the checkerboard-like kernel novelty-based method defined by Foote (Foote, 2000), which is still significantly effective despite having been published almost fifteen years ago; and the Structural Features method published by Serrà et al. (Serrà et al., 2014), which obtains state-of-the art results in some of the evaluation metrics. These two additional algorithms had to be implemented based on their original paper publications since their authors have not released open source versions. Consequently, there might be slight differences from their original results due to implementation variations, such as features computation, or specific values for certain parameters left unexplained. The framework is written such that new algorithms can be added fairly easily, thus encouraging researchers to include implementations of other algorithms in this open source project. In order to facilitate the comparison between algorithms, MSAF provides plotting methods to visually assess the boundaries and labels (see Figure 19 for an example).

MSAF includes the evaluation methods described in Chapter II and it uses the `mir_eval` package (Raffel et al., 2014) described in the same chapter.

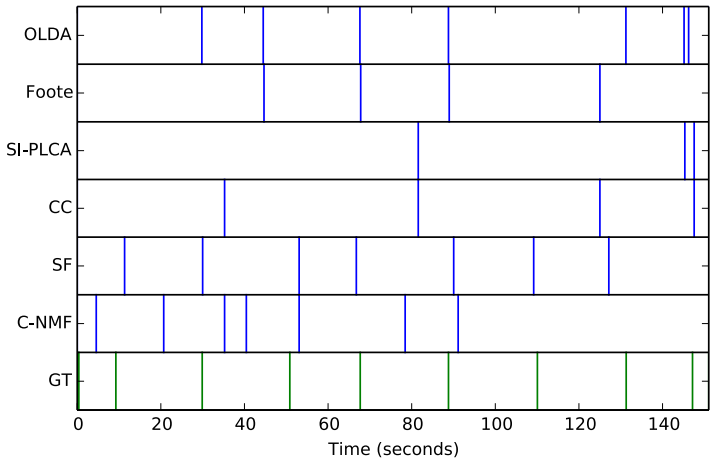
---

*`http://code.soundsoftware.ac.uk/projects/qm-dsp`

†`http://ronw.github.io/siplca-segmentation`

‡`https://github.com/bmcfee/olda`

(a) Boundaries Comparison



(b) Labels Comparison

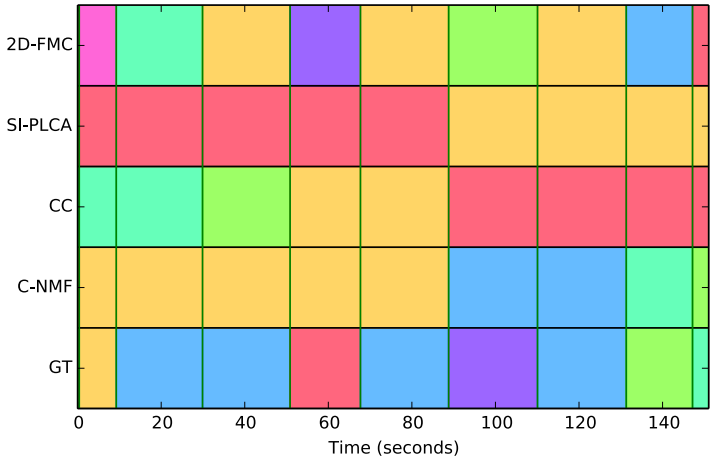


Figure 19: Comparison of the boundaries (top) and labels (bottom) between the outputs of all the algorithms contained in MSAF for the track *And I Love Her* by The Bealtes. GT stands for Ground-Truth.

This results in a straightforward analysis and comparison between algorithms using not only the standard metrics, but the rest of the evaluations that are not commonly reported in original publications (e.g., using 0.5 instead of 3 second windows for the hit rate for the boundaries, or trimming the first and last boundary as these are trivial to be retrieved and could potentially bias the

final scores). In terms of labeling metrics, the normalized conditional entropy scores are not reported as often as the pairwise frame clustering measures, however with MSAF the comparison and computation of these scores with the included algorithms becomes effortless.

2.3 Storing Multiple Annotations: The JAMS Format

MSAF makes use of a novel format in order to facilitate the comparison between multiple segment annotations. This format, called JAMS (JSON Annotated Music Specification) (Humphrey et al., 2014), is based on the open standard, cross-platform and well-established JSON (JavaScript Object Notation) format originally used to transmit dictionary-like data objects in JavaScript. As opposed to other data file formats like XML, humans can read JSON files without previous knowledge of the format, and the vast majority of modern programming languages support JSON. Python, like JavaScript, has a built-in dictionary type that uses JSON for serialization without adding significant computation overhead when reading/writing these type of files, thus making JSON the ideal format for storing MSAF data.

It is common in segmentation to use the so-called *lab* format, which is simply a plain text file with a line-separated series of segments, each containing three tab-separated segment values: starting point, end point, and label. Only one annotation can be included in a lab file, making it challenging to compare multiple annotations, or even algorithm results for a single track. Moreover, useful metadata like, e.g., the annotator's name and contact information, the collection to which the file belongs, the artist and song names, or the creation date, can not be stored in a lab file, which results in a relatively complex file structure for collections with more than one annotation per track, requiring

additional files to store this meta information (e.g., see SALAMI (Smith et al., 2011)).

The JAMS format aims to solve these problems by adding the following three main characteristics:

i **Comprehensive annotation metadata**: Large and complex datasets like SALAMI (Smith et al., 2011), the billboard dataset (Burgoyne et al., 2011), or the million song dataset (Bertin-Mahieux et al., 2011), require content-specific metadata associated with the collection of annotations. JAMS is designed to support the inclusion of this relevant information at a file level.

ii **Multiple annotations for a given task**: JAMS allows any number of annotations to be included in a single file, therefore avoiding complex folder structures like in SALAMI. This is especially relevant for the analysis and comparison of multiple annotations, a topic that will be further discussed in the next chapter.

iii **Multiple concepts for a given signal**: MIR tasks are becoming more interconnected (e.g., chords and downbeats (Papadopoulos and Peeters, 2011), and chords and segments (Mauch et al., 2009b)), and it has been suggested to keep working towards this integration process (Vincent et al., 2010). Even though it is still common to propose algorithms for one task at a time, datasets may contain annotations for several tasks (e.g., Isophonics, which contains keys, chords, segments, beats, and downbeats (Mauch et al., 2009a)). With this in mind, the design of JAMS allows to store data for more than one task in a single file.

As an example, a diagram of a JAMS file is illustrated in Figure 20. Following the standard JavaScript notation of curly brackets (`{ }`) to denote objects (alternatively, dictionaries or structs), and square brackets (`[ ]`) to denote arrays, this example would be completely functional except for the ellipses "..." as continuation characters, which indicate that more information could be included.

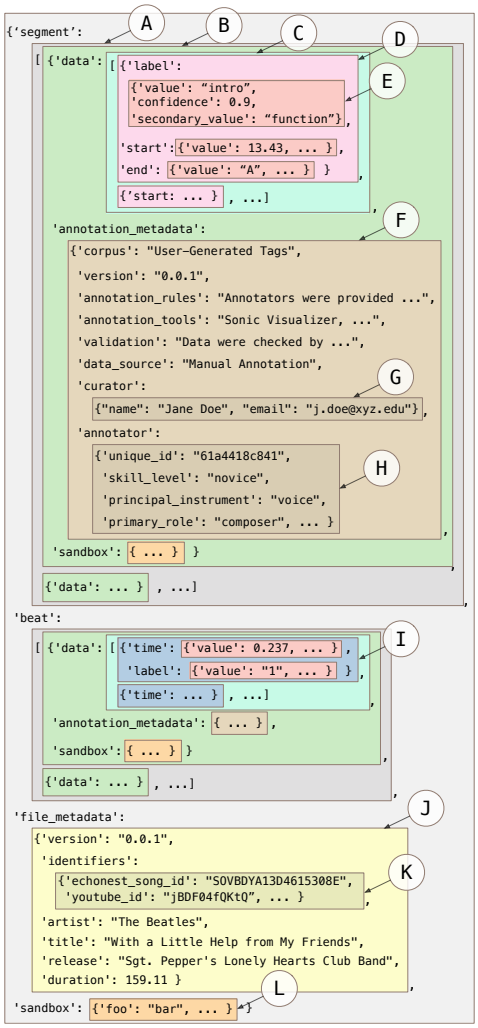


Figure 20: Diagram of the JAMS specification format.

In this example, the two tasks that MSAF employs are included: segments and beats. It can be seen that both tasks are essentially an array

of annotations (A), and each annotation (B) contains three fields: `data`, `annotation_data`, and `sandbox`. The `data` (C) is the actual annotation, and it is in fact an array of specific data types that depend on the task. For the music segmentation task, a *range* type (D) is used, which contains fields for `label`, `start`, and `end` (note that this type is useful for storing chord annotations as well). For each of these three fields, an *observation* (E) containing the fields `value`, `confidence`, and `secondary_value` will be used to store the actual value, the confidence level and an additional value respectively. This secondary value becomes especially useful in segmentation, since some annotations might be hierarchical (e.g., containing large and small scale levels of segmentation, or function names for the segments, cf. SALAMI). The annotation metadata (F) contains a set of fields to store information about how the data of this annotation was collected. Additionally, curator information (G), and annotator information (H) is included in the metadata. An unconstrained object to store extra information as needed is included at an annotation-level called the `sandbox`.

The next task to appear in this example, beats, is analogous to segments but with a different type of data. In this case, the *event* type is used (I), which only includes the fields of `time` and `label`. This type could also be useful in, e.g., downbeat prediction. Global metadata such as artist or track title will be included in the field `file_metadata` (J), which includes identifiers (K) like The Echo Nest ID or the YouTube ID in order to know the exact fingerprint of the piece. Lastly, the JAMS object also contains a global sandbox object (L) to be used as needed. In this way, the specification carves out such space for any unforeseen or otherwise relevant data. A full example of a JAMS file can be seen in Appendix A.

MSAF uses JAMS to store both human references and machine estimations, thus facilitating the comparison and analysis between multiple subjects and algorithms (and their combination, if needed). In the next subsection it is discussed how these files are organized at a folder level such that MSAF can successfully operate on them.

2.4 File Structure for Collections in MSAF

MSAF is designed to run on a collection of audio files, so that the analysis and evaluation of multiple files becomes effortless. These collections must follow a specific folder hierarchy, described as follows under the toy example of a collection named `my_collection`:

```
my_collection
├── audio ..... .mp3, .wav, .aif
├── estimations ..... .jams
├── features ..... .json
└── references ..... .jams
```

As the name indicates, the `audio` folder must contain all the audio files of the collection. By default, they can be of type `.mp3`, `.wav`, or `.aif`, which are standard file types for storing digital audio. In the `estimations` folder, the output of MSAF will be written using the JAMS format. For each audio track in the `audio` folder, one estimation file will be created (first run) or updated (further runs) containing the algorithm outputs with their specific parameters. The `features` folder will be populated by MSAF once it has run on this specific collection. The file type in this folder is plain JSON, containing all the audio features in MSAF described in section 2.1. Once the features are precomputed (this occurs automatically the first time MSAF runs on the

collection), the subsequent runs will be dramatically faster, since the features are invariant to the algorithms and therefore only need to be computed once. Finally, in the `references` folder the JAMS formatted human annotations are included, which will be used to evaluate the algorithms' performance.

Note that if no evaluation is needed, the `references` folder can be empty. Consequently, only the `audio` folder is strictly required for MSAF to be able to run on a collection (`estimations` and `features` folders are automatically populated). Moreover, by virtue of the JAMS format, once MSAF has run on the collection at least once, each folder (except perhaps the `references` folder, which can be empty) will contain the same number of files (with the same file names but different extensions), making this folder hierarchy easy to navigate and to reproduce. Finally, these folder names can be customized in the MSAF configuration file; the folder names discussed above are the default ones.

2.5 MSAF Operating Modes

Two operating modes are available in MSAF: *single file* and *collection*. When using MSAF in single file mode, the file structure described above is not necessary, only an audio file is in fact required. Therefore, the features will be computed every time a file is analyzed in single mode (no JSON file containing the features is saved during this process for the lack of folder structure), making the process slower than in collection mode. However, MSAF in single file mode can output plots like the ones in Figure 19, and it can also *sonify* the identified boundaries, which can be particularly useful for subjectively assessing the quality of the algorithm when no human references are available.

On the other hand, when operating in collection mode, MSAF requires

the path to a folder structured as described in subsection 2.4. As mentioned before, the features will be computed only during the first run, making the ensuing runs significantly faster. Moreover, thanks to the file structure of the collection, and if human references are available in the `references` folder, it will be possible to objectively evaluate the results of the MSAF algorithms on a given collection. All the metrics reported for the algorithms presented in Chapter IV, plus the median time deviations $\mathbf{D}_{E2A}$ and $\mathbf{D}_{A2E}$ described in Chapter II will be reported when using the default MSAF script to evaluate a collection. MSAF is designed to parallelize the processes of running and evaluating a collection over multiple cores. The number of cores to be used can be easily tuned when using MSAF in collection mode.

Finally, in order to assess the quality of the current implementation, the evaluation of the MSAF algorithms for all the presented datasets (Cerulean, Epiphyte, ISO-Beatles, and SALAMI) is reported in Appendix B.

3 Large Music Segmentation Dataset

For this work, a large dataset containing 2,157 tracks with human annotated music segmentation information was collected. This substantial dataset is composed of two well-known datasets (Isophonics and SALAMI) and two additional unpublished datasets (Cerluean and Epiphyte). Detailed information about these sets is provided in the next subsections.

3.1 Isophonics Subset

This dataset was collected by the Centre for Digital Music (C4DM) of Queen Mary University of London, and is available for public download*. It contains 300 annotated tracks of western popular music with segmentation information, including the entire Beatles catalog (the so-called ISO-Beatles used in Chapter IV), the greatest hits by Michael Jackson and Queen, and two additional albums by Carole King and Zweieck. Additionally, beat and downbeat annotations are also included for the ISO-Beatles subset, which can be exploited by MSAF. The annotations are stored using the *lab* format described in (Mauch et al., 2009a). The Beatles annotations for some musical aspects were initially collected by Alan Pollack, which were later revised and enriched by music experts at C4DM. The rest of the annotations were put together by musicologists and researchers at C4DM.

3.2 SALAMI Subset

The structural annotations for large amounts of music information (SALAMI) dataset, also used in the previous chapter, contains human annotations for 751 tracks and is extensively discussed in (Smith et al., 2011). These tracks are reasonably diverse and can be divided into five different classes of music: classical (16%), jazz (17%), popular (23%), world (16%), and live music (28%). Approximately two thirds of the annotations (66.31%) contain two different annotations per track, while the rest (33.69%) contains a single annotation. Moreover, three levels of structure are included in this dataset: small scale, large scale, and functional level. The file structure of SALAMI is relatively

* `http://isophonics.net/datasets`

intricate, where multiple files are used inside custom hierarchical folders to store the annotations using *lab* files —one file for each level of segmentation and annotation—, while a global metadata file in *csv* format is used to store the additional information like annotator name, artist name, or piece title, one row of metadata for each track. The annotations were collected by graduate music students using the Sonic Visualiser tool (Cannam et al., 2006).

3.3 Cerulean Subset

A company that would like to remain anonymous, referred here as *Cerulean*, gathered two annotations from two different subjects for 104 musical pieces. These pieces were manually selected to be particularly challenging in terms of segmentation, following a simple subjective and informal evaluation. The collection includes popular music (28%), classical (18%), jazz and blues (17%), world (6%), and rock (31%), which encompasses a relatively large selection of progressive rock (8%) and heavy metal (12%). The original format of the dataset is a custom JSON format, and the annotations were written by two graduate music students who made use of the Sonic Visualiser in order to simplify the task.

3.4 Epiphyte Subset

The last subset was also collected by a different anonymous company, which is named in this work as *Epiphyte*. 1,002 tracks compose this dataset, and each of these tracks contains a single segmentation annotation. Additionally, beat and downbeat information is also manually annotated for the entire dataset. As opposed to the Cerulean dataset, the audio files in Epiphyte are mostly relatively easy-to-segment pop songs (approximately 80%), with additional

rock tracks (20%) that are, subjectively, more challenging to segment. The original dataset is stored in a custom text format that is a minor variation of a *lab* file, where, for each track, one file is needed for each of the different features (segments, beats, and downbeats). The annotations were collected by musical experts associated with this anonymous company, aided by undisclosed software tools to facilitate the storage of the references in custom *lab* files.

3.5 Consolidating the Large Dataset

The four previously discussed datasets are put together in the large dataset of 2,157 tracks (see Figure 21a for a visual representation of the subsets distributed across the large set). To do so, the original data formats of the subsets are first parsed into the JAMS annotation, therefore having a consistent format for the consolidated dataset. Following the file structure defined in subsection 2.4, the JAMS annotations are placed in the `references` folder and the audio files in `audio`. The final distribution of the genres across the subset and the large dataset can be seen in Figures 21b and 21c, respectively. These figures show how pop-dominated this large dataset is, arguably because (i) pop music is the least challenging type of music in order to run segmentation algorithms on, and (ii) since this MIR task is still far from being completely solved, it is best to first simplify the problem as much as possible. Nonetheless, non-pop tracks still cover almost 40% of the dataset (i.e., over 850 tracks), which should result in a relatively challenging dataset for music segmentation.

Running MSAF on this substantial amount of information, and then evaluating the results across multiple algorithms, should help us determine a subset of the most challenging tracks from a machine analysis point of view. In the next section the creation of a reduced and more challenging dataset is

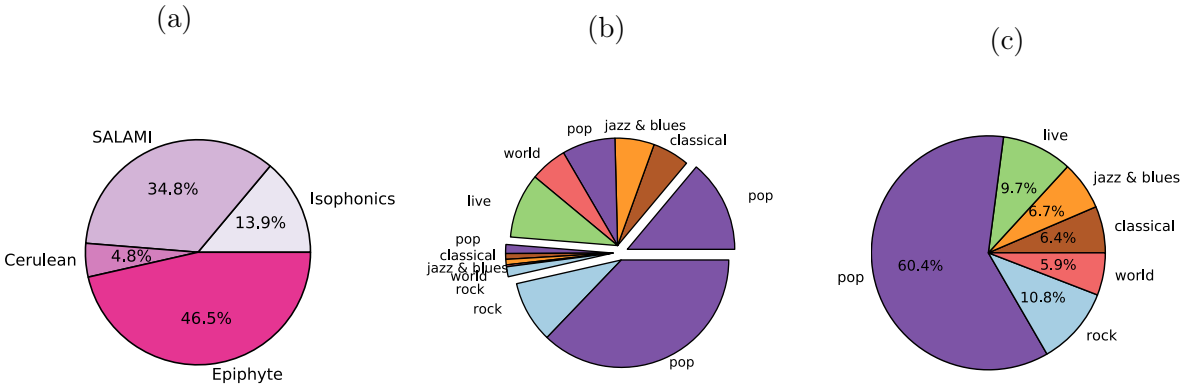


Figure 21: (a): Coverage of each subset in the consolidated large dataset. (b): genre information for each subset under the large dataset context. (c): Merged genre information for the whole large dataset.

detailed, along with how its tracks are analyzed by multiple subjects in order to assess the level of agreement (which can be interpreted as a measure of *subjectivity*) among musical experts when annotating music segments.

4 Collecting Multiple Annotations

Now that a framework containing multiple music segmentation algorithms is available and a large collection of human annotated segments is gathered, it becomes trivial to rank the tracks in this collection based on the segmentation evaluation metrics for each algorithm contained in the framework. Given the substantial amount of time needed to manually annotate music segments for a given track, it becomes impractical to assemble multiple annotations for the entire large dataset described in the previous section. Consequently, the goal here is to obtain a significantly smaller set, named *reduced dataset*, which will contain the tracks that yield a poor performance in a specific metric on the large dataset (i.e., challenging from a machine perspective), so that it can be

used to collect multiple human annotations for each track in order to analyze the degree of agreement between musical experts. In this work the focus will be on the challenging tracks since it can be argued that they are likely to encode more rich and ambiguous segment information which may explicitly reveal the extent of subjectivity in human annotators when analyzing them.

4.1 Reduced Music Segmentation Dataset

As seen in Chapters II and IV, the structural (or labeling) subproblem of music segmentation is strongly correlated with the quality of the retrieved boundaries. Therefore, in order to simplify the problem of discovering the most challenging tracks on a human annotated collection, the output of boundary algorithms is employed exclusively. More specifically, $M = 5$ of these algorithms contained in MSAF (Checkerboard, CC, OLDA, SF, and SI-PLCA) are run over the large collection of $N = 2157$ tracks. This yields a set of boundary results that can be stored in a matrix $B \in \mathbb{R}^{N,M}$, such that $\mathbf{b}_{ij} \in B$ represents the set of boundaries obtained for track $i \in [1, N]$ using algorithm $j \in [1, M]$. Each set of boundary results $\mathbf{b}_{ij}$ is evaluated using the standard Hit Rate with a 3 second window, since, in the literature, this metric is the most standard when assessing the quality of automatically retrieved boundaries. The results are averaged across algorithms, yielding a value that can be interpreted as the mean ground-truth precision for each of the tracks in the dataset. Formally:

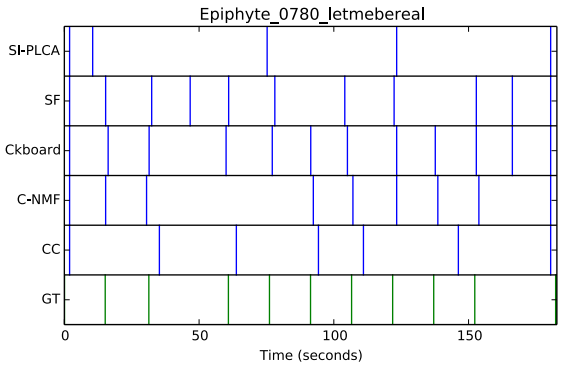
$$\text{MGP}_i(B, g) = \frac{1}{M} \sum_{j=1}^{M} g(\mathbf{b}_{ij}) \tag{30}$$

where g is the evaluation function, in our case the F-measure of the Hit Rate with a 3 second window ($\mathbf{F}_3$) described in Chapter II.

Thus, when sorting the list of $\text{MGP}_i, \forall i \in [1, N]$, it becomes easy to select the tracks that are more challenging by inspecting the bottom of this ranked list (sorted from highest to lowest score). From the bottom of the list, the tracks that only contain speech are removed (found in the SALAMI dataset, and which yield poor performances) and forty five hard tracks are chosen to be included in the reduced dataset. Moreover, the top five scoring tracks are added as a control group within the reduced dataset. This results in a reduced dataset of fifty tracks which will be helpful when collecting additional human annotations. As an example of agreement, in Figure 22 the estimated boundaries of various MSAF algorithms are plotted against the human annotated ground truth for two tracks: one that obtains high agreement (top) and another with poor agreement (bottom) with respect to the human references.

In Figure 23 visual information about the distribution of the subsets and genres across the reduced dataset is shown. Interestingly, the Epiphyte subset, which is intended to be fairly trivial to segment, does not appear in the challenging subset of the reduced dataset (see Figure 23a), suggesting that the methodology used to select these tracks aligns with the initial assumptions. Moreover, pop music is only represented by 15.6% of the tracks in the challenging subset (Figure 23b), which contrasts with the large dataset from which these tracks have been selected. On the other hand, the control tracks do not contain any piece from the Cerulean subset, which was designed to be particularly difficult for music segmentation algorithms. Moreover, four of the five control tracks belong to the 'pop' genre (Figure 23f), indicating that

(a) Good Boundary Agreement



(b) Poor Boundary Agreement

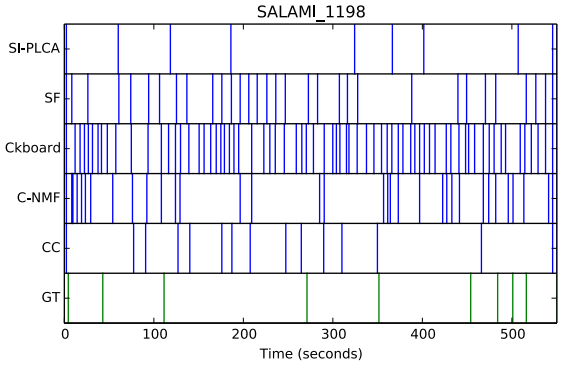


Figure 22: Agreement of estimated boundaries of multiple MSAF algorithms (blue lines) and the human annotated ground-truth (GT row of green lines).

music genres might have a strong impact on the performance of music segmentation algorithms, and reinforcing the hypothesis that pop tracks are simpler to segment.

4.2 Multiple Segmentation Annotations

The 50 tracks that comprise the reduced dataset were further segmented by multiple annotators. Five music students from the Steinhardt School at New Yotk University (four graduates and one undergraduate), volunteered to per-

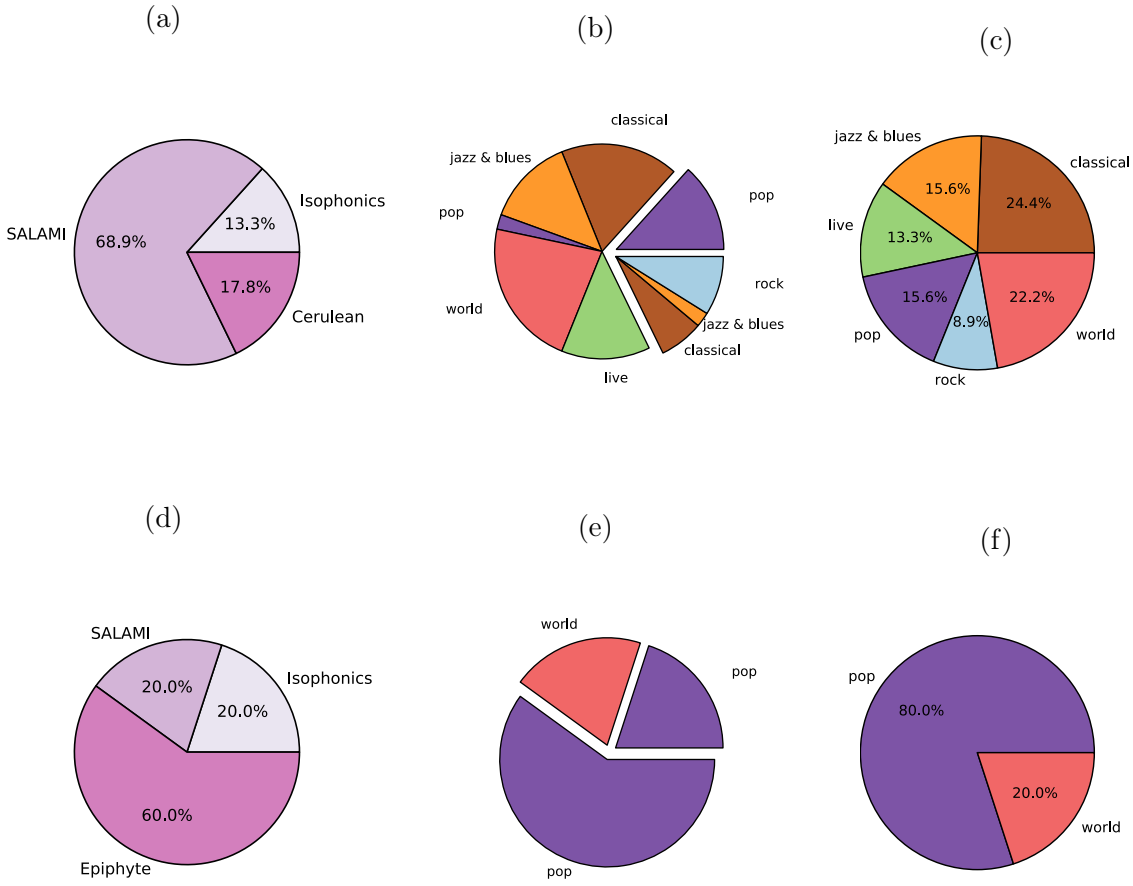


Figure 23: Top Row: Challenging tracks in the reduced dataset. Bottom Row: Control tracks in the reduced dataset. (a)/(d): Subset distribution. (b)/(e): Genre information for each subset. (c)/(f): Merged genre information.

form this task. The average number of years in musical training was 15.3 ± 4.9, and they all played an instrument for at least 10 years. These subjects annotated the 50 tracks following similar guidelines to those in (Smith et al., 2011) (the reader is welcome to peruse them online*). These guidelines ask the subjects to segment the audio using the Sonic Visualiser tool, and they also ask

*`https://files.nyu.edu/onc202/public/SegmentExperiment/`

the user to annotate the tracks at two different scales: large and small. The large scale, which is the most relevant in the context of music segmentation, comprises main substantial musical sections or long phrases, while the small scale includes subsections of the large scale that define riffs, short phrases or repeated motives. A segmentation example of the song "Somebody to Love" by Queen was provided and annotated by the author. Moreover, subjects were asked to report a maximum of two adjectives describing, for each track, why they found it difficult to segment (blank answers mean they did not struggle segmenting the track). As a result of this process, for each track five two-level annotations were collected to better analyze the results of the algorithms and to assess the degree of subjectivity of the subjects when annotating the reduced dataset. These results were parsed and included in JAMS files, one for each track of the reduced dataset. In the next chapter it will be discussed how employing these multiple annotations (either large scale, small scale, or both), can reduce the impact of subjectivity and consequently generate a more perceptually meaningful ground-truth.

5 Summary

In this chapter, the methodology used to collect multiple segmentation annotations of a challenging dataset was presented. To do so, an open-source framework called MSAF was introduced, which facilitates the task of running, analyzing, comparing, and evaluating music segmentation algorithms. This framework makes use of a novel format called JAMS that allows for multiple references and estimations to be stored in a single file, thus facilitating the comparison between numerous human annotations and algorithm outputs.

Then, the large segmentation dataset was described, including the distribution of its songs, the origin of their annotations, and the genres included. Finally, putting together MSAF with the large dataset, a reduced dataset was generated by automatically selecting the most challenging tracks (i.e., the ones for which the algorithms performed poorly when running MSAF) from the large dataset. This reduced dataset was annotated by five music experts in a process analogous to the one followed in the SALAMI dataset. The information collected, parsed into JAMS files, will help us to better understand the perception of music segmentation and its evaluation, which we further analyze and discuss in the following chapter.

CHAPTER VI

PERCEPTUAL EVALUATION OF MUSIC SEGMENTATION

1 Introduction

Music, which is sometimes defined as *organized sound* (Goldman, 1961), produces an auditory experience that is certainly a subjective one, regardless of how much "organization" there is in the sound (Wiggins, 2009). Nonetheless, as it has already been discussed in this dissertation, it is common in MIR to use datasets of tracks with a single human reference annotation in order to compare algorithms' outputs and assess the quality of these results. Even though considerable effort has been put into large scale automatic approaches (especially after the publication of the Million Song Dataset (Bertin-Mahieux et al., 2011)), little work has been done towards the use of multiple (or *larger* scale) human data to evaluate these algorithms following a more perceptual methodology. It could be argued that some of these MIR tasks might need additional annotators in order to contemplate the variations in perception originated by the sometimes intended ambiguity of the music audio signal.

The role of subjectivity (i.e., differences in perception), has been discussed under the task of chord recognition (Ni et al., 2013). It has been shown that the subjectivity effect for this task is important, and in the afore mentioned article they present a *crowd learning* method in order to merge multiple annotations to partially overcome this effect. Similarly, these type of

problems have also been explored for the beat tracking task (Grosche, 2010). In this case, they propose a framework in which multiple performances of an expressive tempo track are analyzed and merged, resulting in more robust beat trackers for pieces that have numerous recordings.

As for the music segmentation task, discovering the underlying structure of music can be daunting even for expert musicologists. Humans do not tend to agree on the perception of musical segment boundaries (Bruderer et al., 2009; Serrà et al., 2014). Therefore, and similarly to the task of chord recognition, relying on annotations that are produced by just one person might yield inaccurate results when evaluating segmentation algorithms. In the previous chapter, a methodology to collect multiple music segmentation annotations per track for a challenging segmentation dataset was discussed, including the introduction of a framework that contains numerous music segmentation algorithms. Having access to these new reference annotations and the estimations of these algorithms, an analysis of the variance as a method to measure the degree of disagreement will now be presented (which could be interpreted as *subjectivity*) in the task of music segmentation, therefore challenging the notion of having a *ground-truth* for this task with only one annotation per track. More specifically, it will be shown how, depending on the reference used, the ranking order of the algorithms varies based on their performance. Furthermore, four novel methods to merge down multiple human segmentation references per track (yielding either weighted flat or hierarchical references) to obtain more perceptually robust evaluations will be proposed. Two new metrics to evaluate the weighted flat and hierarchical merged boundaries will also be presented. Additionally, and independently of the number of references per track, the F-measure of the standard Hit Rate boundary retrieval metric will be challenged

by conducting a series of case studies that reveal that, perceptually, precision seems to have a stronger impact than recall when both scores are sufficiently high. With these results in mind, a proposal to enhance the F-measure will also be discussed at the end of this chapter.

2 Analyzing Annotations Agreement

Given the additional reference annotations collected for the reduced dataset detailed in the previous chapter and the easy access to numerous segmentation algorithms through MSAF, an analysis of the degree of variation that originates when evaluating the estimated segments with the new references can be performed. Particularly, the main focus is to investigate the discrepancy between algorithms' scores depending on which of the five references is used to produce the scores, therefore providing a quantification of the degree of subjectivity for this task. As it has been extensively discussed in the literature (Bruderer et al., 2009; Bruderer, 2008; Bruderer et al., 2006a; Smith et al., 2011; Serrà et al., 2014), the process of identifying boundaries can be regarded as subjective, so high discrepancy is expected in our experiments. Consequently, and in order to simplify and control these studies as much as possible, the focus of this chapter is on the segment boundary problem exclusively, using the F-measure of the hit rate with a 3 seconds window ($\mathbf{F}_3$) to evaluate the results. As discussed in Chapters II and IV, this metric is the most established when evaluating this task.

Firstly, the five supposedly trivial tracks to segment contained in the control group of the reduced dataset are analyzed. As a hypothesis, and given the assumed simplicity of these five songs, the scores of their estimated seg-

ments will unlikely vary when using the different human references. In order to quantify this variation, a two-way ANOVA of the averaged $\mathbf{F}_3$ scores across the control tracks is performed using five different segment boundary retrieval algorithms contained in MSAF,* and the five new human references, plus the original ground-truth annotations as factors. This type of analysis quantifies the degree of variation of these two factors, which will inform whether this disparity in the data is statistically significant or not. In this control case, the ANOVA returns a non-significant main effect on the annotations ($F(5, 120) = 0.22, p = 0.95$), illustrating that all annotations (including the earliest ground-truth) yield similar results when evaluating these algorithms on these five control tracks. The main interaction effect is not significant either ($F(20, 120) = 0.13, p = 0.99$), which validates the results of the ANOVA. In Figure 24 the marginal means are plotted, where it becomes apparent that the scores are relatively flat across annotations, confirming the findings of the ANOVA. When observing the score rankings of the algorithms it becomes apparent that, in average, they are all consistent across annotations except for Annotator 4, where the SF algorithm outperforms the Checkerboard one (being the other way around for the rest of annotators). This is the only irregularity found for the five control tracks, and it remains to be seen how much or how little these findings would deviate when analyzing a larger number of tracks.

The analysis of the annotations follows by investigating the 45 challenging tracks contained in the reduced dataset. In Figure 25 the marginal means for each of the algorithms against these challenging tracks are plotted, and in

*These algorithms are: CC, Checkerboard, OLDA, SF, SI-PLCA, since these were all the available algorithms in MSAF by the time the additional references were collected.

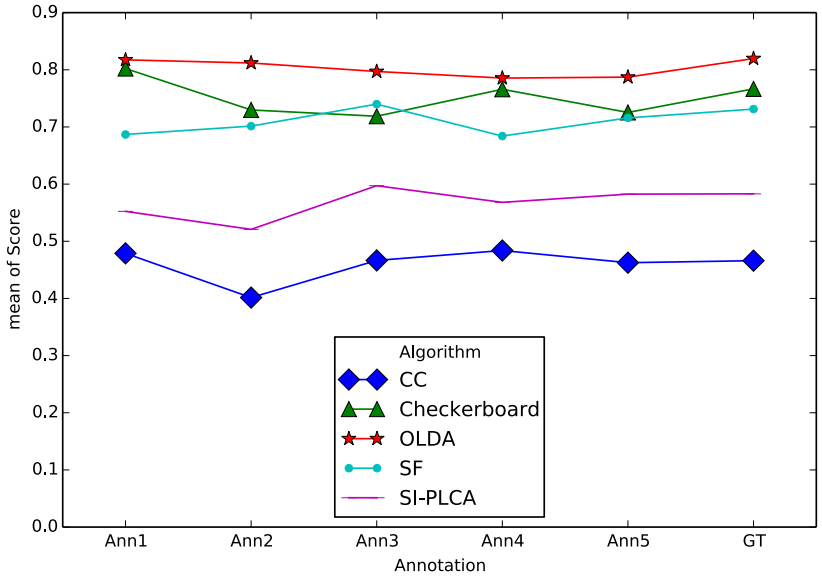


Figure 24: Marginal means of the scores of the algorithms when run on the 5 control tracks of the reduced data set against the multiple annotations.

this case it is visible that the scores vary depending on the annotation used. Note that it is not clear what the best performing algorithm is, since, depending on the annotator, this could be either Checkerboard, SF, or OLDA. A two-way ANOVA of the averaged $\mathbf{F}_3$ with algorithms and annotations as factors as before yields, in this scenario, a significant main effect on the annotation factor ($F(5, 1320) = 6.93, p < 0.01$), without a significant interaction between factors ($F(20, 1320) = 1.13, p = 0.3$). This indicates that the algorithm evaluation score significantly varies depending on the annotation chosen, making a case on how different humans may perceive the boundaries of a given piece, at least when the piece is a challenging one. Given that all of these annotations are valid, and all of them could be considered ground-truth, it is clear that they do not agree on the best/worst performing algorithm, which challenges the notion of a ground-truth produced by a single human. Moreover, reaching a 100% score might not be possible with single references, unless strong *overfit-*

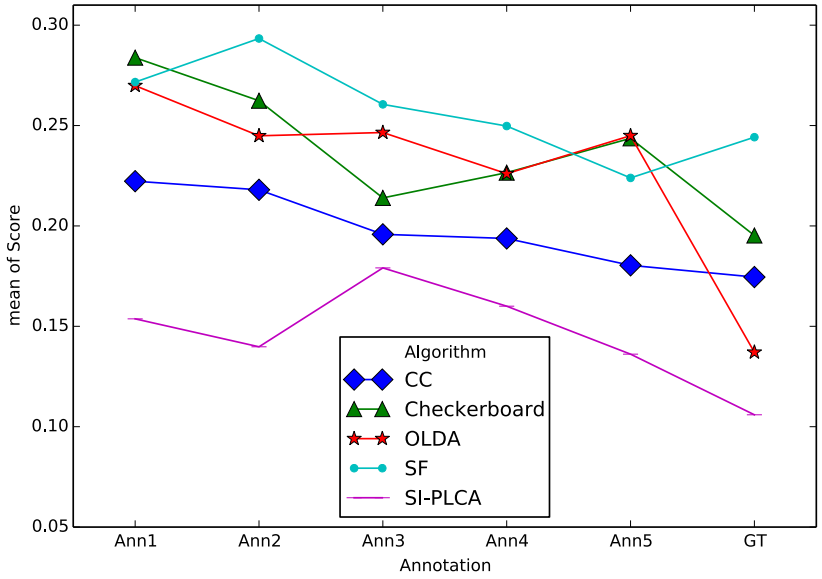


Figure 25: Marginal means of the scores of the algorithms when run on the 45 hard tracks of the reduced data set against the multiple annotations.

ting occurs on the subjectivity of the annotator, which is not desirable either. This motivates an alternative method of evaluating boundaries that can use information from multiple reference sources, thus capturing valid boundaries that might not have been annotated otherwise and reducing the importance of those boundaries that only one expert has identified.

3 Merging Annotations

Multiple annotations can be taken into account by merging them into a single one that includes all the relevant information. Various types of merging can occur, depending on the number of layers of the input/output annotations. In this section four different types of merging multiple boundaries are proposed, starting from the types that yield weighted flat annotations and finishing with the multi-layered (i.e., hierarchical) ones.

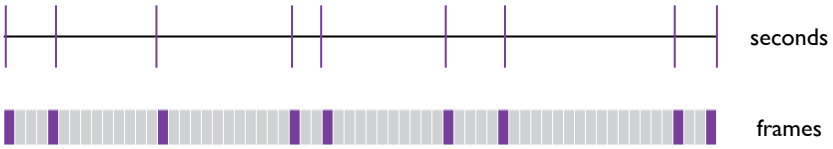


Figure 26: Converting the continuous time boundary annotations into discrete representations.

In order to accomplish this, the boundary times from the continuous time scale are discretized into a set of frames in order to simplify the merging problem, as illustrated in Figure 26. A set of time points are later transformed into a binary vector $\mathbf{x}$ of N frames, where each $x_i \in \mathbf{x}$ frame can only be either positive (boundary) or negative (no boundary). In all the different merging types, a frame rate of 10Hz is used.

3.1 Type I: Flat to Flat

The first merging type aims at aggregating the different large scale flat annotations into a single flat one that additionally contains a set of weights for each frame. This idea is similar to the one described in (Thom et al., 2002), where they also aggregate multiple annotated boundaries to assess the quality of various segmentation algorithms that work on the symbolic domain. To do so, the different annotations are summed in place in order to obtain a flat boundary annotation stored in a real-valued array, where each position represents the weight (or relevance) of the reference boundaries at a specific time frame. The weighted array is then normalized based on the number of annotations. Formally:

$$\mathbf{b_I} = \frac{1}{A} \sum_{i=1}^{A} \mathbf{b}_i^l \tag{31}$$

where A is the number of reference annotations, and $\mathbf{b}_i^l$ is the binary vector

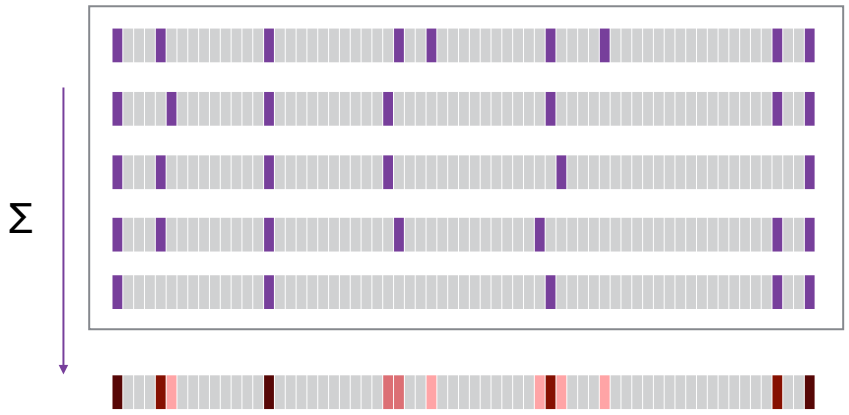


Figure 27: Visual representation of the process of merging flat annotations to a single flat annotation (type I).

representing the i-th large scale boundary annotation. Visually, an example of this process can be seen in Figure 27.

3.2 Type II: Hierarchical to Flat

In this case, the two levels of annotations collected for each track are interpreted as hierarchical. Note that the large annotation is simply a subset of the small scale one: all the boundaries in the large scale annotation will always be contained in the small scale one as well. Exploiting this, a two-level hierarchy is generated for each annotation in a track, and then collapsed into one single weighted flat annotation as depicted in Figure 28.

This process can be simplified by merging the large and small scale annotations of each reference into a weighted flat annotation by simply summing the two levels and normalizing them. Since, as aforementioned, the large scale is always included in the small scale, the large scale boundaries will obtain a higher weight in the aggregated flat annotation. Once these five weighted flat annotations have been computed, it is possible to merge them simply by taking

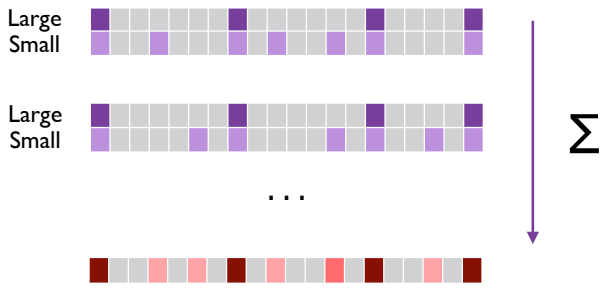


Figure 28: Visual representation of the process of merging hierarchical annotations to a single flat annotation (type II).

the average, following a similar process than in merging type I. Formally:

$$\mathbf{b_{II}} = \frac{1}{2A} \sum_{i=1}^{A} \left(\mathbf{b}_i^l + \mathbf{b}_i^s \right) \tag{32}$$

where $\mathbf{b}_i^s$ represents the i-th small scale boundary annotation.

3.3 Type III: Flat to Hierarchical

In this case, the large scale flat reference annotations is converted into a single hierarchical one. To do so, and taking the type I of merged boundaries ($\mathbf{b_I}$) as the starting point, a hierarchy is built using the weights to determine the layer where the boundaries belong. The set W is defined, which contains all the weights in $\mathbf{b_I}$ that appear at least once. The cardinality of this set $|W|$ is N_W which represents the number of hierarchical layers in the resulting aggregated annotation. This hierarchy can be interpreted as a matrix $\mathbf{B_{III}} \in \mathbb{R}^{N_W \times N}$ containing one binary vector per row, each row representing one layer in the hierarchy. The rows are sorted based on how high in the hierarchy this level is located (based on the weights in W). Formally, the (i, j) position of the

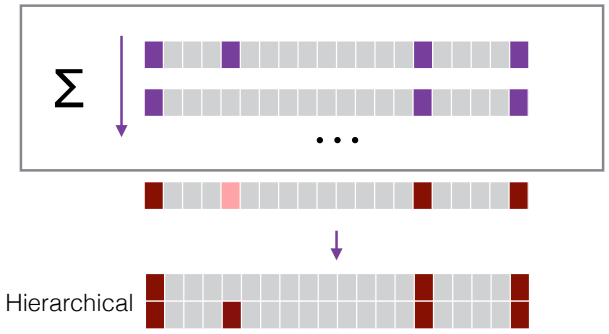


Figure 29: Example of the process of merging flat annotations to a single hierarchical one (type III).

hierarchical matrix can be defined as:

$$\mathbf{B_{III}}_{i,j} = \mathbb{1}(W_i = \mathbf{b_I}_j) \tag{33}$$

where $i \in [1, N_W]$, $j \in [1, N]$, and $\mathbb{1}$ is the indicator function, which returns 1 when its input evaluates to true and 0 otherwise. In Figure 29 an example is depicted.

3.4 Type IV: Hierarchical to Hierarchical

The final merging type aggregates multiple hierarchical annotations into a single hierarchical one. To do so, the merging types II (hierarchical to flat or $\mathbf{b_{II}}$) and III (flat to hierarchical or $\mathbf{B_{III}}$) can be trivially combined to produce the desired merged boundaries. More specifically, $\mathbf{b_{II}}$ is used as the input to the creation of hierarchical boundaries described in Equation 33 of type III. Formally:

$$\mathbf{B_{IV}}_{i,j} = \mathbb{1}(W_i = \mathbf{b_{II}}_j) \tag{34}$$

Note that in this merging type, and as in type II, both the large and

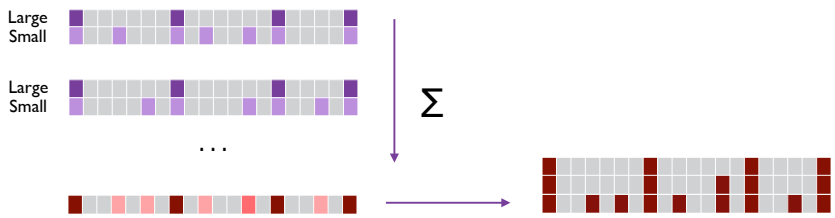


Figure 30: Example of the process of merging hierarchical annotations to a single hierarchical one (type IV).

small scale annotations are used to generate the aggregated one. The reader is referred to Figure 30 for a visual example.

4 Evaluation of Merged Boundaries

In the previous section four types of merging techniques for annotations were presented, which essentially produce two different aggregated boundary references: weighted flat and hierarchical. These references are considerably different to the standard ones —which are simply a set of time points representing the boundaries—, therefore a novel evaluation technique is needed for each of these two new aggregated boundaries. These metrics are presented in the following subsections.

4.1 Weighted Flat Boundaries Evaluation

In order to evaluate these weighted boundaries the actual Hit Rate method described in Chapter II is modified and used to evaluate the segmentation algorithms introduced in Chapter IV. Assuming a reference (or ground truth) of N_R boundaries with associated weights $\mathbf{w}_R = \{w_1, \ldots, w_{N_R}\}$ (that can be trivially obtained from $\mathbf{b_I}$ or $\mathbf{b_{II}}$, simply by converting the frames into time points), and an estimation of N_E boundaries, the weighted hits H, composed

of N elements, can be computed as follows:

$$H = h_1 w_{k_1} + \cdots + h_N w_{k_N} \qquad \text{s.t.} w_i \in \mathbf{w}_R \tag{35}$$

where the indices k_j correspond to those boundaries in the reference for which a hit has been found. In this case, as in the standard evaluation, a hit is found when an estimated boundary is within 3 seconds from the closest reference one, even though this can be seen as a tunable parameter of this metric.

Once the weighted H is computed, the precision and the recall values can be obtained by normalizing using the sum of the weights of the estimation and reference, respectively:

$$\mathcal{P} = \frac{H}{\sum \mathbf{w}_E} \qquad \mathcal{R} = \frac{H}{\sum \mathbf{w}_R} \tag{36}$$

It is worth noting that it is not possible to compute precision P without having the weight estimates for false positives $\mathbf{w}_E$ (i.e., the implemented algorithms in MSAF, or any standard music segmentation algorithm in general, do not output weighted estimations), therefore the weights in $\mathbf{w}_R$ are used to approximate $\mathbf{w}_E$ as follows:

$$\sum \mathbf{w}_E = \sum_{j=1}^{N} w_{k_j} + M\mu_{\mathbf{w}_R} \tag{37}$$

where M is the number of estimated boundaries that are not considered hits. Finally, the weighted F-measure $\mathcal{F}$ between the weighted precision $\mathcal{P}$ and the weighted recall $\mathcal{R}$ is computed as usual (see Equation 11 from Chapter II).

4.2 Hierarchical Boundaries Evaluation

In this subsection a novel ranking-based metric to assess hierarchical boundaries is presented. With this metric, the hierarchical boundaries can be evaluated either against hierarchical or (non-weighted) flat estimations.

Working at a frame level, as when introducing the different merging types, let $\mathbf{b}$ denote a temporally contiguous partition of the time range spanning the track in question. The superscripts $\mathbf{b}^R$ and $\mathbf{b}^E$ are used to denote *reference* and *estimation*, respectively. $\mathbf{b}_i$ denotes the identifier of the partition containing the i-th sample. Following the same ideas as in the pairwise clustering evaluation (Levy and Sandler, 2008), it is required that the same collection of samples be applied to both reference and estimated annotations to obtain the same amount of frames in the estimation and the reference for the song to be evaluated.

In order to represent multi-layered boundaries (i.e., hierarchical annotations), and following the superscript conventions above, $\mathbf{B}$ is used. Note how some of the segments that represent smaller sections (or riffs or motives) will now be contained within larger segments. $\mathbf{B}_i$ indicates the most specific segment containing i (i.e., the segment containing i at the deepest level in the hierarchy). $\mathbf{B}_{i,j}$ denotes the least common ancestor of $\mathbf{B}_i$ and $\mathbf{B}_j$. In order to indicate precedence (or containment), $\prec$ is used: e.g., $\mathbf{B}_i \prec \mathbf{B}_{i,j}$.

As an example, in Figure 31 a representation of a hierarchical annotation of a toy track can be seen. In this case, $\mathbf{B}_2$ would represent the most specific segment in time frame 2: that is, the light yellow one at the bottom left corner of the figure. $\mathbf{B}_{2,8}$ would indicate the first common ancestor of the segments of time frames 2 and 8, which in this case is the first segment of the second

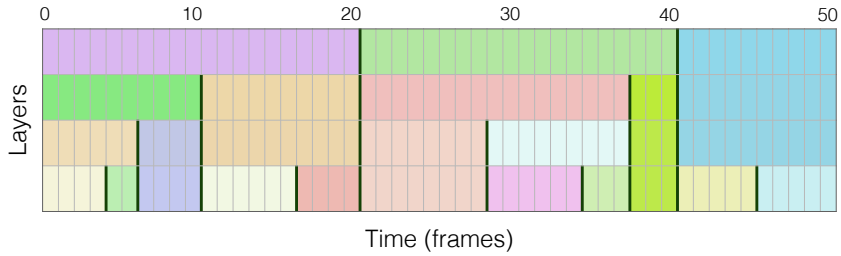


Figure 31: Toy example of the representation of a hierarchical annotation.

layer (green in the figure). Therefore, in this example, the following is true: $\mathbf{B}_2 \prec \mathbf{B}_{2,5} \prec \mathbf{B}_{2,8} \prec \mathbf{B}_{2,12}$.

As it becomes obvious, flat segmentations are a special case of hierarchical segmentations, where there is one node at the root of the hierarchy containing all samples. This is similar to the large and small scale annotations used in the merging types introduced earlier in this chapter.

To further illustrate this notation, the attention can be focused on a query sample q, such that $\mathbf{B}_{q,\cdot}$ induces a partial ranking over the remaining samples. The maximally relevant frames will be contained in $\mathbf{B}_q$, followed by those in $\mathbf{B}_q$'s immediate ancestor, and so on. In the example of Figure 31, the most relevant frames for the query $q = 2$ will be the frames with the same yellow bright color that are contiguous to frame 2 in the last layer. The second most relevant frames will be the frames 4 and 5 (light green), then the four following frames (light purple), and finally, the rest of the frames until frame index 20 (light yellow and red). The rest of the frames in the track will not be relevant to this query. This provides a connection between hierarchical time-series decompositions and ranking evaluation, which is key in order to fully understand this novel technique.

4.2.1 Evaluation Description

In order to describe this method, the case of non-weighted flat boundaries is first considered, which exposes the link between partial ranking and boundary evaluation. Let q denote an arbitrary sample, and let i and j denote any two samples such that $\mathbf{b}_q^R = \mathbf{b}_i^R$ and $\mathbf{b}_q^R \neq \mathbf{b}_j^R$. Consequently, i can be considered *relevant* for q, and j may be considered *irrelevant* for q. This leads to a straightforward reduction to bipartite ranking (i.e., either the current segment belongs to q or it does not).

Formally, and assuming that both $\mathbf{b}^E$ and $\mathbf{b}^R$ have N samples:

$$f(q; \mathbf{b}^E) = \sum_{\substack{i \in \mathbf{b}_q^R \\ j \notin \mathbf{b}_q^R}} \frac{\mathbb{1}(\mathbf{b}_q^E = \mathbf{b}_i^E \neq \mathbf{b}_j^E)}{|\mathbf{b}_q^R| \cdot (N - |\mathbf{b}_q^R|)} \tag{38}$$

which can be read as the score for sample q is the fraction of pairs (i, j) for which $\mathbf{b}^E$ agrees with $\mathbf{b}^R$ with respect to q (i.e., membership in the segment).

In order to get the mean sample recall metric, all samples q are averaged:

$$\rho(\mathbf{b}^E) := \frac{1}{N} \sum_q f(q; \mathbf{b}^E) \tag{39}$$

In order to extend the flat evaluation to a hierarchical one, Equation 38 can be expressed using strict hierarchical precedences instead of membership (in)equalities. Therefore, we can convey this as $\mathbf{B}_{q,i} \prec \mathbf{B}_{q,j}$, rather than compare i and j where $\mathbf{b}_q^E = \mathbf{b}_i^E \neq \mathbf{b}_j^E$. That is, the pair (q, i) merges before

(q, j):

$$g(q; \mathbf{B}^E) \coloneqq \frac{1}{Z_q} \sum_{\substack{i,j \\ \mathbf{B}^R_{q,i} \prec \mathbf{B}^R_{q,j}}} \mathbb{1}(\mathbf{B}^E_{q,i} \prec \mathbf{B}^E_{q,j}) \tag{40}$$

where Z_q is a normalization term that counts the number of elements in the summation.

Similarly to f in Equation 38, and using the bipartite ranking analogy, g can be viewed as a classification accuracy of correctly predicting pairs (i, j) as positive (q and i merge first) or negative (q and j merge first). The case when $\mathbf{B}_{q,i} = \mathbf{B}_{q,j}$ is prevented by the strict precedence operator in the summation.

Equation 40 can be alternatively be viewed as a generalized area under the curve (AUC) over the partial ranking induced by the hierarchical segmentation, where depth within the estimated hierarchy $\mathbf{B}^E$ plays the role of the relevance threshold. An exhaustive review on partial ranking methods is published in (Fagin et al., 2006).

Since $g(q; \mathbf{B}^E)$ acts as a generalization of recall, an estimate $\mathbf{B}^E$ will achieve a low score if it fails to distinguish between pairs i and j. Similarly to the normalized conditional entropy metrics reviewed in Chapter II, this can be interpreted as a form of *under-segmentation*, where a low score indicates either an ordering error or a lack of specificity (Lukashevich, 2008).

By aggregating over all frames q of the track, the hierarchical *under-segmentation* metric is obtained:

$$\mathcal{H}_u(\mathbf{B}^E) \coloneqq \frac{1}{N} \sum_q g(q; \mathbf{B}^E). \tag{41}$$

The over-segmentation metric $\mathcal{H}_o(\mathbf{B}^E)$ is defined analogously by swapping the roles of $\mathbf{B}^E$ and $\mathbf{B}^R$ in Equations 40 and 41. As in the normalized condi-

tional entropy scores, the two metrics can be combined to summarize structural agreement between estimated and reference hierarchical annotations by using the F-measure between $\mathcal{H}_u$ and $\mathcal{H}_o$ to obtain $\mathcal{H}_f$.

4.2.2 Windowing in Time

The hierarchical metrics described above are able to apprehend the notion of multi-layered, frame-level relevance, but two technical limitations arise from their definition. One of them is the dependence of the length of the track N when computing the scores, which is problematic when comparing the scores of two tracks of different durations. Note that when N is large enough, Equation 40 may be dominated by trivially irrelevant comparison points j which lie far from q in time, i.e., $|q - i| \ll |q - j|$. Nevertheless, tracks with small N (i.e., short duration) have fewer such trivial comparisons. Consequently, tracks that have a long duration might obtain higher scores when compared to short duration tracks, only because of the number of trivial comparisons that arise in the long tracks. Ideally, one would want to avoid these dynamic range discrepancies. The second issue is related to the expensive computation time that these metrics require ($\mathcal{O}(N^3)$ using a direct implementation of Equation 41).

To address these problems, a parameter is added to our system: a time window of w seconds that normalizes the dynamic range of the metric and simplifies its calculation. Using w, the number of samples under consideration is constrained to $M = \lceil w \cdot f_r \rceil$. Adding this windowing property to our

hierarchical metric equations yields the following windowed version:

$$g(q, M; \mathbf{B}^E) := \frac{1}{Z_q(M)} \sum_{\substack{M_s \leq i,j \leq M_e \\ \mathbf{B}^R_{q,i} \prec \mathbf{B}^R_{q,j}}} \mathbb{1}(\mathbf{B}^E_{q,i} \prec \mathbf{B}^E_{q,j}), \tag{42}$$

$$\mathcal{H}_u(\mathbf{B}^E, M) := \frac{1}{N} \sum_q g(q, M; \mathbf{B}^E), \tag{43}$$

where $M_s = \min\{0, q - M/2\}$ and $M_e = \max\{q + M/2, N\}$ are the start and ending, respectively, of the current window in frame indices. Thanks to this addition, the computational complexity of the metric is reduced to $\mathcal{O}(NM^2)$. Furthermore, the track duration is no longer a factor that might impact the final scores, since each query frame q now operates over a fixed number of comparisons (i, j), independently of N.

4.2.3 Choosing a time window

Given the significant impact of the size of the window, it is important to choose a value that generalizes well across multiple music genres. Local changes will be successfully captured by small windows (e.g., $w \leq 3$ seconds), since large-scale structural changes usually occur at greater time intervals (Smith and Chew, 2013). Ideally, the window should be long enough to capture boundaries of segments at multiple resolutions, but not so large as to become dominated by trivial comparisons. Inspired by the Hit Rate measure described in Chapter II, which is recommended to be computed with two different time windows (0.5 and 3 seconds, since there may not be a single window length that is optimum for all tracks), these metrics are suggested to be reported for two time windows: $w = 3$ and $w = 30$. These values empirically seem to capture the variations that might arise due to the differences between segment durations across songs.

4.2.4 Transitivity

A final potential problem of these hierarchical metrics originates when deep hierarchies are present in the annotations. In order to illustrate this, consider the sequence $\mathbf{B}_{q,i} \prec \mathbf{B}_{q,j} \prec \mathbf{B}_{q,k}$. Due to the transitive containment structure of $\mathbf{B}$, the following is true: $i \in \mathbf{B}_{q,i} \subseteq \mathbf{B}_{q,j}$. The pair (i, k) will appear twice in the summation of Equation 40, since this summation ranges over all precedence comparisons

The summation can be restricted to only range over direct precedence relations in order to address this issue. In practice, only frames from successive levels in the hierarchy will be compared, therefore removing redundant comparisons and increasing the range of g. In the following section this metric will be used with $w = 30$ in order to evaluate the hierarchical merged evaluations.

5 Robustness of Merged Boundaries

In Section 2 of this chapter it was discussed how unreliable it is to depend on only one annotation in order to assess the quality of various algorithms when analyzing challenging tracks. Now it will be shown that once the various annotations have been merged using the techniques described in Section 3, and having evaluated them using the metrics described above, the scores of these evaluations may become more trustful, reaching confidence levels similar to the ones obtained when assessing the control tracks against a single annotator.

To do so, we can not simply merge the five annotations and evaluate the four different types of merging, since there are no references to compare them to. Therefore, and inspired by the cross-validation model standard in machine learning assessment (Kohavi, 1995), the five annotations are combined into sets

of three, which results into 10 different sets ($\binom{5}{3} = 10$) of three annotations to be compared against each other. For each merging type, a two-way ANOVA on the average evaluation score is computed as before (using the weighted F-measure $\mathcal{F}$ for the weighted flat boundaries and the hierarchical score $\mathcal{H}_f$ for the hierarchical boundaries) with algorithm and set as factors, and the main effect of the set is explored. If it is statistically significant, the scores of the evaluation of the algorithms would depend on the set used, therefore making this type of merging less reliable when assessing algorithms, similarly to what happens when using just one annotator, as we saw in Section 2. On the other hand, if this main effect is not significant, it might be the case that this type of merging makes the evaluation more stable (at least with the current annotators), since having three annotators per track makes the scores as dependent on the set of annotations used. This behavior would be similar to the one observed when using the control tracks.

In Table 12 the results of the two-way ANOVAs for each merging type are shown. All types seem to yield statistically similar results independently of the set used except for type III. This indicates that, on average, significantly similar scores across all of the sets will be obtained when merging the boundaries using types I, II and IV. However, when using the merging type III, significantly different scores across the different sets are procured, which could result in a potential problem when comparing the quality of various algorithms.

To better contrast these findings, the marginal means for each type are plotted in Figure 32. Even though the ANOVA showed that the merging type III may not be as stable as the other types (the averages differ across sets, especially in sets 1 and 6), the plots describe how in types III and IV the same

Merge Type	$F(9, 2200)$	p-value
I	.23	.99
II	.42	.92
III	3.35	$< .01$
IV	1.56	.12

Table 11: Assessing the quality of the types of merging multiple annotations using three-annotation sets

ranking of the averages of the algorithms is obtained. On the other hand, types I and II yield, in average, the same scores across sets in a significant way, but the ranking slightly differs in the higher scores. This could be problematic, but still their relatively flat scores are similar to those obtained when using a single annotation to assess the quality of the algorithms, therefore making them qualitatively as reliable.

When comparing sets of three annotators extracted from a total of five there is always at least one annotator that is common in each comparison, which will likely and undesirably reduce the differences between sets. A way to overcome this overlapping without requiring additional annotators is to organize the annotators into sets of two. Even though this is not ideal, since merging two annotators might not be sufficient, it might shed some light in the consistency across annotators. The same analysis is repeated with the 10 sets of two annotators ($\binom{5}{2} = 10$), and in this case only the main effect on the set is not significant for types I and II (see Table 12 for the two-way ANOVA results).

However, when visualizing the marginal means across sets (Figure 33), a similar behavior occurs: the rankings of the algorithm average scores remains

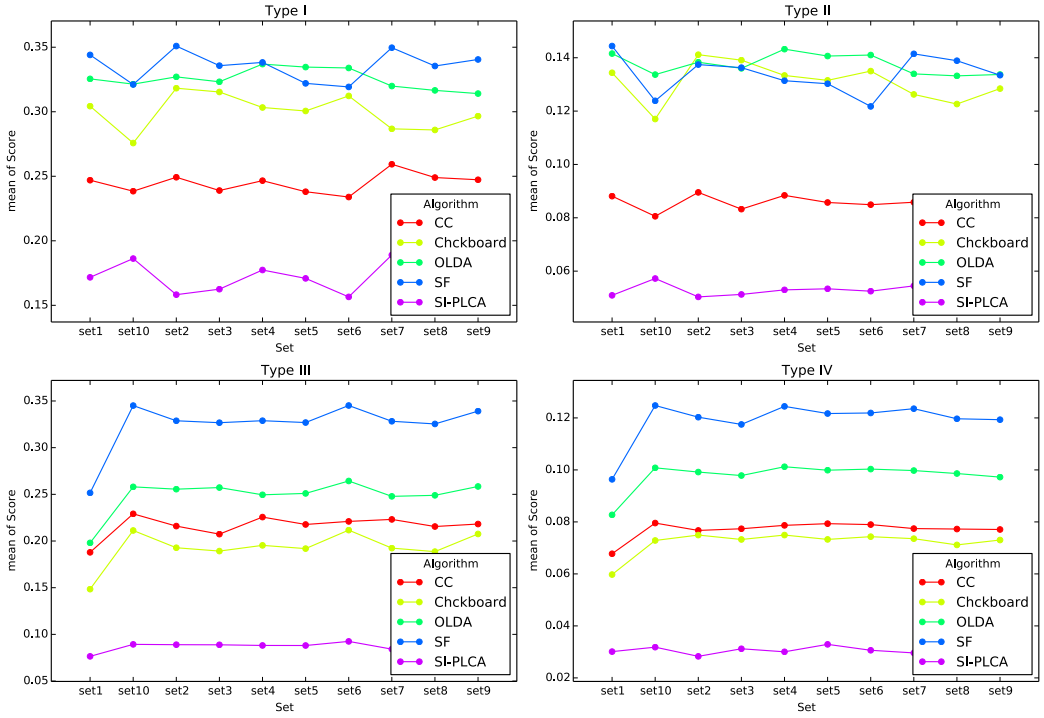


Figure 32: Comparison of the marginal means of the four types of merging boundaries using sets of three annotations each.

the same across sets in types III and IV, while it slightly varies in the higher scores of types I and II. These results suggest that using two annotators per track might be sufficient in order to better assess automatic methods.

Note that, ideally, to assess the quality of the merging of the five annotations, we would need additional sets of 5 annotations performed by different humans. Therefore, it has been shown that at least two annotations per track seem to suffice in order to evaluate the quality of various algorithms when running on challenging tracks, even though using three seems to yield more stable results.

Merge Type	$F(9, 2200)$	p-value
I	.68	.71
II	.97	.46
III	12.71	$< .01$
IV	7.35	$< .01$

Table 12: Assessing the quality of the types of merging multiple annotations using sets of pairs of annotations

6 Reconsidering the F-measure of the Hit Rate Metric

So far, in this Chapter the importance of having additional annotations in order to reduce the effect of subjectivity when evaluating segment boundaries has been reviewed. Besides having additional annotations for challenging tracks, one might want to analyze the behavior of the current techniques to assess this task. In this section the standard metric to evaluate boundaries, the F-measure of the Hit Rate at three seconds $\mathbf{F}_3$, is challenged by raising awareness about its limitations when perceptually comparing algorithms with the same F-measure but different precision $\mathbf{P}_3$ and recall $\mathbf{R}_3$ values. Here the results of multiple experiments are presented and discussed, where subjects listened to different musical excerpts containing boundary indications and had to rate the quality of the boundaries. These ratings usually favored the sets of boundaries that had a higher precision. Based on these results, an alternative evaluation based on the F-measure that emphasizes precision over recall is presented, aiming at making the section boundary evaluation more expressive and reliable.

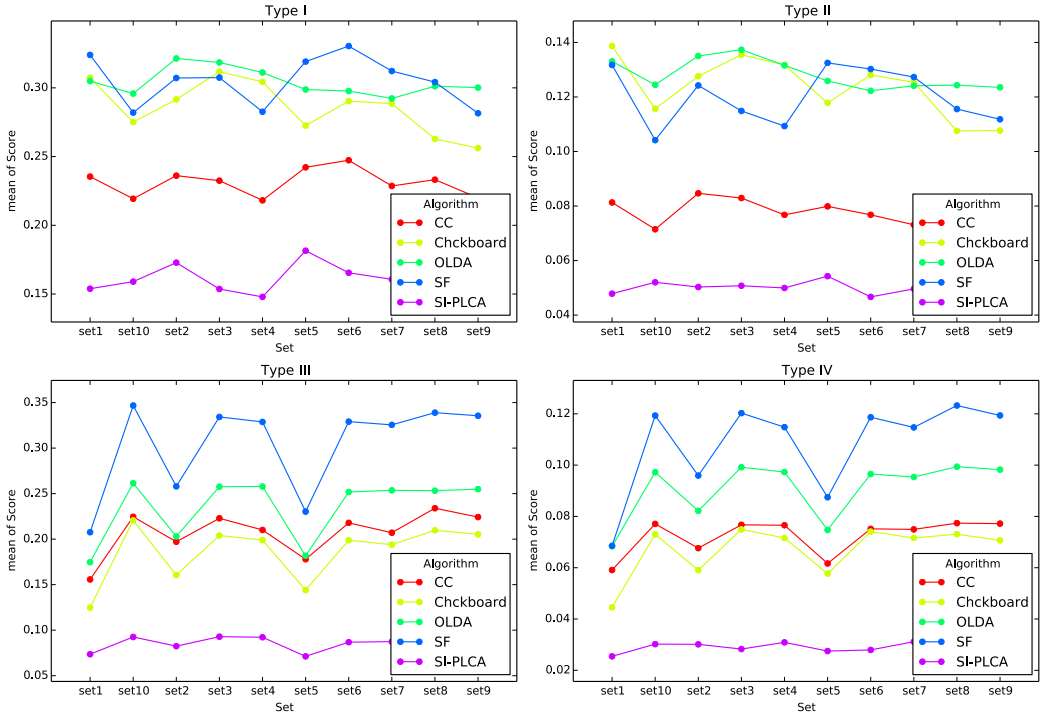


Figure 33: Comparison of the marginal means of the four types of merging boundaries using sets of two annotations each.

6.1 Preliminary Study

An initial experiment was carried out in order to assess the quality of three algorithms, which are named $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. $\mathcal{A}$ is an unpublished algorithm that relies on homogeneous repeated section blocks; $\mathcal{B}$ is an existing algorithm that uses novelty in audio features to identify boundaries; and $\mathcal{C}$ combines the previous two methods. Following standard procedures, these three techniques were optimized to maximize $\mathbf{F}_3$ on the structure-annotated Levy dataset (Levy and Sandler, 2008), composed of 60 tracks of western popular music. Table 13 shows each method's average F-measure, precision, and recall values across the entire set. One would expect that $\mathcal{C}$, which maximizes the F-measure, would be the algorithm that yields the best results from a perceptual perspective.

Preliminary Study			
Algorithm	$\mathbf{F}_3$	$\mathbf{P}_3$	$\mathbf{R}_3$
$\mathcal{A}$	49	57	47
$\mathcal{B}$	44	46	46
$\mathcal{C}$	51	47	64

Table 13: Algorithms and their ratings used to generate the input for the preliminary study. These ratings are averaged across the 60 songs of the Levy dataset.

Two college music majors from Berklee College of Music were asked to rank the three algorithms' results for all the 60 songs of western popular music from the Levy catalog. The main objective was to compare the algorithms with each other and determine the best one from a perceptual point of view. More specifically, the participants were told to listen to each of the algorithm outputs for all the songs and rank the algorithms by the quality of their estimated section boundaries; no particular constraints were given on what to look for. As when collecting additional boundaries in Chapter V, Sonic Visualiser (Cannam et al., 2006) was used to display the waveform and three section panels for each of the algorithms in parallel (see Figure 34). While playing the audio, listeners could both see the sections and hear the boundaries indicated by a distinctive percussive sound. The algorithms were randomly organized for each song, such that listeners had no way of knowing the algorithms to be chosen.

Analysis of the results showed that 68.3% of the time, the two participants chose the same best algorithm. In 23.3% of the cases, they disagreed on the best, and in just 8.3% of the cases, they chose opposite rankings. When they actually agreed on the best algorithm, they chose $\mathcal{A}$ 58.5% of the time. This algorithm did not have the highest F-measure but the highest precision.

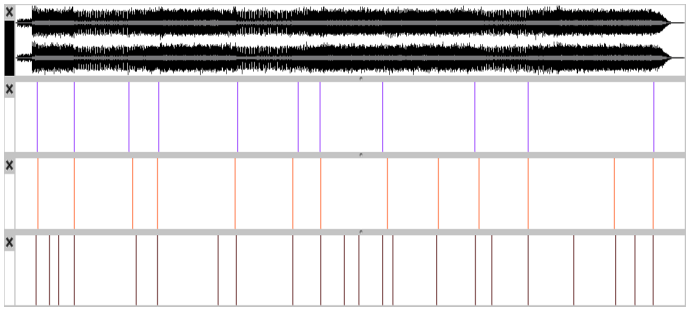


Figure 34: Screenshot of Sonic Visualiser used in the preliminary experiment. The song is "Smells Like Teen Spirit" by Nirvana. In this case, algorithms are ordered as $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ from top to bottom.

Perhaps more surprising, they chose $\mathcal{C}$ only 14.6% of the time even though that algorithm had the highest F-measure and the highest recall.

These preliminary results raised the following questions: Is the F-measure informative enough to evaluate the accuracy of automatically estimated boundaries in a perceptually-meaningful way? Is precision more important than recall when assessing music boundaries? Are the two subjects who took this preliminary study representative of the general population? If we had enough data, could similar results inform more perceptually meaningful metrics?

To address these questions two additional formal experiments were run in order to better understand this apparent problem and aim at identifying a possible solution.

6.2 Experiment 1: Rating Boundaries

To further explore the hypothesis that arose in the preliminary study about precision being more perceptually relevant than recall, these two values should be carefully manipulated in a controlled environment. For this new experiment, a set of boundaries was synthesized by setting specific values for preci-

sion and recall while maintaining a near-constant F-measure. Furthermore, the subjects size is increased in order to obtain more robust findings. With these considerations in mind, the experiment was designed to be both shorter in time and available on line as a web survey in order to facilitate participation*.

6.2.1 Methodology

Five track excerpts were automatically selected from the Levy catalog by finding the one-minute segments containing the highest number of boundaries across the 60 songs of the dataset, in order the keep subjects' attention as high as possible. By having short excerpts instead of full songs, the duration of the entire experiment could be reduced with negligible effect on the results —past studies have shown that boundaries are usually perceived locally instead of globally (Tillmann and Bigand, 2001). Three different segmentations were synthesized for each excerpt: ground truth boundaries (GT) with $\mathbf{F}_3 = 100\%$; high precision (HP) boundaries with $\mathbf{P}_3 = 100\%$ and $\mathbf{R}_3 \approx 65\%$; and high recall (HR) boundaries with $\mathbf{R}_3 = 100\%$ and $\mathbf{P}_3 \approx 65\%$. The extra boundaries for the HR version were randomly distributed (using a normal distribution) across a 3 seconds window between the largest regions between boundaries. For the HP version, the boundaries that were most closely spaced were removed. Table 14 presents the Hit Rate values $\mathbf{F}_3$, $\mathbf{P}_3$ and $\mathbf{R}_3$ for the five tracks along with the average values across excerpts. Note how similar the $\mathbf{F}_3$ values are for HP and HR.

Subjects had to rate the "quality" of the boundaries for each version of the five tracks by choosing a discrete value between 1 and 5 (lowest and

* `http://urinieto.com/NYU/BoundaryExperiment/`

Experiment 1 Excerpt List						
Song Name (Artist)	HP			HR		
	$\mathbf{F}_3$	$\mathbf{P}_3$	$\mathbf{R}_3$	$\mathbf{F}_3$	$\mathbf{P}_3$	$\mathbf{R}_3$
Black & White (Michael Jackson)	80.9	100	68.0	79.4	65.8	100
Drive (R.E.M.)	78.5	100	64.7	79.1	65.4	100
Intergalactic (Beastie Boys)	76.4	100	61.9	79.2	65.6	100
Suds And Soda (Deus)	78.2	100	65.3	80.0	66.6	100
Tubthumping (Chumbawamba)	74.4	100	59.3	79.4	65.9	100
Average	77.7	100	63.6	79.4	65.9	100

Table 14: Excerpt list with their evaluations for experiment 1. $\mathbf{F}_3$ of GT is 100% (not shown on the table).

highest ratings, respectively). Although this might arguably bias the subjects towards the existing boundaries only (reducing the influence of the missing ones), it is unclear how to design a similar experiment that would avoid this. Excerpts were presented in random order. The results could only be submitted once the participants had listened to all of the excerpts. As in the preliminary experiment, auditory cues for the section boundaries were added to the original audio signal in the form of a salient sharp sound. For this experiment, no visual feedback was provided because the excerpts were short enough for listeners to retain a general perception of the accuracy of the boundaries. The entire experiment lasted around 15 minutes (5 excerpts $\times$ 3 versions $\times$ one minute per excerpt).

An announcement to various specialized mailing lists was sent in order to recruit participants. As such, most subjects had a professional interest in

music, and some were even familiar with the topic of musical structure analysis. A total number of 48 participants took part in the experiment; subjects had an average of 3.1 ± 1.6 years of musical training and 3.7 ± 3.3 years of experience playing an instrument.

6.2.2 Results and Discussion

The results reported the following perceived accuracy order for the three different versions: ground truth version (GT) first, then high precision version (HP), and finally high recall version (HR). The averages across versions for all the results with their standard deviations can be seen in Figure 35.

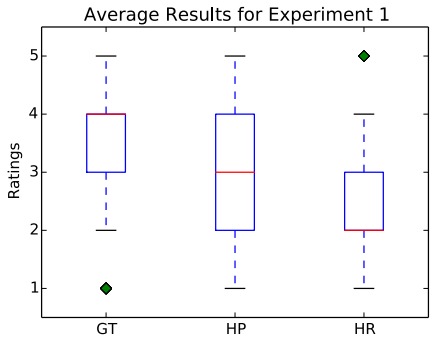


Figure 35: Average ratings across excerpts for Experiment 1; GT = ground truth; HP = high precition; HR = high recall.

A two-way, repeated-measures ANOVA was performed on the accuracy ratings with type (GT, HP, HR) and excerpt (the five songs) as factors. There were 48 data points in each Type × Excerpt category. The main effects of type, $F(2,\ 94) = 90.74$, $MSE = 1.10$, $p < .001$, and excerpt, $F(4,\ 188) = 59.84$, $MSE = 0.88$, $p < .001$, were significant. Additionally, an interaction effect was significant, $F(6.17,\ 290.01) = 9.42$, $MSE = 0.74$, $p < .001$ (Greenhouse-Geisser corrected), indicating that rating profiles differed based on excerpt. Mean ratings by type and excerpt are shown in Figure 36.

It is clear from these results that there is a pattern showing that subjects preferred segmentations with high precision over high recall (Figure 36). Post-hoc multiple comparisons indicated that differences between means of all three types were significant. The only excerpt where precision was not rated more highly than recall was in Excerpt 5 (Tubthumping), a difference that contributed primarily to the interaction. In this case, the excerpt contains a distinctive chorus where the lyrics "I get knocked down" keep repeating. This feature is likely the reason some subjects were led to interpret every instance of this refrain as a possible section beginning even though the harmony underneath follows a longer sectional pattern that is annotated in the ground truth. On the other hand, Excerpt 3 (Intergalactic) obtained similar ratings for GT and HP, likely due to the high number of different sections and silences it contains. This can become problematic when extra boundaries are added (therefore obtaining poor ratings for the high-recall version). Nevertheless, given the subjectivity of this task (Bruderer et al., 2009) and the multi-layer organization of boundaries (Peeters and Deruty, 2009), it is not surprising that this type of variability appears in the results, especially when only one annotation per track exists on this ground-truth.

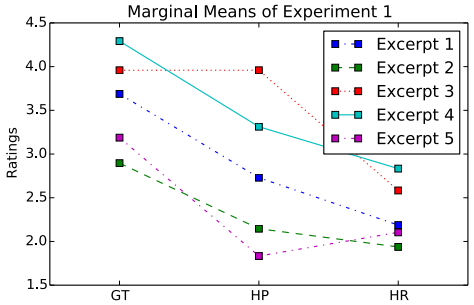


Figure 36: Means for excerpt and version of the results of Experiment 1.

The results of this experiment indicate that, for these tracks and the

given ground-truth, precision is more perceptually relevant than recall for the evaluation of boundaries, validating the preliminary findings in a controlled scenario and with a much larger population of subjects. However, the number of tracks employed in this experiment was limited. As a follow-up, these findings were explored using a larger dataset in Experiment 2.

6.3 Experiment 2: Selecting Boundaries

In the previous subsection the relative importance of precision over recall for a reduced dataset of five tracks was depicted. Regardless, it remains to be seen whether the F-measure, precision, and recall can predict a listener's preference when faced with a real-world evaluation scenario (i.e., boundaries that algorithms estimated, not synthesized). In order to address this, in Experiment 2 the following was used: excerpts sampled from a larger set of music, boundaries computed with state-of-the-art algorithms, and evaluation limited to pairwise preferences. Moreover, instead of rating each excerpt version, now subjects had to select the one that they found more appropriate (i.e., had the highest quality boundaries).

6.3.1 Methodology

The algorithms used to compute the boundaries, which are available in MSAF (introduced in Chapter V), are the following: structural features (SF, (Serrà et al., 2014)), convex non-negative matrix factorization (C-NMF, (Nieto and Farbood, 2013b)), and shift-invariant probabilistic latent component analysis (SI-PLCA, (Weiss and Bello, 2011)). These three algorithms yield ideal results for our experimental design since SF outputs some of the best results reported so far on boundaries recognition (high precision and high recall). C-NMF

tends to over segment (higher recall than precision), and SI-PLCA, depending on parameter choices, tends to under segment (higher precision than recall).

These three algorithms were run on a relatively large database of 463 songs composed of the conjunction of the Beatles dataset, the Levy catalog (Levy and Sandler, 2008), and the freely available songs of the SALAMI dataset (Smith et al., 2011). Once computed, three criteria were applied to filter the results:

1) At least two algorithms' outputs from the three estimated have a similar F-measure (within a 5% threshold).

2) The F-measure of both algorithms must be at least 45%.

3) At least a 10% difference between the precision and recall values of the two selected algorithm outputs exists.

41 out of the 463 tracks met the above criteria. Furthermore, a qualitative selection was also applied to these filtered tracks (there are many free tracks in the SALAMI dataset that are live recordings with poor audio quality or simply speech), resulting in a final set of 20 songs. Even if the number of these carefully selected tracks is relatively low, it is expected to be representative enough to address our research questions. Given the two algorithmic outputs, two differently segmented versions were created for each track based on their Hit Rate values: high precision (HP) and high recall (HR). Moreover, similar to Experiment 1, only one minute of audio from each track was utilized, starting 15 seconds into the song.

The average metrics across the 20 selected tracks are shown in Table 15. As it can be seen, $\mathbf{F}_3$ values remain the same, while precision and recall vary.

Boundaries Version	$\mathbf{F}_3$	$\mathbf{P}_3$	$\mathbf{R}_3$
HP	65	82	56
HR	65	54	83

Table 15: Average F-measure, precision, and recall values for the two versions of excerpts used in Experiment 2.

In order to facilitate participation, and similarly to Experiment 1, the interface for Experiment 2 was on line*. Each participant was presented with five random excerpts selected from the set of 20. Instead of assessing the accuracy on a scale like in Experiment 1, listeners had to choose the version they found more accurate. In order to uniformly distribute excerpts across total trials, selection of excerpts was constrained by giving more priority to those excerpts with fewer collected responses. An average of 5.75 results per excerpt was obtained. The two versions were presented in random order, and subjects had to listen to the audio at least once before submitting the results. As in previous experiments, boundaries were marked with a salient sound.

A total of 23 subjects, recruited from professional mailing lists as in the previous study, participated in the experiment. Participants had an average of 2.8 ± 1.4 years of musical training and 3.2 ± 2.9 years of experience playing an instrument.

6.3.2 Results and Discussion

In this case, 67% of the times subjects chose the HP version, while HR was only chosen 33% of the times. Even though this clearly shows a preference for the HP version, these results were further analyzed by performing a binary logistic

*`http://cognition.smusic.nyu.edu/boundaryExperiment2/`

regression test (Peng et al., 2002) with the goal of understanding what specific values of the F-measure were actually useful in predicting subject preference (the binary values representing the versions picked by the listeners). Logistic regression enables us to compute the following probability:

$$P(Y|X_1, \cdots, X_n) = \frac{e^{k+\beta_1 X_1 + \cdots + \beta_n X_n}}{1 + e^{k+\beta_1 X_1 + \cdots + \beta_n X_n}} \tag{44}$$

where Y is the dependent, binary variable, X_i are the predictors, β_i are the weights for these predictors, and k is a constant value. Parameters β_i and k are learned through the process of training the regressor. In our case, Y informs whether a certain excerpt was chosen or not according to the following predictors: the F-measure (X_1), the signed difference between precision and recall (X_2), and the absolute difference between precision and recall (X_3).

A total of $23 \times 5 \times 2 = 230$ observations were used as input to the regression with the parameters defined above, since 23 subjects took part in the experiment and there were five different tracks with two versions per excerpt. A Hosmer & Lemeshow test (Hosmer and Lemeshow, 2004) was run in order to understand the predictive ability of the input data. If this test is not statistically significant ($p > 0.05$), we know that logistic regression can indeed help us predict Y. As we can see on Table 16, a value of $p = 0.763$ is obtained ($\chi^2 = 4.946$, with 8 degrees of freedom) which describes that the data for this type of analysis fits well, and that the regressor has enough predictive power.

In Table 17 the analysis of the results of the learned model is shown. $\mathbf{F}_3$ is expected to not be able to predict the selected version since it is similar in both versions, and that is exactly what can be seen in the results ($p = 0.992$), providing clear evidence that the metric is inexpressive and perceptually irrel-

Goodness-of-Fit Test on Logistic Regression Model			
Test	χ^2	df	p
Hosmer & Lemeshow	4.946	8	.763

Table 16: Hosmer & Lemeshow test, showing the capacity of the model to predict results, given the high value of p.

evant for the evaluation of segmentation algorithms. Moreover, $\mathbf{P}_3 - \mathbf{R}_3$ can predict the results in a statistically significant manner ($p = 0.000$), while the absolute difference $|\mathbf{P}_3 - \mathbf{R}_3|$, though better than the F-measure, has low predictive power ($p = 0.482$). This shows the asymmetrical relationship between $\mathbf{P}_3$ and $\mathbf{R}_3$: it is not sufficient that $\mathbf{P}_3$ and $\mathbf{R}_3$ are different, but the sign matters: $\mathbf{P}_3$ has to be higher than $\mathbf{R}_3$.

Logistic Regression Analysis of Experiment 2						
Predictor	β	S.E. β	Wald's χ^2	df	p	e^{β}
$\mathbf{F}_3$	-.012	1.155	.000	1	.992	.988
$\mathbf{P}_3 - \mathbf{R}_3$	2.268	.471	23.226	1	.000	1.023
$\lvert\mathbf{P}_3 - \mathbf{R}_3\rvert$	-.669	.951	.495	1	.482	.512
k	.190	.838	.051	1	.821	1.209

Table 17: Analysis of Experiment 2 data using logistic regression. According to these results, $\mathbf{P}_3 - \mathbf{R}_3$ can predict the version of the excerpt that subjects will choose.

Based on this experiment, and at least for this set of tracks, the following can be claimed:

1) The F-measure does not sufficiently characterize the perception of boundaries.

2) Precision is clearly more important than recall.

3) There might be a better parameterization of the F-measure that encodes relative importance.

An attempt to address this final point is presented in the following subsection.

6.4 Enhancing the F-Measure

In the previous subsections empirical evidence indicates that high precision is perceptually more relevant than high recall for the evaluation of segmentation algorithms. Here the values of precision and recall are weighted differently in order to obtain a more expressive and perceptually informative version of the F-measure for benchmarking estimated boundaries.

The F-measure, also known as the F_1-measure, is a specific case of the F_α-measure:

$$F_\alpha = (1+\alpha^2)\frac{P \cdot R}{\alpha^2 P + R} \tag{45}$$

where $\alpha = 1$, resulting in P and R having the same weight. However, it is clear from the equation that $\alpha < 1$ should be imposed in order to give more importance to P to act accordingly with the experimental results. Note that an algorithm that outputs fewer boundaries does not necessarily increase its F_α-measure, since the fewer predicted boundaries could still be incorrect. Nevertheless, the question remains: how could the value of α be determined?

The following is proposed: to sweep α from 0 to 1 using a step size of 0.05 and perform logistic regression analysis at each step using the F_α-measure (instead of the F_1-measure) as the only predictor (X_1=F_α, $n = 1$). The p-value of the F_α-measure predicting subject preference in Experiment 2 across all α is shown in Figure 37.

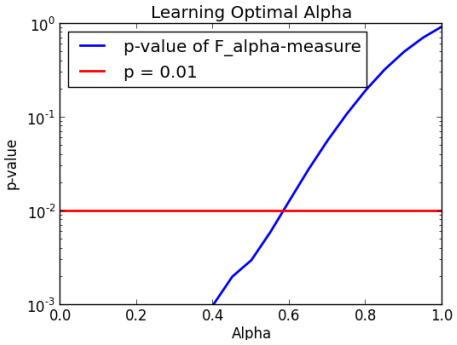


Figure 37: Statistical significance of the F_α-measure predicting the perceptual preference of a given evaluation for $\alpha \in [0, 1]$

Importantly, data from Experiment 2 is limited as it does not include information at the limits of the difference between precision and recall. Therefore, the proposed model always predicts that decreases of α always lead to highest predictive power. Naturally, this is undesirable since it would eventually remove all influence from recall in the measure and favor solutions that may contain too few boundaries. It is expected that, as $\mathbf{P}_3 - \mathbf{R}_3$ increases, at some point subject preference will decrease, as preserving a minimum amount of recall becomes more important. Consequently, it should be possible to choose the first value of α (0.58) for which F_α-based predictions of subject preference become accurate at the statistically significant level of 0.01.

In order to illustrate that this value behaves as expected, the evaluation of Experiments 1 and 2 using the $F_{0.58}$-measure (i.e., $\alpha = 0.58$) is re-run. For Experiment 1, 83.3% for HP and 72.1% for HR are obtained (instead of 77.7% and 79.4% respectively). For Experiment 2, the values of HP and HR become 71.8% and 58.9% respectively, whereas they were both 65.0% originally. This shows how the new approximated measure is well coordinated with the preferences of the subjects from Experiments 1 and 2, therefore making this evaluation of section boundaries more expressive and perceptually relevant.

It is important to note that this specific α value is highly dependent on the empirical data, which is one of the limitations of using reduced data sets as compared to the real world—in other words, there might exist a high degree of overfitting to our data. Nonetheless, based on these findings, there should be a value of $\alpha < 1$ that better represents the relative importance of precision and recall.

Finally, and to close this section, the results of the two main experiments discussed here are available on line*.

7 Summary

In this chapter the use of multiple annotations to assess automatic methods of music segmentation has been encouraged by collecting five additional annotations for fifty different tracks. It has been shown that the scores of the challenging tracks significantly vary depending on the annotator used for the evaluation, confirming the difference in perception of musical boundaries, and suggesting that a combination of annotators might yield more robust and perceptually enhanced evaluations. Four different types of merging multiple annotations have been presented, along with a description on how the current algorithms could be evaluated against these merged annotations with two novel metrics: weighted flat boundaries and hierarchical boundaries evaluations. The quality of these merges has been analyzed by cross-validating the uniformity of the scores using sets of two and three annotators, and it has been shown that two annotators per track seems to be sufficient in order to analyze challenging tracks, producing similar results than when evaluating tracks

* `http://www.urinieto.com/NYU/ISMIR14-BoundariesExperiment.zip`

annotated by just one human (i.e., those in the control group). Furthermore, it has been discussed that a single annotation appears to be enough when evaluating these control tracks. This could potentially help reduce the cost of annotating datasets by only having more than one annotation per track in the most challenging tracks, which can be automatically identified by ranking them using the methodology presented in the previous chapter. Moreover, merging types III and IV (i.e., the ones that produce hierarchical annotations) seem to yield results in which the comparison between algorithms is more consistent across sets, while types I and II (i.e., the ones that yield weighted flat ones) produce annotations with similar average scores across sets.

Additionally, and independently on how many references are available per track, a series of experiments were presented on the F-measure of the Hit Rate metric that conclude that precision is perceived as more relevant than recall when evaluating boundaries in music. Furthermore, the shortcomings of the current F-measure were exposed when evaluating results in a perceptually meaningful way. By using the general form of the F-measure, and based on these experiments, it should be possible to obtain more relevant results when precision is emphasized over recall ($\alpha < 1$). Since this weighted F-measure would be perceptually enhanced, I believe that it should be more effective when evaluating music boundaries.

CHAPTER VII

CONCLUSIONS

This dissertation has addressed the problem of the automatic discovery of structure in music from audio signals by introducing novel approaches and proposing perceptually enhanced evaluations. In this final chapter its findings and their implications will be discussed, along with a more speculative discussion about the future perspectives of this problem.

1 Findings

First, the problem of music structure analysis has been reviewed from the fields of MIR and MPC, discussing the limitations and current challenges in both disciplines. One of the main differences is the type of data these two fields tend to work with: in MIR it is common to operate with large amounts of musical data to implement automatic methods, while in MPC significant human data are typically required to design generic models. This motivated the possibility of combining both fields to have a better understanding of this problem. An overview of the most standard techniques to discover structure in music have also been presented, along with a transparent open source software called `mir_eval` to evaluate them (Chapter II). Additionally, a series of novel MIR techniques have been presented (Chapters III and IV), some of which are later evaluated using perceptually enhanced metrics with the aim of combining MIR

and MPC under the problem of the automatic discovery of music structure (Chapters V and VI).

The first algorithm presented, the compression criteria for music summaries introduced in Section 3 of Chapter III, is one of the few existing ones that produces audible music summaries containing the most representative parts of a given track. Based on the presented experiments, Tonnetz features are better candidates to approach this task than regular PCP features, likely due to the continuous-valued geometric space in which Tonnetz reside, where beat estimation errors are smoothly interpolated as opposed to PCPs.

The second algorithm, introduced in Section 4 of the same chapter, aims at discovering all the repeated musical parts (or patterns) of a given piece. This method employs standard techniques commonly used for the task of music segmentation, and it yields state-of-the-art results when using audio as input. The most significant finding of this approach is perhaps the fact that these standard and simple techniques yield better results when establishing patterns in a piece compared to other more complex approaches that take exceedingly long times to process*.

Next, an algorithm to discover the large-scale segments of a musical piece has been presented in Section 4 of Chapter IV, which uses an unsupervised machine learning method that adds a convex constraint to the standard Non-negative Matrix Factorization process. This constraint yields factorized matrices that contain more meaningful prototypes of the different unique segments, from which the boundaries can be extracted and then grouped based on their similarity. This technique focuses on the *homogeneous* type of segments

* Some of the existing techniques can take weeks to produce outputs (Lartillot, 2014).

of a piece, and it yields the best results compared to other techniques that also aim at extracting these type of segments.

To finish the set of MIR techniques, the usage of 2D-Fourier Magnitude Coefficients to approach the structural grouping problem (i.e., label the different segments by their acoustical similarity) has been proposed in Section 3 of the same chapter. These coefficients, which are applied to this problem for the first time, can be employed in an efficient algorithm to group the segments, with results that can be considered state-of-the-art when using specific datasets and metrics.

As a commonality to these methods, the parameters chosen to extract music features from an audio signal have a strong impact on the actual results produced by the algorithms. The importance of feature design is a widely discussed topic in MIR (Pachet and Zils, 2004; Hamel and Eck, 2010; Humphrey et al., 2012b), and in this dissertation it has manifested when running implementations of already existing algorithms (altering the feature extraction process only) and comparing the obtained results with the ones reported in the original publication.

After presenting these four novel MIR techniques, the idea of applying more MPC-oriented approaches has been considered to obtain perceptually relevant evaluations for music segmentation. In this task it is common to evaluate algorithms against a *ground-truth* that only contains a single human reference per track, which can be problematic due to the inherent subjectivity when perceiving musical structure. A methodology to automatically obtain the most difficult tracks for machines to annotate has been presented in order to design a human study to collect multiple human annotations (Chapter V). To do so, a novel open source framework called MSAF has been introduced.

This framework contains the most relevant music segmentation algorithms and it uses `mir_eval` to transparently evaluate them.

MSAF uses JAMS, a new format to contain multiple annotations for several tasks in a single file, which simplifies the dataset design and the analysis of agreement across different human references. It has been discussed that the genre of western pop music is not usually considered when automatically selecting the most challenging tracks, thus exposing one of the reasons why this type of music is the one for which computational models have been best optimized. The human study to collect additional annotations has been described, where five new annotations for fifty tracks were stored.

Finally, these novel annotations have been analyzed in Chapter VI, confirming the problem of having *ground-truth* datasets with a single annotator per track due to the high degree of disagreement among annotators for the challenging tracks. To alleviate this, these annotations have been merged to produce a more robust human reference annotation. This finding could be extrapolated to other subjective MIR tasks to make them more perceptually relevant. As a further finding, the least challenging tracks (the ones in a controlled group), do not suffer from the subjectivity problem, so no additional annotations should be required in order to successfully evaluate them. Since these merged segments contain weighted flat or hierarchical annotations, two novel methods to evaluate them have been presented. Additionally, and to conclude, the standard F-measure of the hit rate measure to evaluate music segmentation has been analyzed when access to additional annotations is not possible, and it has been shown, via multiple human studies, that precision seems more perceptually relevant than recall.

2 Implications

By having published and discussed four new competitive methods to automatically analyze the structure of a musical piece from an audio signal, this work brings us a step closer to a world where machines can better discover the structure of music. More specifically, the publication of these algorithms has the following practical implications:

- The novel music summarization algorithm could potentially be used to produce more meaningful *audio previews*, typically available when purchasing tracks on-line or browsing large music catalogs.

- The presented pattern discovery method might inspire future researchers to approach this problem from a music segmentation point of view, thus simplifying the complex heuristics that have been used for the past years and obtaining similar or superior results.

- The two new music segmentation algorithms could be applied to large digital music catalogs, such that listeners can obtain segment-level recommendations. Moreover, intra-piece navigation could also be enhanced by adding functionalities like *skipping to next section*.

Besides these four MIR methods, the presented framework MSAF could also have significant implications in the field. Researchers interested in the automatic discovery of structure in music are encouraged to use MSAF in order to simplify their implementation processes and easily analyze and identify the strengths and weaknesses of their methods. MSAF has been designed to be effortlessly extensible, with the hopes of becoming the central infrastructure

to develop music segmentation algorithms. Additionally, MSAF makes use of both JAMS and `mir_eval`, resulting in a good showcase example of a direct usage of these novel format and package, respectively. Given that these methods and MSAF are open source projects, the reproducibility of these results should not only be easy to perform, but also encouraged. This transparency, which is of high importance in any science publication, might lead to code improvements that could result in better MIR approaches or an enriched framework to analyze them.

Another relevant feature of MSAF is its allowance for a ranking of the tracks of a specific dataset using multiple algorithms in order to know the most and least challenging ones from a machine point of view. This, plus the finding that suggests that a single human annotation for the tracks that are "easy" to segment is enough, could make MSAF the starting point for a desired methodology to automatically identify the tracks for which additional annotations are needed, thus saving both resources and time while having a more robust dataset. This could be useful in, e.g., MPC scenarios in which human studies are designed to evaluate challenging structural pieces or in the industry where specific algorithms must be evaluated in a more perceptually enhanced manner.

By merging multiple annotations to provide an alternative evaluation that can better assess music segmentation, MIR practitioners could easily tune their solutions to produce systems that are closer to user preference, without altering the standard methodology of developing algorithms in MIR (i.e., to optimize algorithms given an estimated output, an annotated reference, and an evaluation to compare them). This perceptual evaluation would simply change the goal (i.e., the score to optimize), leaving the rest of the process untouched.

This merging of annotations could also be applied to other MIR areas, as it is one of the central challenges when assessing these tasks as objectively as possible (Urbano et al., 2013). In fact, it has already been considered in chord recognition (Ni et al., 2013) and beat tracking (Davies and Böck, 2014). Nevertheless, due to the benefits of having multiple human annotations, upcoming dataset publishers are encouraged to add more than one reference of music segmentation for each track*.

Moreover, the findings regarding the perceptual alignment of the F-measure of the hit rate metric should result in a more careful treatment of this metric. Whenever considering music segmentation results, researchers should have in mind the perceptual preference towards precision over recall (at least when using 3 second windows), which should benefit the discussion and analysis of their methods. Similarly to the merging of multiple annotations, this more perceptually relevant F-measure could easily yield MIR solutions that better behave as users would expect.

3 Future Perspectives

The future of this field is both fascinating and challenging. As of now, and as reviewed throughout this work, it is a hard task to automatically obtain the complete structure of music from a given piece. Nevertheless, it may seem that in the not so distant future results that are almost as good as human estimations might be reached for specific types of music at a certain level of their structural hierarchy (e.g., large-scale sections for determined types of pop

* One such dataset already exists (Smith et al., 2011), which contains two human annotations for most of its tracks.

music using the hit rate at 3 seconds (Peeters and Bisot, 2014)). An algorithm or a model to capture the entire essence of the structure of music regardless of the genre might not be developed in the next years, however it appears to be the correct path to combine more engineering-based approaches of MIR with more psychological methods from MPC.

As it has been discussed in this work, there might be more than one valid set of segments that define the structure of a given track. In order to address this, it has been shown how multiple annotations could be merged to reduce this subjectivity factor. However, it might also be interesting to explore the possibility to design algorithms that produce more than one valid result. This way, and acknowledging the ambiguity of this task, users could potentially choose the result that better fits their needs.

Promising steps towards more generalizable algorithms have been recently taken regarding the usage of *big data* applied to music segmentation. More specifically, feed-forward convolutional neural networks trained over a relatively large dataset yield encouraging good results (Ullrich et al., 2014; Ullrich, 2014; Schlüter et al., 2014). Such systems also seem to yield promising results in other areas of MIR (Humphrey et al., 2012b). It is likely that as, more powerful computers and larger datasets become available, these methods will produce more accurate results. Regardless, it remains to be seen how these approaches could be used for other tasks besides the identification of boundaries (e.g., the structural grouping of the segments), which are likely to be developed in the following years.

The usage of these deep learning methodologies exposes the need of having more human annotated data to train these networks to yield more robust results. However, and when it is not realistic to have access to such data,

it is interesting to frame some of the existing solutions under the context of deep learning. For example, the convex constraint in NMF presented in this dissertation could be seen as an additional layer of a deep learning system, where a sparsity constraint is added to obtain a more meaningful decomposition (this type of perspective has been recently taken when understanding NMF (Sprechmann et al., 2014)). Having some of these precomputed features would help reduce the amount of additional human annotations needed for the deep belief networks to hypothetically discover the structure of music.

As of late, a tendency towards the automatic discovery of *hierarchical* structure in music has originated, where some approaches are not only able to identify the large-scale segments but other more specific layers (McFee and Ellis, 2014b,a). The bottom layers would contain motives or short riffs, which would be directly linked with the short patterns of the pattern discovery task. It is possible that in the near future a higher degree of overlap between the music segmentation and the pattern discovery tasks occurs. Now that a metric to evaluate hierarchical approaches has been presented, hierarchical approaches should be easily compared and analyzed, thus encouraging researchers to publish approaches that produce this type of segmentations.

As presented in this work, the merging of the human annotated segment boundaries appears to be beneficial towards obtaining a more robust dataset on which to compare estimated boundaries. However, it is still unclear how to merge the segment labels as well, such that these perceptually enhanced datasets can be employed in the whole task of music segmentation, not only on the boundary identification subtask. To do so, the merged boundaries as presented here could be used when collecting multiple segment label data in future experiments.

Finally, it remains unclear if, at some point, too many annotators could become problematic for the proposed merging methods (e.g., outlier boundaries might hardly be considered as correct), and future work should focus on obtaining more annotations not only for the challenging tracks but for all of them. Further steps should be taken in order to determine a more specific and generalizable value of α in our proposed weighted F-measure. Alternatively, smaller time windows for this metric might more accurately align to perception without having to weight precision and recall differently, as it has been recently shown in the context of beat-tracking (Davies and Böck, 2014). It is likely that 3 second windows (the standard nowadays) are too far from the human understanding of musical boundaries. Nevertheless, the importance of having additional human data when evaluating automatic approaches to discover structure in music has been stated in this work, with the hopes that in the near future this becomes a standard practice in the field.

4 Outro

In this work I have attempted to make machines slightly better at discovering the structure of music. To undertake this endeavor, not only engineering techniques, but also cognitively inspired evaluation methods have been proposed with the aim of narrowing the sometimes large gap between the fields of MIR and MPC. Given the —usually— intended ambiguity of music, and the diverse impact it has on humans, it is to be expected that our perception (and its disagreements) must be by some means encoded when developing automatic approaches. Perhaps the beauty of the structure of music resides in the differences it produces on perception. And even though we are still far

from having real *thinking machines*, a time is approaching when computers could discover certain structures so that we, humans, may better understand why music is structured the way it is.

BIBLIOGRAPHY

Abdallah, S., Noland, K., Sandler, M., Casey, M., and Rhodes, C. (2005). Theory and Evaluation of a Bayesian Music Structure Extractor. In *Proc. of the 6th International Society of Music Information Retrieval Conference*, pages 420–425, London, UK. 18

Aucouturier, J. J. and Bigand, E. (2012). Mel Cepstrum & Ann Ova: The Difficult Dialog Between MIR and Music Cognition. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal. 13

Barrington, L., Turnbull, D., and Lanckriet, G. (2012). Game-powered Machine Learning. *Proceedings of the National Academy of Sciences of the United States of America*, 109(17):6411–6. 41

Bartsch, M. and Wakefield, G. (2005). Audio thumbnailing of popular music using chroma-based representations. *Multimedia, IEEE Transactions on*, 7(1):96–104. 58

Bello, J. P. (2009). Grouping Recorded Music by Structural Similarity. In *Proc. of the 10th International Society of Music Information Retrieval*, number Ismir, pages 531–536, Utrecht, Netherlands. 19

Bello, J. P., Grosche, P., Müller, M., and Weiss, R. J. (2011). Content-based Methods for Knowledge Discovery in Music. *ACM Computers in Entertainment*, 1(1):1–24. 4

Bent, I. and Drabkin, W. (1987). *The New Grove Handbook in Music Analysis*. Macmillian Press. London. 14

Bertin-Mahieux, T. and Ellis, D. P. W. (2012). Large-Scale Cover Song Recognition Using The 2D Fourier Transform Magnitude. In *Proc. of the 13th In-*

ternational Society for Music Information Retrieval Conference, pages 241–246, Porto, Portugal. 37, 106, 117

Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., and Lamere, P. (2011). The Million Song Dataset. In *Proc of the 12th International Society of Music Information Retrieval*, Miami, FL, USA. 128, 144

Bimbot, F., Deruty, E., Sargent, G., and Vincent, E. (2012). Semiotic Structure Labeling of Music Pieces: Concepts, Methods and Annotation Conventions. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, pages 235–240, Porto, Portugal. 19

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., and Serra, X. (2013). Essentia: An Audio Analysis Library for Music Information Retrieval. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, pages 493–498, Curitiba, Brazil. 89, 122

Bruderer, M. J. (2008). *Perception and Modeling of Segment Boundaries in Popular Music*. PhD thesis, Universiteitsdrukkerij Technische Universiteit Eindhoven. 146

Bruderer, M. J., McKinney, M., and Kohlrausch, A. (2006a). Structural Boundary Perception in Popular Music. In *Proc. of the 7th International Society of Music Information Retrieval Conference*, volume 4, pages 198–201, Victoria, BC, Canada. 29, 146

Bruderer, M. J., Mckinney, M. F., and Kohlrausch, A. (2006b). Perception of structural boundaries in popular music. In *Proc. of the 9th International Conference on Music Perception and Cognition*, number 1983, pages 157–162, Bologna, Italy. 27, 29, 111

Bruderer, M. J., Mckinney, M. F., and Kohlrausch, A. (2009). The Perception of Structural Boundaries in Melody Lines of Western Popular Music. *MusicæScientiæ*, 13(2):273–313. 5, 16, 29, 120, 145, 146, 173

Burgoyne, J. A., Wild, J., and Fujinaga, I. (2011). An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis. In *Proc. of the 12th*

International Society of Music Information Retrieval, pages 633–638, Miami, FL, USA. 128

Cambouropoulos, E. (2001). The Local Boundary Detection Model (LBDM) and its Application in the Study of Expressive Timing. In *Proc. of the International Computer Music Conference*, La Havana, Cuba. 14, 27

Cannam, C., Landone, C., Sandler, M., and Bello, J. P. (2006). The Sonic Visualiser: A Visualisation Platform for Semantic Descriptors from Musical Signals. In *Proc. of the 7th International Conference on Music Information Retrieval*, pages 324–327, Victoria, BC, Canada. 135, 168

Casey, M., Rhodes, C., and Slaney, M. (2008a). Analysis of Minimum Distances in High-Dimensional Musical Spaces. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(5):1015–1028. 61

Casey, M. A. and Slaney, M. (2006). The Importance Of Sequences In Musical Similarity. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 5–8, Toulouse, France. IEEE. 4

Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., and Slaney, M. (2008b). Content-based Music Information Retrieval: Current Directions and Future Challenges. *Proceedings of the IEEE*, 96(4). 5, 6

Cho, T. and Bello, J. P. (2014). On the Relative Importance of Individual Components of Chord Recognition Systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2):477–492. 37, 91

Clarke, E. F. (1989). Mind the Gap: Formal Structures and Psychological Processes in Music. *Contemporary Music Review*, 3(1):1–13. 23

Clarke, E. F. and Krumhansl, C. L. (1990). Perceiving Musical Time. *Music Perception*, 7(3):213–251. 27

Collins, T. (2013). Discovery of Repeated Themes & Sections. 20, 52, 74, 75, 82, 83

Collins, T., Arzt, A., Flossmann, S., and Widmer, G. (2014a). SIARCT-CFP: Improving Precision and the Discovery of Inexact Musical Patterns in Point-set Representations. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, pages 549–554, Curitiba, Brazil. 21, 82, 83, 85, 86

Collins, T., Sebastian, B., Krebs, F., and Widmer, G. (2014b). Bridging the Audio-Symbolic Gap: The Discovery of Repeated Note Content Directly From Polyphonic Music Audio. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, pages 1–12, London, UK. 20, 21, 83, 84

Conklin, D. and Anagnostopoulou, C. (2001). Representation and Discovery of Multiple Viewpoint Patterns ∗. In *Proc. of the International Computer Music Conference*, pages 479–485, La Havana, Cuba. 21

Cooper, M. and Foote, J. (2003). Summarizing popular music via structural similarity analysis. In *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 127–130, New Paltz, NY, USA. IEEE. 22, 58

Dannenberg, R. B. and Goto, M. (2008). Music Structure Analysis from Acoustic Signals. In Havelock, D., Kuwano, S., and Vorländer, M., editors, *Handbook of Signal Processing in Acoustics*, pages 305–331. Springer, New York, NY, USA. 4

Davies, M. E. P. and Böck, S. (2014). Evaluating the Evaluation Measures for Beat Tracking. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 637–642, Taipei, Taiwan. 5, 28, 50, 189, 192

Deliège, I., Mélen, M., Stammers, D., and Cross, I. (1996). Musical Schemata in Real-Time Listening to a Piece of Music of Cambridge. *Music Perception*, 14(2):117–159. 27

Ding, C., Li, T., and Jordan, M. I. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):45–55. 90, 92, 93

Downie, J. S. (2008). The Music Information Retrieval Evaluation Exchange (2005–2007): A Window into Music Information Retrieval Research. *Acoustical Science and Technology*, 29(4):247–255. 41

Downie, J. S., Byrd, D., and Crawford, T. (2009). Ten years of ISMIR: Reflections on challenges and opportunities. In *Proceedings of the 10th International Conference on Music Information Retrieval*, number Ismir, pages 13–18. 6

Ehmann, A. F., Bay, M., Downie, J. S., Fujinaga, I., and Roure, D. D. (2011). Music Structure Segmentation Algorithm Evaluation: Expanding on MIREX 2010 Analyses and Datasets. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, pages 561–566, Miami, FL, USA. 47

Ellis, D. P. W. and Poliner, G. E. (2007). Identifying 'Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking. In *Proc. of the 32nd IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1429–1432, Honolulu, HI, USA. 36, 91, 123

Euler, L. (1739). *Tentamen Novae tTeoriae Musicae ex Certissismis Harmoniae Principiis Dilucide Expositae*. PhD thesis, Saint Petersburg Academy. 32

Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., and Vee, E. (2006). Comparing Partial Rankings. *SIAM Journal on Discrete Mathematics*, 20(3):628–648. 159

Flexer, A. (2014). On Inter-rater Agreement in Audio Music Similarity. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 245–250, Taipei, Taiwan. 5, 28

Foote, J. (2000). Automatic Audio Segmentation Using a Measure Of Audio Novelty. In *Proc. of the IEEE International Conference of Multimedia and Expo*, pages 452–455, New York City, NY, USA. 17, 32, 39, 124, 125

Forth, J. C. (2012). *Cognitively-motivated Geometric Methods of Pattern Discovery and Models of Similarity in Music*. PhD thesis, Glodsmiths, University of London. 21, 27

Forth, J. C. and Wiggins, G. A. (2009). An Approach for Identifying Salient Repetition in Multidimensional Representations of Polyphonic Music. In Chan, J., Daykin, J. W., and Rahman, M. S., editors, *London Algorithmics 2008: Theory and Practice*, pages 44–58. UK: College Publications. 21

Fujishima, T. (1999). Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music. In *In Proc. of the International Conference on Computer Music*, pages 464–467, Beijing, China. 31

Ganchev, T., Fakotakis, N., and Kokkinakis, G. (2005). Comparative Evaluation of Various MFCC Implementations on the Speaker Verification Task. In *Proc. of the International Conference on Speach and Computer*, pages 191–194, Patras, Greece. 123

Goldman, R. F. (1961). Varèse: Ionisation; Density 21.5; Intégrales; Octandre; Hyperprism; Poème Electronique by Robert Craft; Varèse. *The Musical Quarterly*, 47(1):133–134. 144

Gómez, E. (2006). Tonal Description of Polyphonic Audio for Music Content Processing. *INFORMS Journal on Computing*, 18(3):294–304. 32, 122

Goto, M. (2003). A Chorus-section Detecting Method for Musical Audio Signals. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, pages 437–440, Hong Kong, China. 17, 40

Grohganz, H., Clausen, M., Jiang, N., and Müller, M. (2013). Converting Path Structures into Block Structures using Eigenvalue Decomposition of Self-Similarity Matrices. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil. 117

Grosche, P. (2010). What Makes Beat Tracking Difficult? A Case Study on Chopin Mazurkas. In *Proc. of the International Society of Music Information Retrieval*, Utrecht, Netherlands. 101, 145

Grosche, P. and Müller, M. (2011). Extracting Predominant Local Pulse Information From Music Recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1688–1701. 56, 57, 66

Grosche, P., Serrà, J., Müller, M., and Arcos, J. L. (2012). Structure-Based Audio Fingerprinting For Musical Retrieval. In *Proc of the 13th International Society of Music Information Retrieval*, number Ismir, pages 55–60, Porto, Portugal. 19

Hamanaka, M., Hirata, K., and Tojo, S. (2004). Automatic generation of grouping structure based on the GTTM. In *Proc. of the International Computer Music Conference*, Miami, FL, USA. 14

Hamel, P. and Eck, D. (2010). Learning Features from Music Audio with Deep Belief Networks. In *Proc. of the 11th International Society of Music Information Retrieval*, pages 339–344, Utrecht, Netherlands. 185

Harte, C., Sandler, M., and Gasser, M. (2006). Detecting Harmonic Change in Musical Audio. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, pages 21–26, Santa Barbara, CA, USA. ACM Press. 33, 56, 123

Holzapfel, A., Davies, M. E. P., Zapata, J. R., Oliveira, J. L., and Gouyon, F. (2012). Selective Sampling for Beat Tracking Evaluation. *IEEE Transactions On Audio, Speech, And Language Processing*, 20(9):2539–2548. 36

Hosmer, D. W. and Lemeshow, S. (2004). *Applied Logistic Regression*. John Wiley & Sons. 177

Hoyer, P. O. (2004). Non-negative Natrix Factorization with Sparseness Constraints. *The Journal of Machine Learning Research*, 5:1457–1469. 93

Hubert, L. and Arabie, P. (1985). Comparing Partitions. *Journal of Classification*, 2(1):193–218. 47

Humphrey, E., Cho, T., and Bello, J. (2012a). Learning a Robust Tonnetzspace Transform for Automatic Chord Recognition. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 453–456, Kyoto, Japan. 34

Humphrey, E. J., Bello, J. P., and LeCun, Y. (2012b). Moving Beyond Feature Design: Deep Architecture and Automatic Feature Learning in Music Informatics. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, pages 403–408, Porto, Portugal. 185, 190

Humphrey, E. J., Nieto, O., and Bello, J. P. (2013). Data Driven and Discriminative Projections for Large-Scale Cover Song Identification. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil. 106

Humphrey, E. J., Salamon, J., Nieto, O., Forsyth, J., Bittner, R. M., and Bello, J. P. (2014). JAMS: A JSON Annotated Music Specification for Reproducible MIR Research. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 591–596, Taipei, Taiwan. 10, 127

Huron, D. (2006). *Sweet Anticipation: Music and the Psychology of Expectation*. MIR Press, Cambridge, MA. 24

Janssen, B., Haas, W. B. D., Volk, A., and Kranenburg, P. V. (2013). Discovering Repeated Patterns in Music: State of Knowledge, Challenges, Perspectives. In *Proc. of the 10th International Symposium on Computer Music Multidisciplinary Research*, Marseille, France. 20, 74

Jiang, N. and Müller, M. (2013). Automated Methods for Analyzing Music Recordings in Sonata Form. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil. 19

Juslin, P. N. and Västfjäll, D. (2008). Emotional responses to music: the need to consider underlying mechanisms. *The Behavioral and Srain sciences*, 31(5):559–621. 29

Kaiser, F. and Peeters, G. (2013). A Simple Fusion Method of State and Sequence Segmentation for Music Structure Discovery. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil. 18, 35

Kaiser, F. and Sikora, T. (2010). Music Structure Discovery in Popular Music Using Non-Negative Matrix Factorization. In *Proc. of the 11th International Society of Music Information Retrieval*, pages 429–434, Utrecht, Netherlands. 18, 32, 90, 96, 98, 99, 115, 117

Karydis, I., Radovanovic, M., Nanopoulos, A., and Ivanovic, M. (2010). Looking Trough the "Glass Ceiling": A Conceptual Framework for the Problems of Spectral Similarity. In *Proc of the 11th International Society of Music Information Retrieval*, Utrecht, Netherlands. 5

Kim, S., Kwon, S.-b., and Kim, H. (2006). A Music Summarization Scheme using Tempo Tracking and Two Stage Clustering. In *Proc. of the IEEE Workshop on Multimedia Signal Processing*, pages 225–228, Victoria, BC, Canada. IEEE. 22

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *International Joint Conference on Artificial Intelligence*, pages 1137–1145. 162

Krumhansl, C. L. (1996). A Perceptual Analysis Of Mozart ' s Piano Sonata K . 282. *Music Perception*, 13(3). 24

Krumhansl, C. L. and Castellano, M. a. (1983). Dynamic Processes in Music Perception. *Memory & Cognition*, 11(4):325–334. 28

Lartillot, O. (2005). Multi-Dimensional motivic pattern extraction founded on adaptive redundancy filtering. *Journal of New Music Research*, 34(4):375–393. 21

Lartillot, O. (2014). In-depth Motivic Analysis Based on Multiparametric Closed Pattern and Cyclic Sequence Mining. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 361–366, Taipei, Taiwan. 21, 87, 184

Lee, D. D. and Seung, H. S. (2000). Algorithms for Non-negative Matrix Factorization. *Advances in Neural Information Processing Systems*, 13:556–562. 92

Lemström, K. (2000). *String Matching Techniques for Music Retrieval*. PhD thesis, University of Helsinki, Finland. 21

Lerdahl, F. and Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press. 14, 27

Levy, M. and Sandler, M. (2008). Structural Segmentation of Musical Audio by Constrained Clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):318–326. 18, 35, 46, 99, 104, 115, 117, 124, 156, 167, 175

Li, S., Hou, X. W. H. X. W., Zhang, H. J. Z. H. J., and Cheng, Q. S. C. Q. S. (2001). Learning spatially localized, parts-based representation. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, volume 1, pages 1–6. 93

Livingstone, S. R., Palmer, C., and Schubert, E. (2012). Emotional response to musical repetition. *Emotion (Washington, D.C.)*, 12(3):552–67. 29

Logan, B. (2000). Mel Frequency Cepstral Coefficients for Music Modeling. In *Proc. of the 1st International Society for Music Information Retrieval Conference*, Plymouth, MA, USA. 34

Logan, B. and Chu, S. (2000). Music summarization using key phrases. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 749–752, Istanbul, Turkey. 22

Lorenzo, P. D. and Maio, G. D. (2006). The Hausdorff Metric in the Melody Space: A New Approach to Melodic Similarity. In *Proc. of the 9th International Conference on Music Perception and Cognition*, Bologna, Italy. 21

Lukashevich, H. (2008). Towards Quantitative Measures of Evaluating Song Segmentation. In *Proc. of the 10th International Society of Music Information Retrieval*, pages 375–380, Philadelphia, PA, USA. 47, 48, 49, 104, 112, 116, 159

Margulis, E. H. (2005). A Model of Melodic Expectation. *Music Perception: An Interdisciplinary Journal*, 22(4):663–714. 25

Marwan, N., Carmenromano, M., Thiel, M., and Kurths, J. (2007). Recurrence Plots for the Analysis of Complex Systems. *Physics Reports*, 438(5-6):237–329. 39

Mauch, M., Cannam, C., Davies, M., Dixon, S., Harte, C., Kolozali, S., Tidhar, D., and Sandler, M. (2009a). OMRAS2 Metadata Project 2009. In *Late Breaking Session of the 10th International Society of Music Information Retrieval*, page 2009, Kobe, Japan. 128, 134

Mauch, M. and Dixon, S. (2010). Approximate Note Transcription for the Improved Identification of Difficult Chords. In *Proc. of the 11th International Society of Music Information Retrieval*, number 1, pages 135–140, Utrecht, Netherlands. 56

Mauch, M., Noland, K., and Dixon, S. (2009b). Using Musical Structure to Enhance Automatic Chord Transcription. In *Proc. of the 10th International Society of Music Information Retrieval*, pages 231–236, Kobe, Japan. 17, 106, 116, 117, 128

McFee, B. and Ellis, D. P. W. (2014a). Analyzing Song Structure with Spectral Clustering. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 405–410, Taipei, Taiwan. 18, 19, 191

McFee, B. and Ellis, D. P. W. (2014b). Learnign to Segment Songs With Ordinal Linear Discriminant Analysis. In *Proc. of the 39th IEEE International Conference on Acoustics Speech and Signal Processing*, pages 5197–5201, Florence, Italy. 18, 19, 100, 115, 124, 191

McVicar, M., Ellis, D. P. W., and Goto, M. (2014). Leveraging Repetition for Improved Automatic Lyric Transcription in Popular Music. In *Proc. of the 39th IEEE International Conference on Acoustics Speech and Signal Processing*, pages 3117 – 3121, Florence, Italy. 19

Meintanis, K. A. and Shipman, F. M. (2008). Creating and Evaluating Multi-Phrase Music Summaries. In *Proc. of the International Society of Music Information Retrieval*, volume 5, pages 507–512, Philadelphia, PA, USA. 22

Meredith, D. (2006). Point-set Algorithms For Pattern Discovery And Pattern Matching In Music. In Crawford, T. and Veltkamp, R. C., editors, *Proc. of the Dagstuhl Seminar on Content-Based Retrieval.*, Dagstuhl, Germany. 21

Meredith, D. (2013). COSIATEC and SIATECCompress: Pattern Discovery by Geometric Compression. In *Music Information Retrieval Evaluation eXchange*, Curitiba, Brazil. 21, 83, 84, 85, 86

Mermelstein, P. (1976). Distance Measures for Speech Recognition, Psychological and Instrumental. *Pattern Recognition and Artificial Intelligence*, 116:374–388. 34

Müller, M. (2007). *Information Retrieval for Music and Motion*. Springer. 39, 40, 41

Müller, M. and Clausen, M. (2007). Transposition-Invariant Self-Similarity Matrices. In *Proc. of the 8th International Conference on Music Information Retrieval*, pages 47–50, Vienna, Austria. 75

Müller, M., Grosche, P., and Jiang, N. (2011). A Segment-Based Fitness Measure for Capturing Repetitive Structures of Music Recordings. In *ISMIR*, pages 615–620, Miami, FL, USA. 22

Müller, M. and Jiang, N. (2012). A Scape Plot Representation For Visualizing Repetitive Structures of Music Recordings. In *Proc of the 13th International Society of Music Information Retrieval*, number Ismir, pages 97–102, Porto, Portugal. 19

Narmour, E. (1992). *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. University of Chicago Press, Chicago, IL, USA. 24

Ni, Y., McVicar, M., Santos-Rodriguez, R., and De Bie, T. (2013). Understanding Effects of Subjectivity in Measuring Chord Estimation Accuracy. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(12):2607–2615. 5, 144, 189

Nichols, E. P. (2012). *Musicat: A Computer Model of Musical Listening and Analogy-Making*. PhD thesis, Indiana University. 28

Nieto, O. and Bello, J. P. (2014). Music Segment Similarity Using 2D-Fourier Magnitude Coefficients. In *Proc. of the 39th IEEE International Conference on Acoustics Speech and Signal Processing*, pages 664–668, Florence, Italy. 9, 89, 116

Nieto, O. and Farbood, M. (2013a). MIREX 2013: Discovering Musical Patterns Using Audio Structural Segmentation Techniques. In *Music Information Retrieval Evaluation eXchange*, Curitiba, Brazil. 22

Nieto, O. and Farbood, M. (2013b). MIREX 2013: Discovering Musical Patterns Using Audio Structural Segmentation Techniques. In *Music Information Retrieval Evaluation eXchange*, Curitiba, Brazil. 83, 84, 85, 86, 115, 174

Nieto, O. and Farbood, M. M. (2012). Perceptual Evaluation of Automatically Extracted Musical Motives. In *Proc. of the 12th International Conference on Music Perception and Cognition*, pages 723–727, Thessaloniki, Greece. 21

Nieto, O. and Farbood, M. M. (2014a). Identifying Polyphonic Patterns From Audio Recordings Using Music Segmentation Techniques. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 411–416, Taipei, Taiwan. 9, 55, 74

Nieto, O. and Farbood, M. M. (2014b). MIREX 2014 Entry: Music Segmentation Techniques and Greedy Path Finder Algorithm to Discover Musical Patterns. In *Music Information Retrieval Evaluation eXchange*, Taipei, Taiwan. 22

Nieto, O., Farbood, M. M., Jehan, T., and Bello, J. P. (2014). Perceptual Analysis of the F-measure for Evaluating Section Boundaries in Music. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 265–270, Taipei, Taiwan. 10

Nieto, O., Humphrey, E. J., and Bello, J. P. (2012). Compressing Audio Recordings into Music Summaries. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, pages 313–318, Porto, Portugal. 9, 55

Nieto, O. and Jehan, T. (2013). Convex Non-Negative Matrix Factorization For Automatic Music Structure Identification. In *Proc. of the 38th IEEE International Conference on Acoustics Speech and Signal Processing*, pages 236–240, Vancouver, Canada. 9, 89

Nieto, O. and Smith, J. B. L. (2013). 2013 Late-break Session on Music Segmentation. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil. 45, 100

Nikrang, A., Collins, T., and Widmer, G. (2014). PatternViewer: An Application for Exploring Repetitive and Tonal Structure. In *Late-Brake Demo of the Proc. of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan. 19

Pachet, F. and Zils, A. (2004). Automatic Extraction of Music Descriptors from Acoustic Signals. In *Proc. of the 5th International Society of Music Information Retrieval*, Barcelona, Spain. 185

Panagakis, Y., Kotropoulos, C., and Arce, G. R. (2011). $\ell 1$ -Graph Based Music Structure Analysis. In *Proc of the 12th International Society of Music Information Retrieval*, pages 495–500, Miami, FL, USA. 18

Papadopoulos, H. and Peeters, G. (2011). Joint Estimation of Chords and Downbeats From an Audio Signal. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):138–152. 128

Paulus, J. and Klapuri, A. (2009). 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. 18

Paulus, J., Müller, M., and Klapuri, A. (2010). Audio-Based Music Structure Analysis. In *Proc of the 11th International Society of Music Information Retrieval*, pages 625–636, Utrecht, Netherlands. 4, 16, 17, 32, 91

Pearce, M. T. (2005). *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition*. PhD thesis, City University, London, London, UK. 24

Pearce, M. T., Müllensiefen, D., and Wiggins, G. A. (2010). The Role of Expectation and Probabilistic Learning in Auditory Boundary Perception: A Model Comparison. *Perception*, 39(10):1365–1389. 24

Peeters, G. and Bisot, V. (2014). Improving Music Structure Segmentation Using Lag-priors. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 337–342, Taipei, Taiwan. 18, 190

Peeters, G., Burthe, A. L., and Rodet, X. (2002). Toward Automatic Music Audio Summary Generation from Signal Analysis. In *Proc. of the 3rd International Society of Music Information Retrieval*, pages 94–100, Paris, France. 18, 22, 58

Peeters, G. and Deruty, E. (2009). Is Music Structure Annotation Multi-Dimensional? A Proposal for Robust Local Music Annotation . In *Proc. of the 3rd International Worskhop on Learning Semantics of Audio Signals*, pages 75–90, Graz, Austria. 173

Pelleg, D. and Moore, A. (2000). X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *Proc. of the 17th International Conference on Machine Learning*, pages 727–734, Stanford, CA, USA. 110, 114

Peng, C.-Y. J., Lee, K. L., and Ingersoll, G. M. (2002). An Introduction to Logistic Regression Analysis and Reporting. *The Journal of Educational Research*, 96(1):3–14. 177

Raffel, C., Mcfee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and Ellis, D. P. W. (2014). mir_eval: A Transparent Implementation of Common MIR Metrics. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 367–372, Taipei, Taiwan. 9, 44, 53, 83, 100, 105, 125

Rhodes, C., Casey, M., Abdallah, S., and Sandler, M. (2006). A Markov-Chain Monte-Carlo Approach to Musical Audio Segmentation. In *Proc. of the IEEE International Conference on Acoustics Speech and Signal Processing*, pages 797–800. 18

Rockafellar, R. T., Wets, R. J., and Wets, M. (1998). *Variational Analysis*. Springer. 21

Rodríguez-López, M., Volk, A., and Bountouridis, D. (2014). Multi-Strategy Segmentation of Melodies. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 207–212, Taipei, Taiwan. 25

Romming, C. A. and Selfridge-Field, E. (2007). Algorithms for Polyphonic Music Retrieval the Hausdorff Metric and Geometric Hashing. In *Proc. of the 8th International Society for Music Information Retrieval Conference*, pages 457–462, Vienna, Austria. 21

Royal, M. S. (1995). The Analysis and Cognition of Basic Melodic Structures and The Analysis and Cognition of Melodic Complexity. *Music Theory Online*, 1(6). 24

Schlüter, J., Ullrich, K., and Grill, T. (2014). Structural Segmentation with Convolutional Neural Networks MIREX Submission. In *Music Information Retrieval Evaluation eXchange*, Taipei, Taiwan. 190

Schnitzer, D., Flexer, A., Schedl, M., and Widmer, G. (2011). Using Mutual Proximity to Improve Content-Based Audio Similarity. In *Proc of the 12th International Society of Music Information Retrieval*, pages 79–84, Miami, FL, USA. 18

Serrà, J., Müller, M., Grosche, P., and Arcos, J. L. (2012). Unsupervised Detection of Music Boundaries by Time Series Structure Features. In *Proc. of the 26th AAAI Conference on Artificial Intelligence*, number 2009, pages 1613–1619, Toronto, Canada. xiii, 49, 50

Serrà, J., Müller, M., Grosche, P., and Arcos, J. L. (2014). Unsupervised Music Structure Annotation by Time Series Structure Features and Segment Similarity. *IEEE Transactions on Multimedia, Special Issue on Music Data Mining*, 16(5):1229 – 1240. 4, 18, 99, 100, 106, 107, 111, 115, 117, 123, 124, 125, 145, 146, 174

Shao, X., Maddage, N., and Xu, C. (2005). Automatic Music Summarization Based On Music Structure Analysis. In *Proc. of the IEEE International*

Conference on Acoustics Speech and Signal Processing, pages 1169–1172, Philadelphia, PA, USA. IEEE. 22, 58

Shiu, Y., Jeong, H., and Kuo, C. J. (2006). Similarity Matrix Processing for Music Structure Analysis. In *Proc. of the 1st ACM workshop on Audio and Music Computing Multimedia*, pages 69–76, Santa Barbara, CA, USA. 17

Smith, J. B., Burgoyne, J. A., Fujinaga, I., De Roure, D., and Downie, J. S. (2011). Design and Creation of a Large-Scale Database of Structural Annotations. In *Proc. of the 12th International Society of Music Information Retrieval*, pages 555–560, Miami, FL, USA. 41, 99, 111, 128, 134, 141, 146, 175, 189

Smith, J. B. L. (2014). *Explaining Listener Differences in the Perception of Musical Structure*. PhD thesis, Queen Mary, University of London. 5

Smith, J. B. L. and Chew, E. (2013). A Meta-Analysis of the MIREX Structure Segmentation Task. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil. 16, 45, 47, 100, 103, 161

Smith, J. B. L., Chuan, C.-h., and Chew, E. (2013). Audio Properties of Perceived Boundaries in Music. *IEEE Transactions on Multimedia*, (upcoming). 107

Smith, J. O. (2007). *Mathematics of the Discrete Fourier Transform*. W3K Publishing. 117

Smith, J. O. (2010). *Spectral Audio Signal Processing*. W3K Publishing. 30, 123

Sprechmann, P., Bronstein, A. M., and Sapiro, G. (2014). Supervised Non-Euclidean Sparse NMF via Bilevel Optimization with Applications to Speech Enhancement. In *2014 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, pages 11–15, Villers-les-Nancy, France. Ieee. 191

Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A Scale for the Measurement of the Psychological Magnitu Pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190. 34

Temperley, D. (2001). *The Cognition of Basic Musical Structures*. The MIT Press. 27

Tenney, J. and Polansky, L. (1980). Temporal Gestalt Perception in Music. *Journal of Music Theory*, 24(2):205–241. 27

Thom, B., Spevak, C., and Karin, H. (2002). Melodic Segmentation: Evaluating the Performance of Algorithms and Musical Experts. In *Proc. of the 29th International Computer Music Conference*, Göteborg, Sweden. 150

Thurau, C., Kersting, K., and Bauckhage, C. (2009). Convex Non-negative Matrix Factorization in the Wild. In *Proc. of the 9th IEEE International Conference on Data Mining*, pages 523–532, Miami, FL, USA. 93

Tillmann, B. and Bigand, E. (2001). Global Context Effect in Normal and Scrambled Musical Sequences. *Journal of Experimental Psychology: Human Perception and Performance*, 27(5):1185–1196. 25, 170

Turnbull, D., Lanckriet, G., Pampalk, E., and Goto, M. (2007). A Supervised Approach for Detecting Boundaries in Music Using Difference Features and Boosting. In *Proc. of the 5th International Society of Music Information Retrieval*, pages 42–49, Vienna, Austria. 17, 45

Typke, R. (2007). *Music Retrieval Based on Melodic Similarity*. PhD thesis, Utrecht University. 21

Ullrich, K. (2014). *Feed-Forward Neural Networks for Boundary Detection in Music Structure Analysis*. PhD thesis, University of Amsterdam. 190

Ullrich, K., Schlüter, J., and Grill, T. (2014). Boundary Detection in Music Structure Analysis Using Convolutional Neural Networks. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 417–422, Taipei, Taiwan. 18, 190

Urbano, J., Schedl, M., and Serra, X. (2013). Evaluation in Music Information Retrieval. *Journal of Intelligent Information Systems*, 41(3):345–369. 41, 189

Vincent, E., Raczynski, S., and Ono, N. (2010). A Roadmap Rowards Versatile MIR. In *Proc of the 11th International Society of Music Information Retrieval*, number Ismir, pages 662–664, Utrecht, Netherlands. 128

Wang, J.-C., Lee, H.-S., Wang, H.-M., and Jeng, S.-K. (2011). Learning the Similarity of Audio Music in Bag-of-frames Representation from Tagged Music Data. In *Proc. of the 12th International Society of Music Information Retrieval*, pages 85–90, Miami, FL, USA. 18

Weiss, R. and Bello, J. P. (2011). Unsupervised Discovery of Temporal Structure in Music. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1240–1251. 18, 21, 32, 51, 99, 106, 115, 117, 124, 174

West, K., Kumar, A., Shirk, A., Zhu, G., Downie, J. S., Ehmann, A., and Bay, M. (2010). The Networked Environment for Music Analysis (NEMA). *2010 6th World Congress on Services*, pages 314–317. 53

Wiggins, G. A. (2009). Semantic Gap?? Schemantic Schmap!! Methodological Considerations in the Scientific Study of Music. In *Proc of the 11th IEEE International Symposium on Multimedia*, pages 477–482. 5, 144

Zapata, J. R., Davies, M. E. P., and Gómez, E. (2013). Multi-feature Beat Tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):816–825. 36, 91, 123

Zhao, Q., Hautamaki, V., and Fränti, P. (2008). Knee Point Detection in BIC for Detecting the Number. In *Advanced Concepts for Intelligent Vision Systems*, pages 664–673, Nice, France. 110

APPENDIX I

JAMS FILE EXAMPLE

An example of a JAMS file containing music segmentation references is presented here. Two annotators, starting at lines 3 and 199, each with two different layers of segmentation ("function" and "large_scale") are contained in this example*. Note how only one file is needed to include all the annotations (lines 2 to 425) and the metadata (lines 426 to 431) of this song, thus simplifying the organization and management of multiple music segmentation —and potentially other— annotations. These human annotations are included in the SALAMI dataset for the track "I've Got Two Legs" by Monty Python.

```
1  {
2    "segment": [
3      {
4        "annotation_metadata": {
5          "annotation_tools": "Sonic Visualizer",
6          "annotator": {
7            "name": "2",
8            "email": "unknown"
9          },
10         "version": "1.2",
11         "corpus": "SALAMI",
12         "annotation_rules": "http://www.music.mcgill.ca/
13                 ~jordan/salami/SALAMI-Annotator-Guide.pdf",
14         "data_source": "Codaich"
15       },
16       "data": [
17         {
18           "start": {
19             "confidence": 1.0,
20             "value": 0.0
21           },
22           "end": {
23             "confidence": 1.0,
24             "value": 0.255419501
25           },
```

*The "small_scale" levels are not displayed to improve readability.

```
            "label": {
              "confidence": 1.0,
              "secondary_value": "function",
              "value": "Silence"
            }
          },
          {
            "start": {
              "confidence": 1.0,
              "value": 0.255419501
            },
            "end": {
              "confidence": 1.0,
              "value": 12.074376417
            },
            "label": {
              "confidence": 1.0,
              "secondary_value": "function",
              "value": "Intro"
            }
          },
          {
            "start": {
              "confidence": 1.0,
              "value": 12.074376417
            },
            "end": {
              "confidence": 1.0,
              "value": 28.775691609
            },
            "label": {
              "confidence": 1.0,
              "secondary_value": "function",
              "value": "Verse"
            }
          },
          {
            "start": {
              "confidence": 1.0,
              "value": 28.775691609
            },
            "end": {
              "confidence": 1.0,
              "value": 29.303582766
            },
            "label": {
              "confidence": 1.0,
              "secondary_value": "function",
              "value": "no_function"
            }
          },
          {
            "start": {
              "confidence": 1.0,
              "value": 29.303582766
            },
            "end": {
              "confidence": 1.0,
              "value": 33.60632653
            },
            "label": {
              "confidence": 1.0,
              "secondary_value": "function",
              "value": "break"
```

```
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 33.60632653
          },
          "end": {
            "confidence": 1.0,
            "value": 35.02371882
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "function",
            "value": "Silence"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 0.0
          },
          "end": {
            "confidence": 1.0,
            "value": 0.255419501
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "Silence"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 0.255419501
          },
          "end": {
            "confidence": 1.0,
            "value": 12.074376417
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "Z"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 12.074376417
          },
          "end": {
            "confidence": 1.0,
            "value": 28.775691609
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "A"
          }
        },
        {
          "start": {
```

```
            "confidence": 1.0,
            "value": 28.775691609
          },
          "end": {
            "confidence": 1.0,
            "value": 29.303582766
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "A"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 29.303582766
          },
          "end": {
            "confidence": 1.0,
            "value": 33.60632653
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "Z"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 33.60632653
          },
          "end": {
            "confidence": 1.0,
            "value": 35.02371882
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "Silence"
          }
        }
      ]
    },
    {
      "annotation_metadata": {
        "annotation_tools": "Sonic Visualizer",
        "annotator": {
          "name": "8",
          "email": "unknown"
        },
        "version": "1.2",
        "corpus": "SALAMI",
        "annotation_rules": "http://www.music.mcgill.ca/
                ~jordan/salami/SALAMI-Annotator-Guide.pdf",
        "data_source": "Codaich"
      },
      "data": [
        {
          "start": {
            "confidence": 1.0,
            "value": 0.0
          },
```

```
          "end": {
            "confidence": 1.0,
            "value": 0.185759637
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "function",
            "value": "Silence"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 0.185759637
          },
          "end": {
            "confidence": 1.0,
            "value": 8.007664399
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "function",
            "value": "no_function"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 8.007664399
          },
          "end": {
            "confidence": 1.0,
            "value": 12.538820861
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "function",
            "value": "Intro"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 12.538820861
          },
          "end": {
            "confidence": 1.0,
            "value": 28.892517006
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "function",
            "value": "no_function"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 28.892517006
          },
          "end": {
            "confidence": 1.0,
            "value": 29.876077097
          },
```

```
            "label": {
              "confidence": 1.0,
              "secondary_value": "function",
              "value": "Verse"
            }
          },
          {
            "start": {
              "confidence": 1.0,
              "value": 29.876077097
            },
            "end": {
              "confidence": 1.0,
              "value": 32.738820861
            },
            "label": {
              "confidence": 1.0,
              "secondary_value": "function",
              "value": "no_function"
            }
          },
          {
            "start": {
              "confidence": 1.0,
              "value": 32.738820861
            },
            "end": {
              "confidence": 1.0,
              "value": 35.007619047
            },
            "label": {
              "confidence": 1.0,
              "secondary_value": "function",
              "value": "Silence"
            }
          },
          {
            "start": {
              "confidence": 1.0,
              "value": 0.0
            },
            "end": {
              "confidence": 1.0,
              "value": 0.185759637
            },
            "label": {
              "confidence": 1.0,
              "secondary_value": "large_scale",
              "value": "Silence"
            }
          },
          {
            "start": {
              "confidence": 1.0,
              "value": 0.185759637
            },
            "end": {
              "confidence": 1.0,
              "value": 8.007664399
            },
            "label": {
              "confidence": 1.0,
              "secondary_value": "large_scale",
              "value": "Z"
```

```
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 8.007664399
          },
          "end": {
            "confidence": 1.0,
            "value": 12.538820861
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "I"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 12.538820861
          },
          "end": {
            "confidence": 1.0,
            "value": 28.892517006
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "A"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 28.892517006
          },
          "end": {
            "confidence": 1.0,
            "value": 29.876077097
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "A"
          }
        },
        {
          "start": {
            "confidence": 1.0,
            "value": 29.876077097
          },
          "end": {
            "confidence": 1.0,
            "value": 32.738820861
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "Z"
          }
        },
        {
          "start": {
```

```
            "confidence": 1.0,
            "value": 32.738820861
          },
          "end": {
            "confidence": 1.0,
            "value": 35.007619047
          },
          "label": {
            "confidence": 1.0,
            "secondary_value": "large_scale",
            "value": "Silence"
          }
        }
      ]
    }
  ],
  "file_metadata": {
    "duration": 35.0,
    "title": "I__ve_Got_Two_Legs",
    "jams_version": "0.0.1",
    "artist": "Monty_Python"
  }
}
```

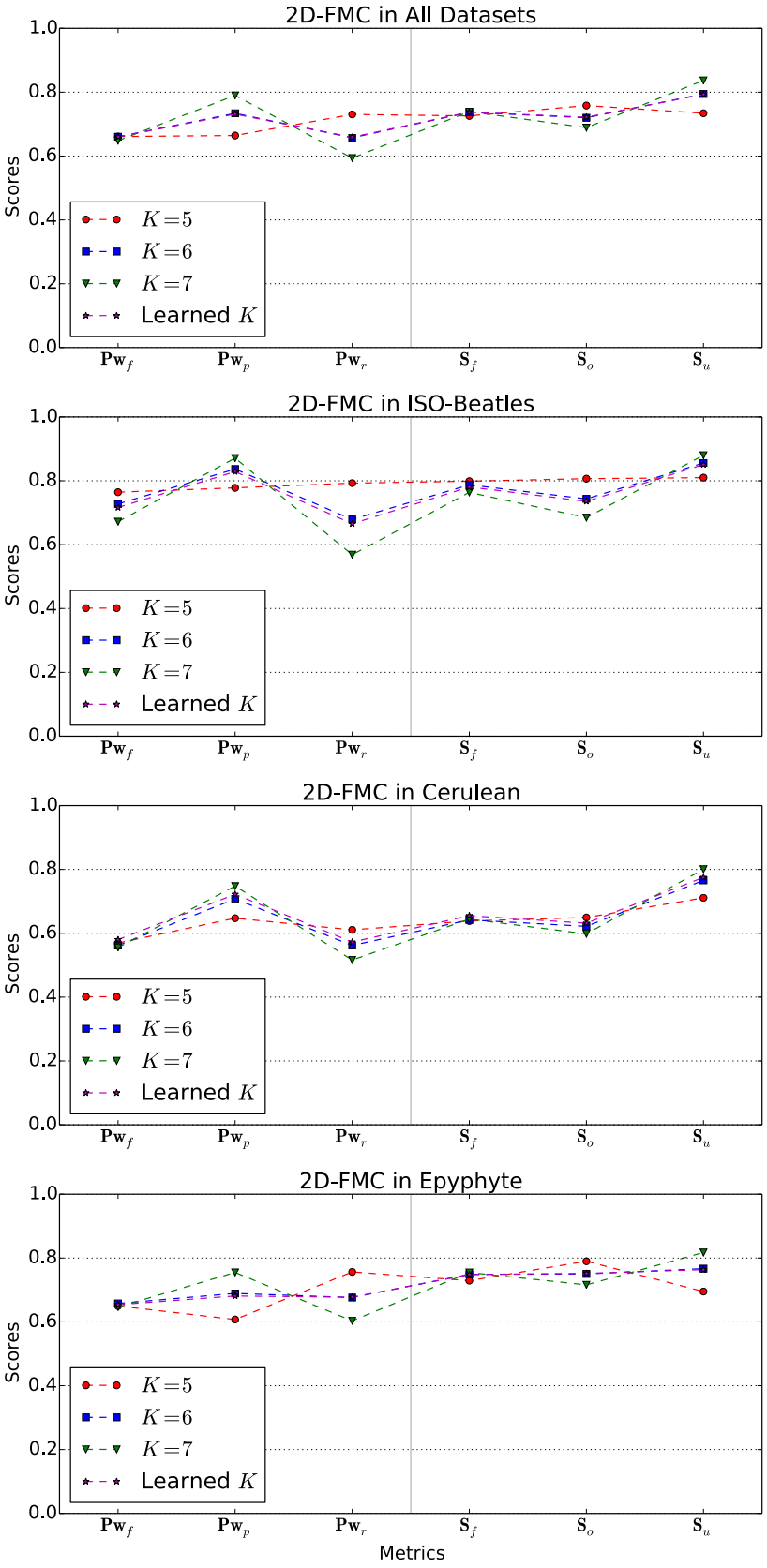
APPENDIX II

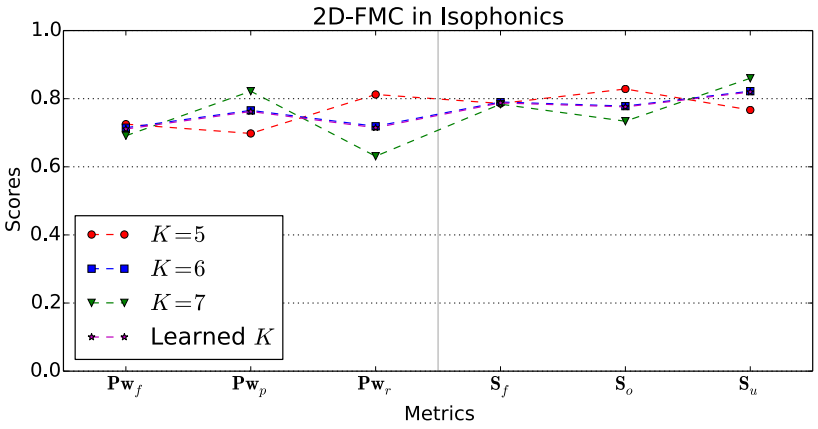
MSAF RESULTS

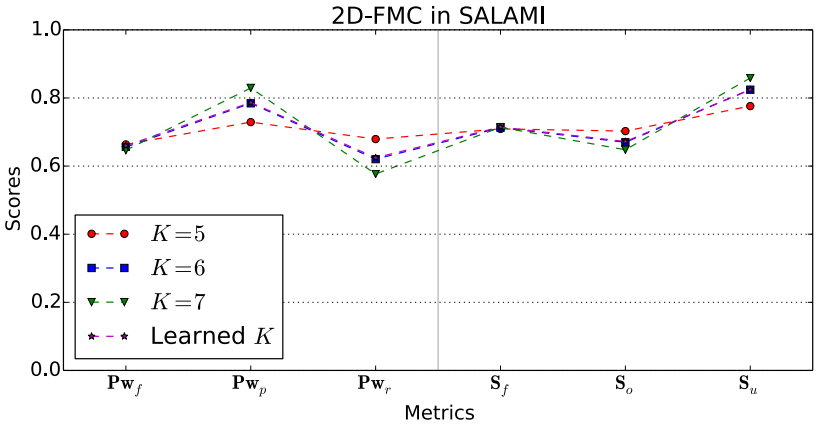
In this appendix the results of MSAF over the datasets presented in this work are displayed. For each algorithm, different features are used as input (when possible) to assess the impact of the type of musical aspect being considered. The metrics are the standard ones used for this task, as described in Chapter II. The structural metrics (i.e., pairwise frame clustering and normalized conditional entropies) have been computed using the human annotated boundaries, in order to avoid the bias that automatically estimated boundaries might introduce. The plots are sorted by algorithm name and further sub-grouped by dataset name. Individual results for each file used to produce these plot are publicly available*.

0.1 2D Fourier Magnitude Coefficients Method

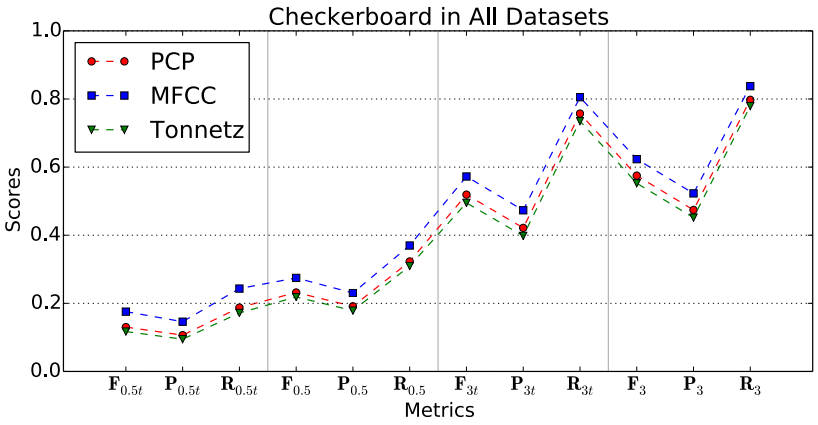
*https://github.com/urinieto/msaf/blob/master/results/results-141207-NietoDissertation.zip

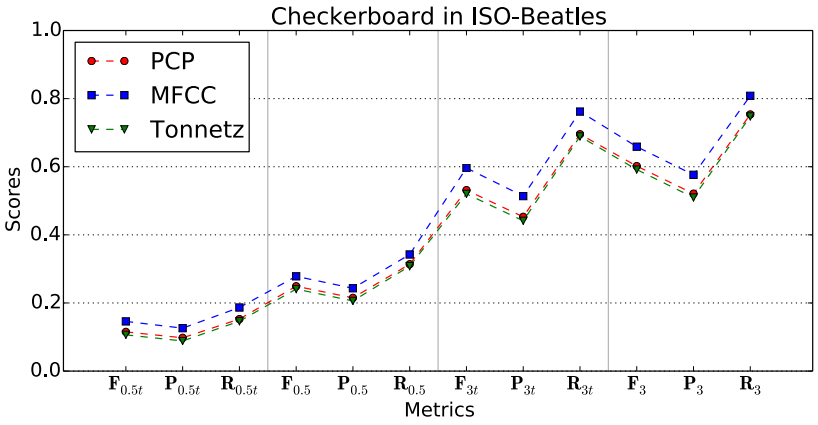

2D-FMC in All Datasets
2D-FMC in ISO-Beatles
2D-FMC in Cerulean
2D-FMC in Epyphyte
Scores
Metrics
1.0
0.8
0.6
0.4
0.2
0.0
$K=5$
$K=6$
$K=7$
Learned K
$\mathbf{Pw}_f$
$\mathbf{Pw}_p$
$\mathbf{Pw}_r$
$\mathbf{S}_f$
$\mathbf{S}_o$
$\mathbf{S}_u$

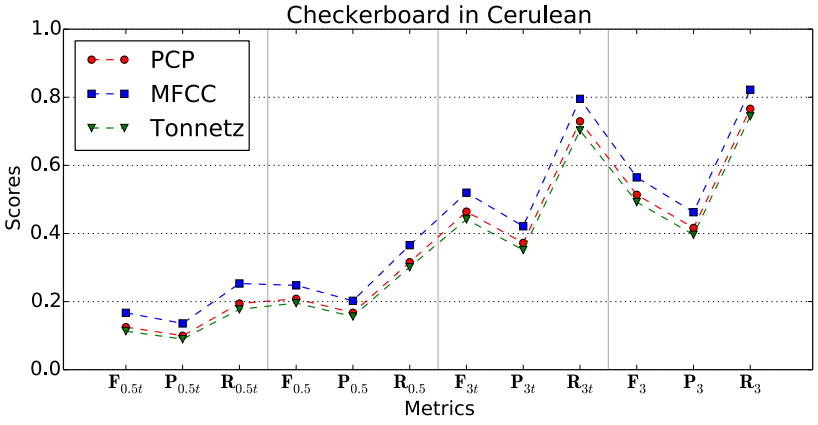

2D-FMC in Isophonics
Scores
Metrics
1.0
0.8
0.6
0.4
0.2
0.0
$K=5$
$K=6$
$K=7$
Learned K
$\mathbf{Pw}_f$
$\mathbf{Pw}_p$
$\mathbf{Pw}_r$
$\mathbf{S}_f$
$\mathbf{S}_o$
$\mathbf{S}_u$

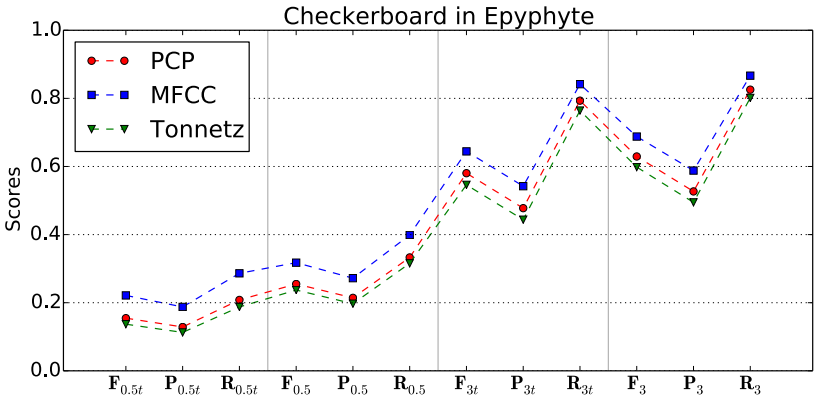

2D-FMC in SALAMI
Scores
Metrics
1.0
0.8
0.6
0.4
0.2
0.0
$K=5$
$K=6$
$K=7$
Learned K
$\mathbf{Pw}_f$
$\mathbf{Pw}_p$
$\mathbf{Pw}_r$
$\mathbf{S}_f$
$\mathbf{S}_o$
$\mathbf{S}_u$

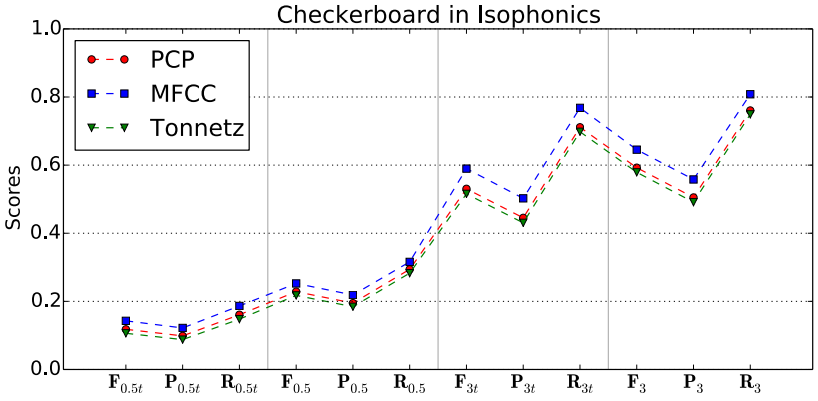
0.2 Checkerboard Method

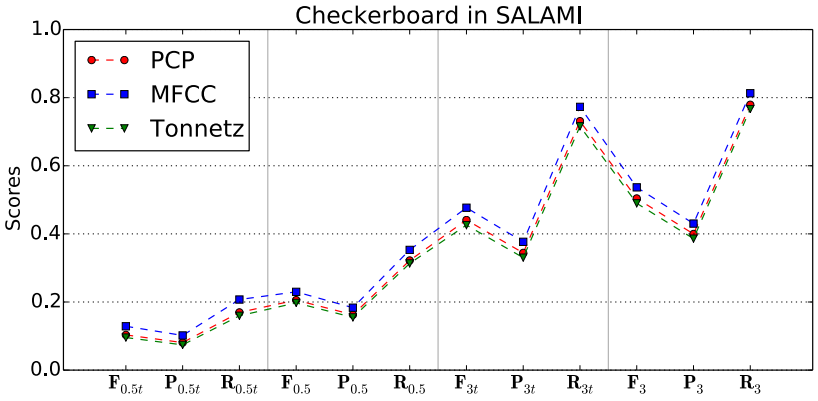


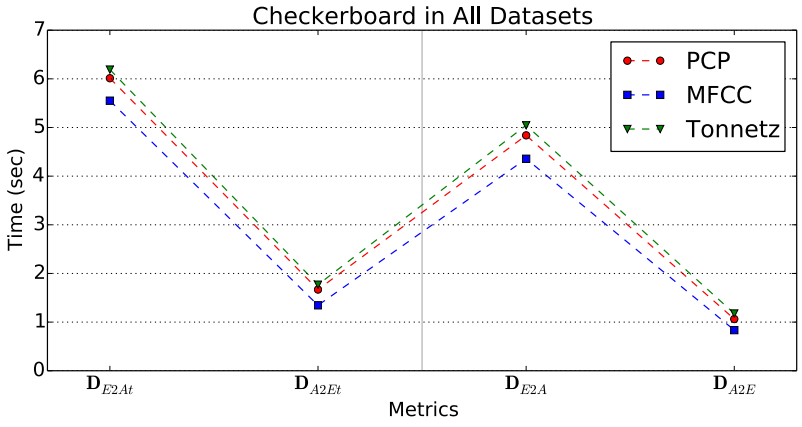


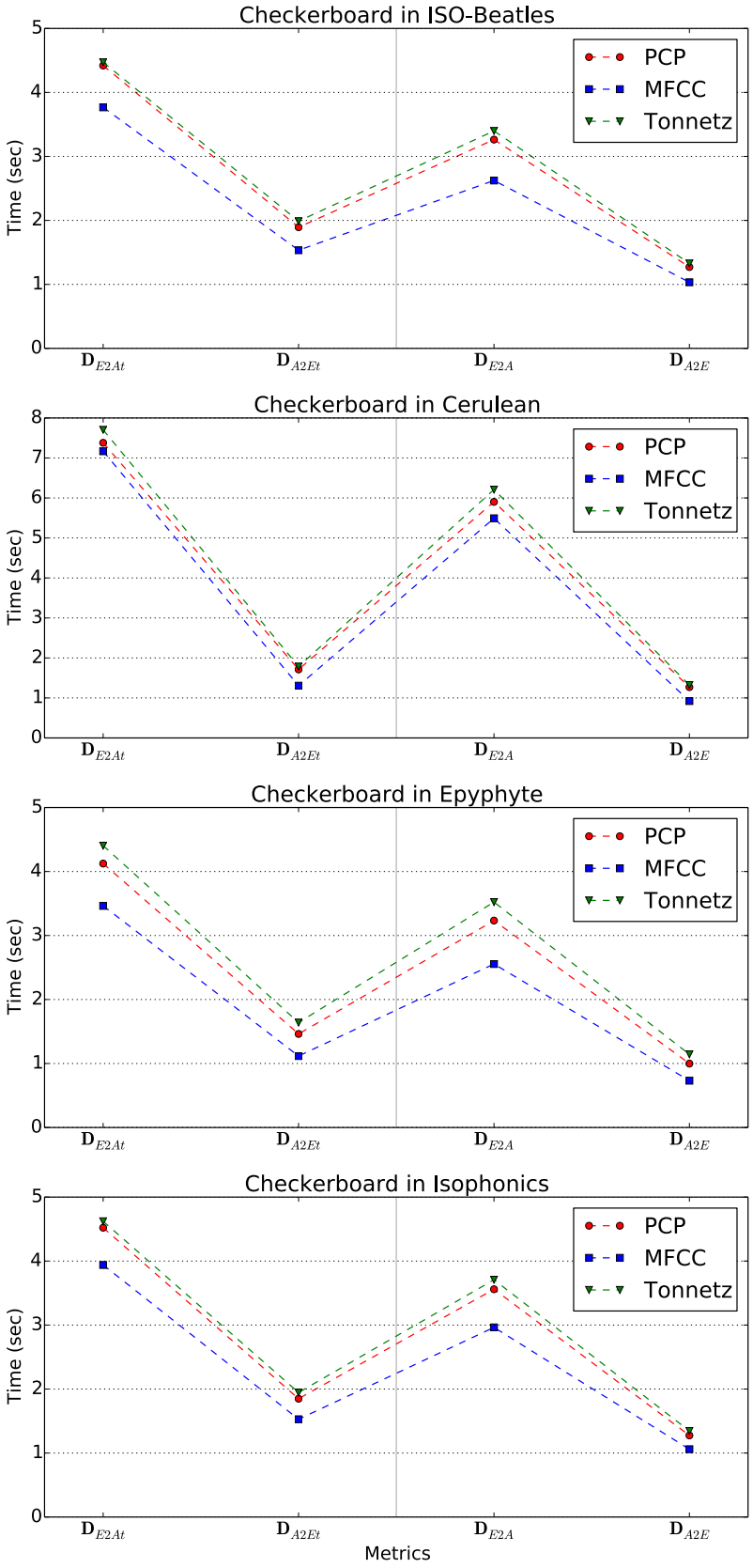


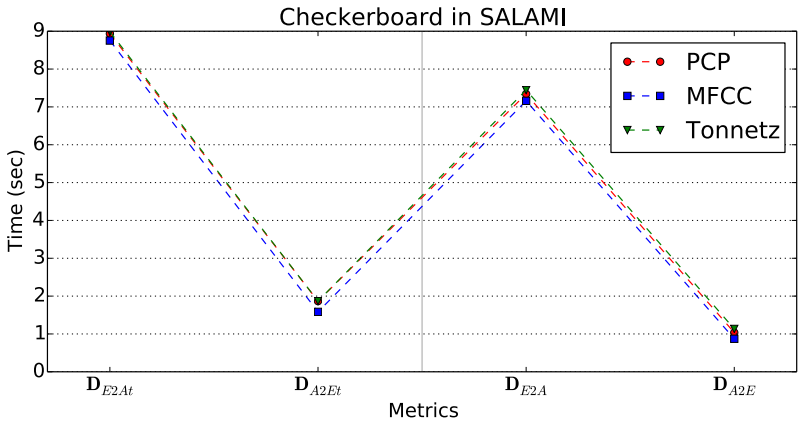

Checkerboard in Epyphyte
PCP
MFCC
Tonnetz
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$


Checkerboard in Isophonics
PCP
MFCC
Tonnetz
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$

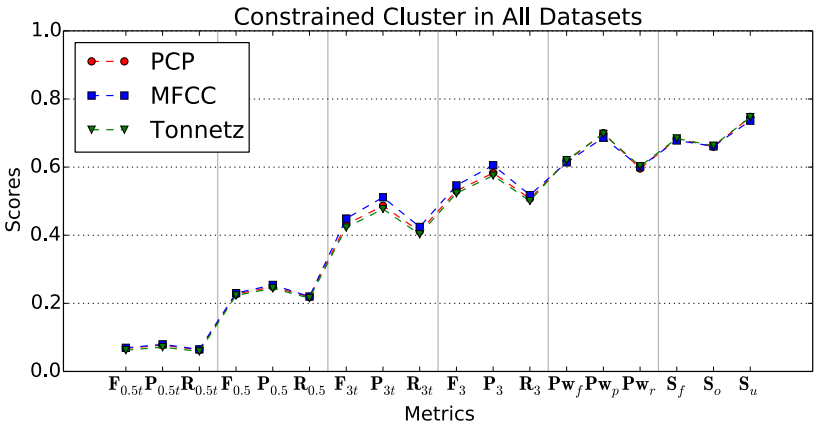

Checkerboard in SALAMI
PCP
MFCC
Tonnetz
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$

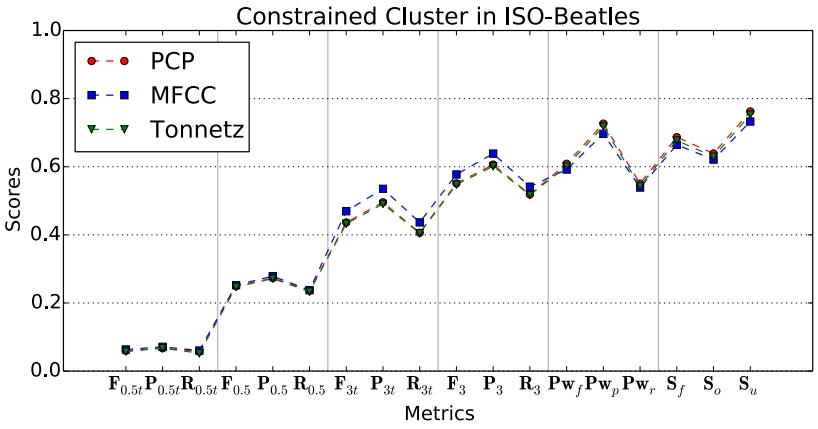

Checkerboard in All Datasets
PCP
MFCC
Tonnetz
Time (sec)
7
6
5
4
3
2
1
0
$\mathbf{D}_{E2At}$ $\mathbf{D}_{A2Et}$ $\mathbf{D}_{E2A}$ $\mathbf{D}_{A2E}$
Metrics

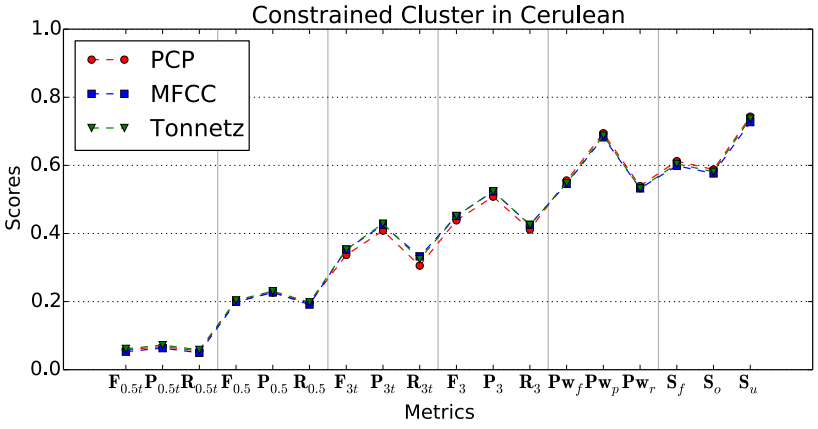

Checkerboard in ISO-Beatles
Checkerboard in Cerulean
Checkerboard in Epyphyte
Checkerboard in Isophonics
PCP
MFCC
Tonnetz
Time (sec)
Metrics
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$

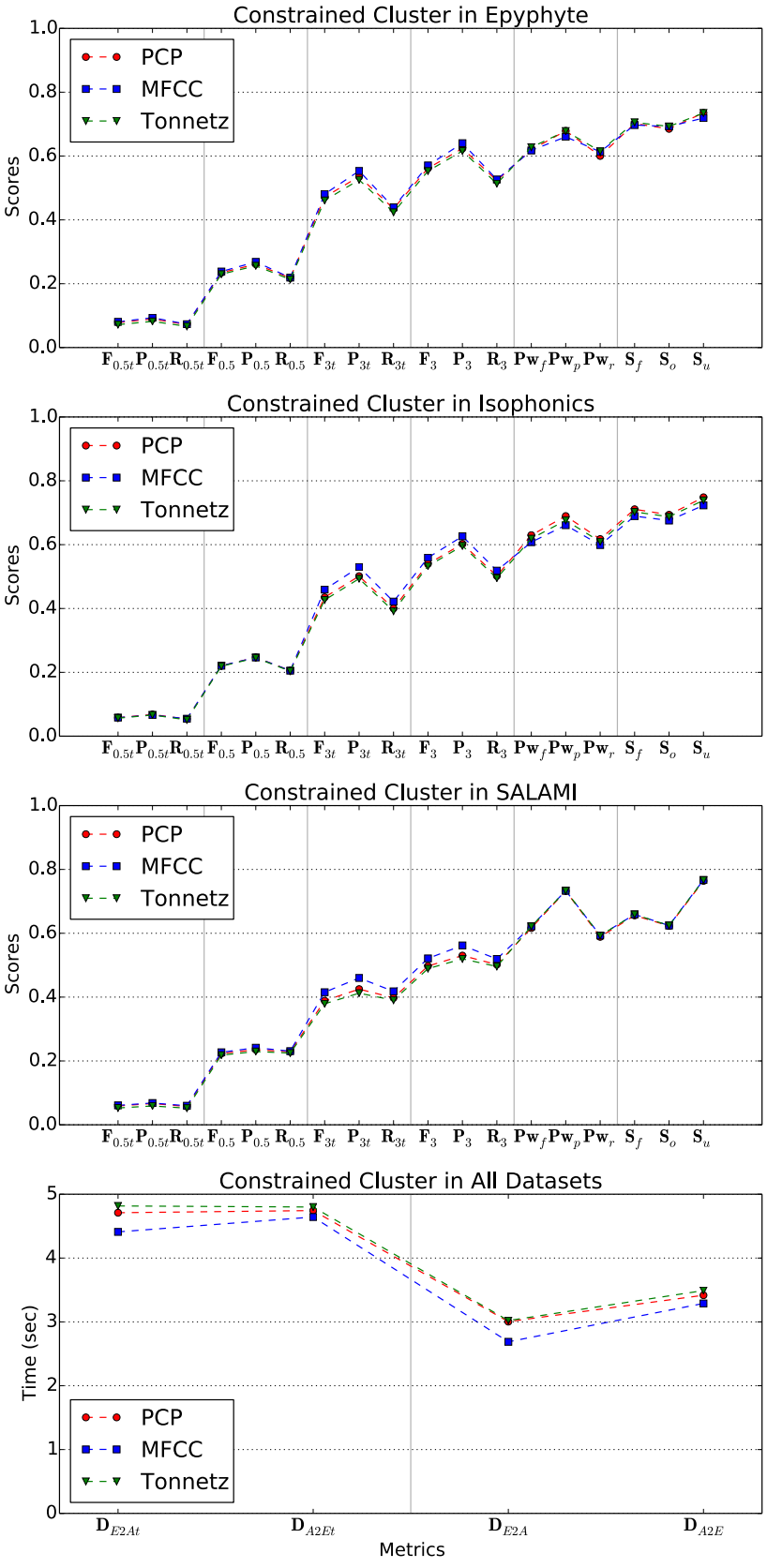

Checkerboard in SALAMI
PCP
MFCC
Tonnetz
Time (sec)
0
1
2
3
4
5
6
7
8
9
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Metrics

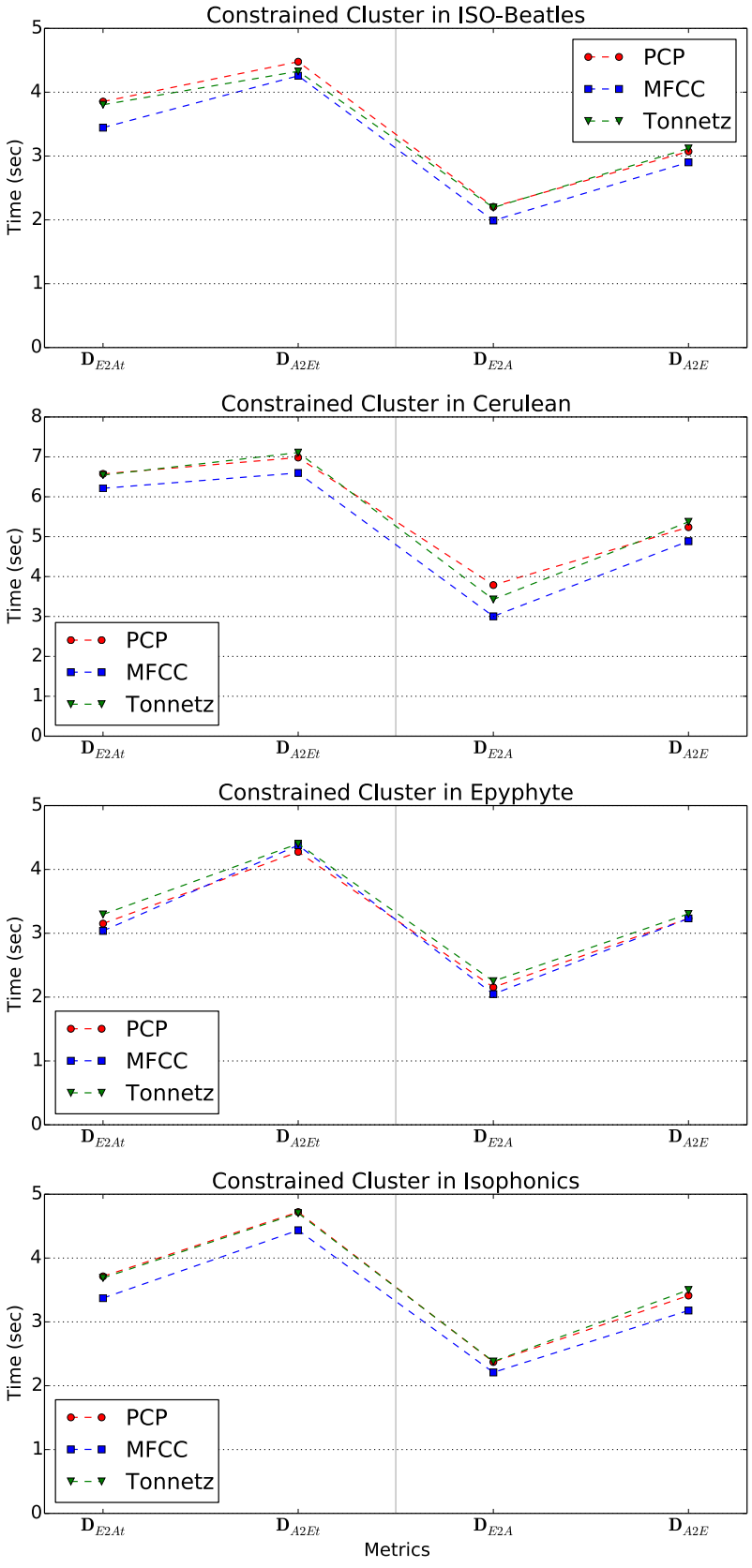
0.3 Constrained Cluster

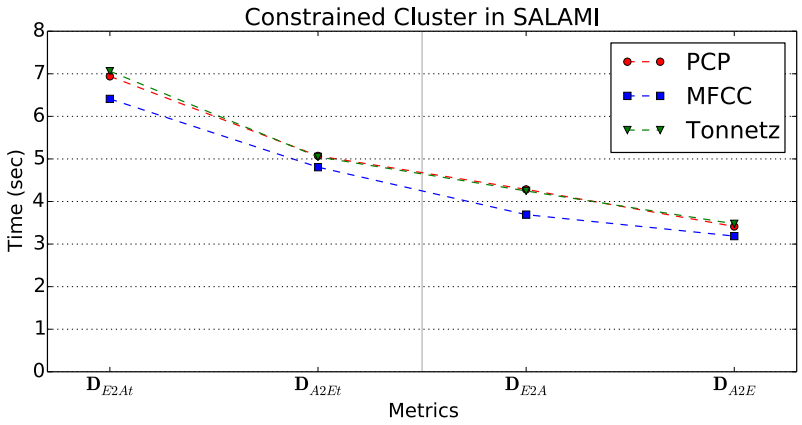




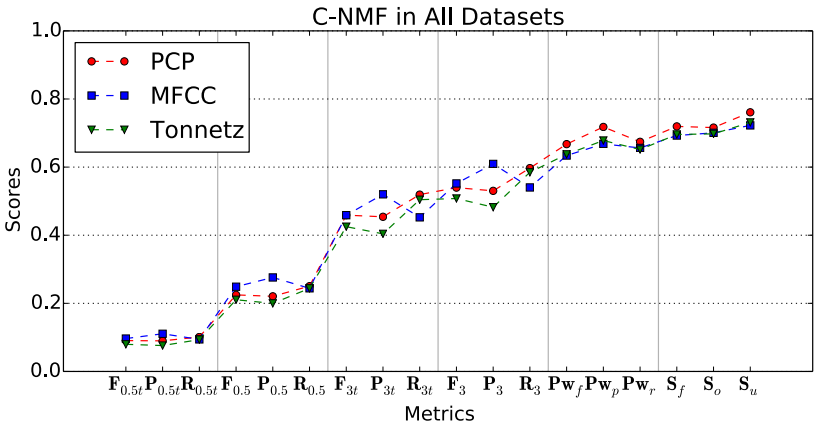


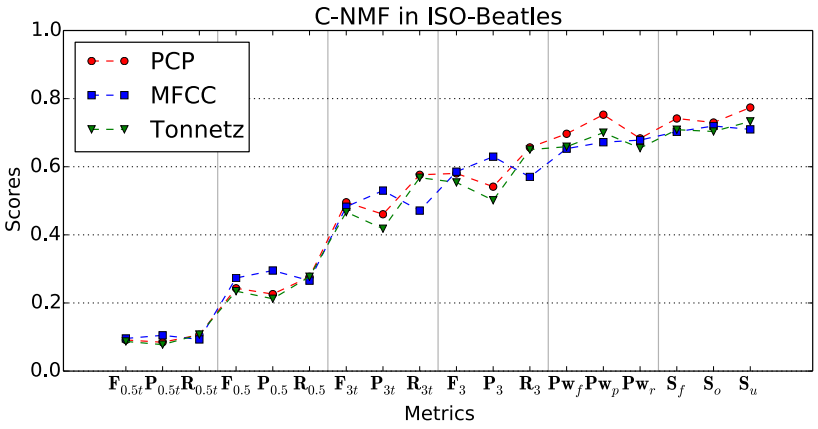

Constrained Cluster in Epyphyte
Constrained Cluster in Isophonics
Constrained Cluster in SALAMI
Constrained Cluster in All Datasets
PCP
MFCC
Tonnetz
Scores
Time (sec)
Metrics
1.0
0.8
0.6
0.4
0.2
0.0
5
4
3
2
1
0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$ $\mathbf{Pw}_{f}$ $\mathbf{Pw}_{p}$ $\mathbf{Pw}_{r}$ $\mathbf{S}_{f}$ $\mathbf{S}_{o}$ $\mathbf{S}_{u}$
$\mathbf{D}_{E2At}$ $\mathbf{D}_{A2Et}$ $\mathbf{D}_{E2A}$ $\mathbf{D}_{A2E}$

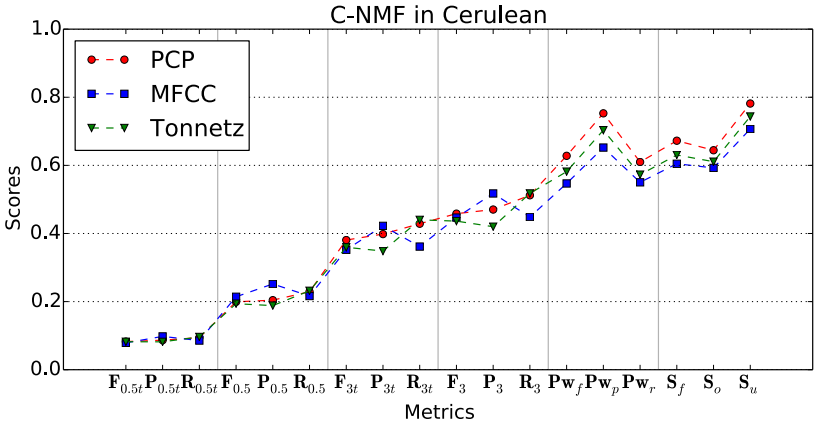

Constrained Cluster in ISO-Beatles
Time (sec)
PCP
MFCC
Tonnetz
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Constrained Cluster in Cerulean
Time (sec)
PCP
MFCC
Tonnetz
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Constrained Cluster in Epyphyte
Time (sec)
PCP
MFCC
Tonnetz
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Constrained Cluster in Isophonics
Time (sec)
PCP
MFCC
Tonnetz
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Metrics

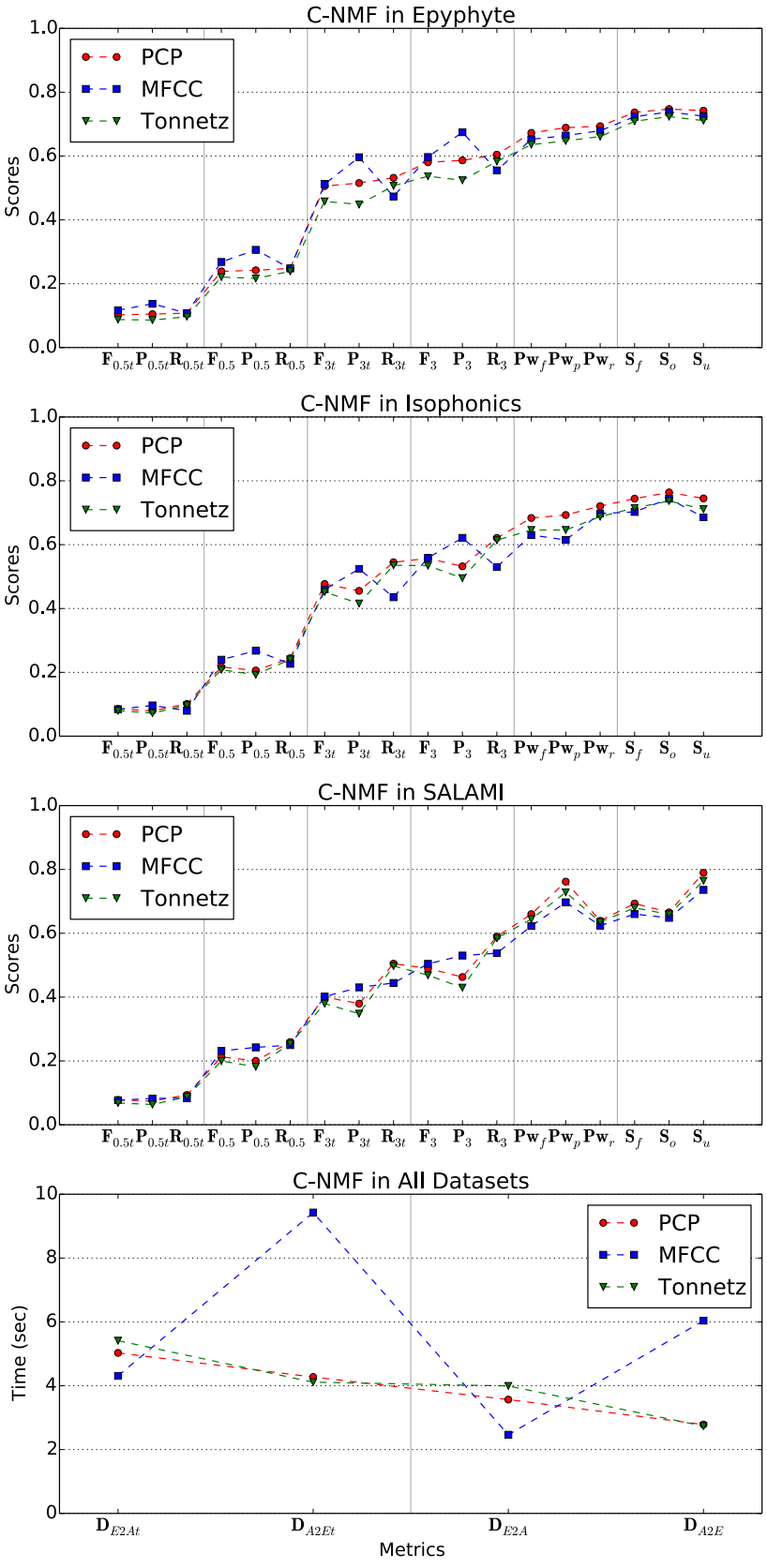

Constrained Cluster in SALAMI
PCP
MFCC
Tonnetz
Time (sec)
0
1
2
3
4
5
6
7
8
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Metrics

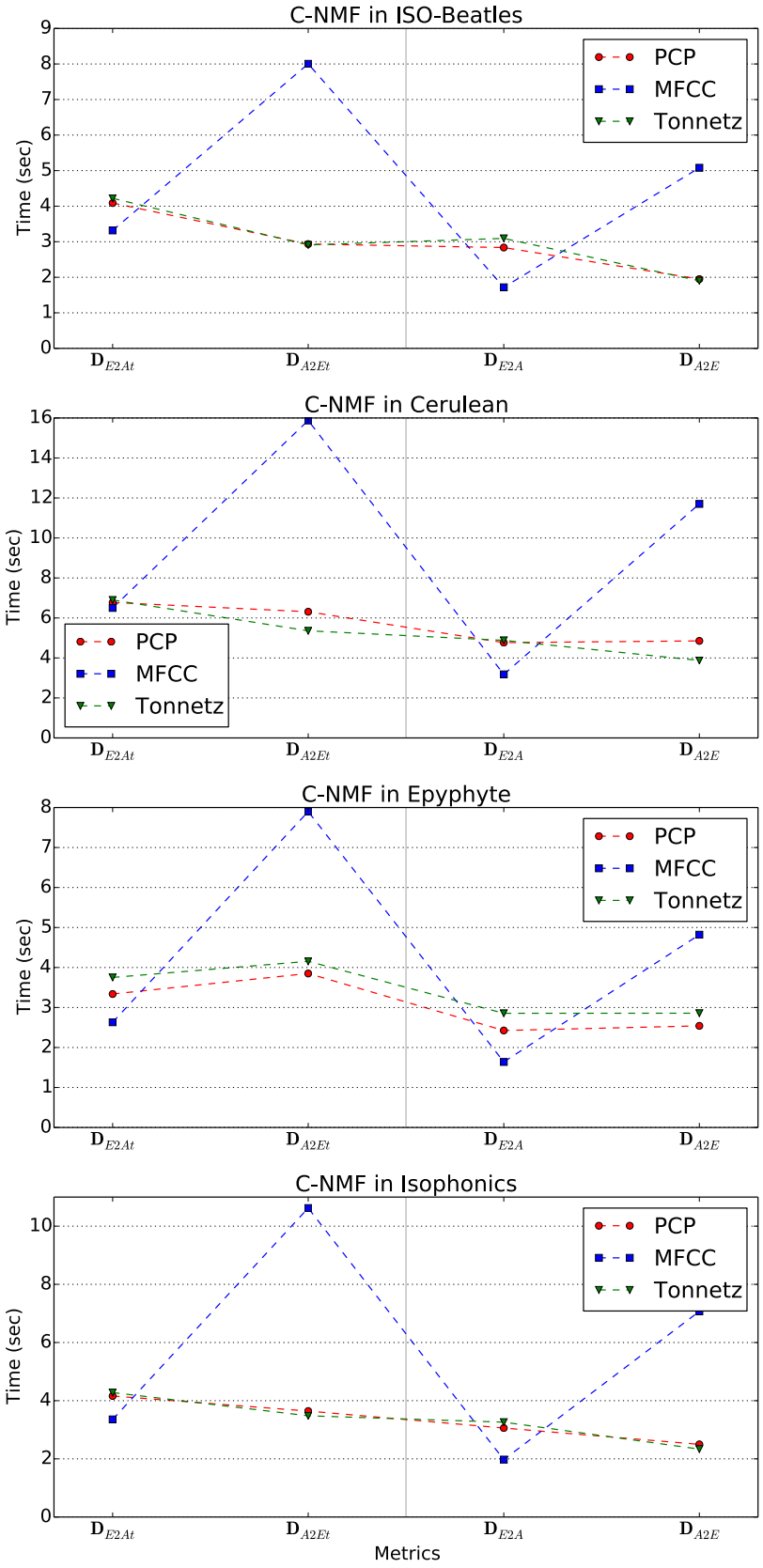
0.4 Convex NMF

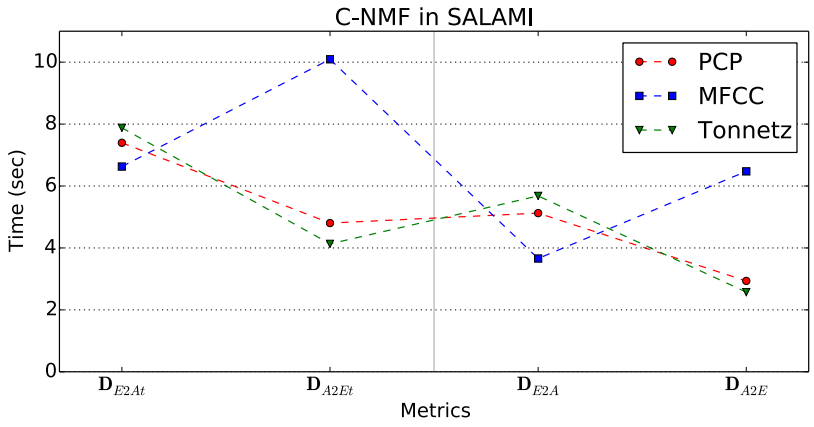






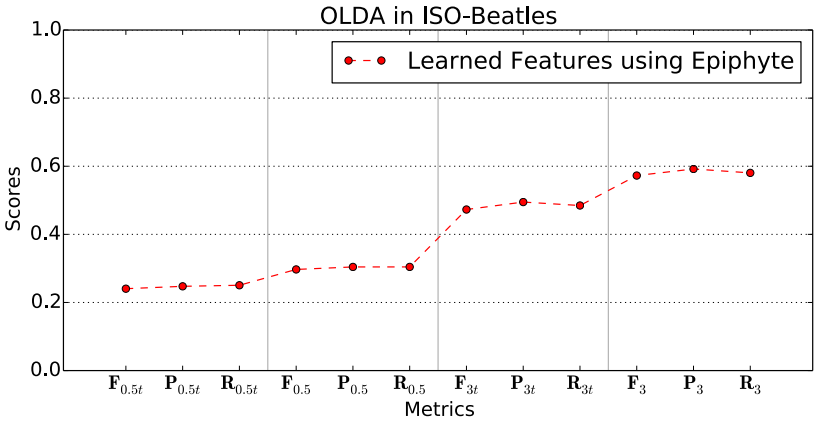

C-NMF in Epyphyte
C-NMF in Isophonics
C-NMF in SALAMI
C-NMF in All Datasets
PCP
MFCC
Tonnetz
Scores
Time (sec)
Metrics
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$ $\mathbf{Pw}_{f}$ $\mathbf{Pw}_{p}$ $\mathbf{Pw}_{r}$ $\mathbf{S}_{f}$ $\mathbf{S}_{o}$ $\mathbf{S}_{u}$
$\mathbf{D}_{E2At}$ $\mathbf{D}_{A2Et}$ $\mathbf{D}_{E2A}$ $\mathbf{D}_{A2E}$

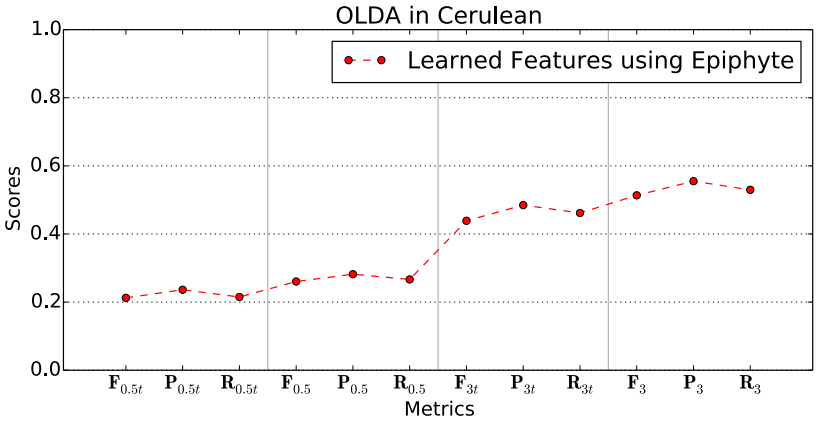

C-NMF in ISO-Beatles
C-NMF in Cerulean
C-NMF in Epyphyte
C-NMF in Isophonics
PCP
MFCC
Tonnetz
Time (sec)
Metrics
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$

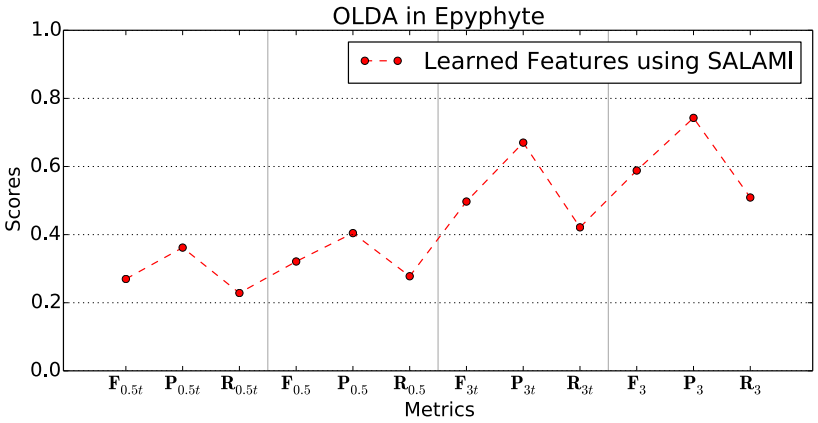

C-NMF in SALAMI
PCP
MFCC
Tonnetz
Time (sec)
0
2
4
6
8
10
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Metrics

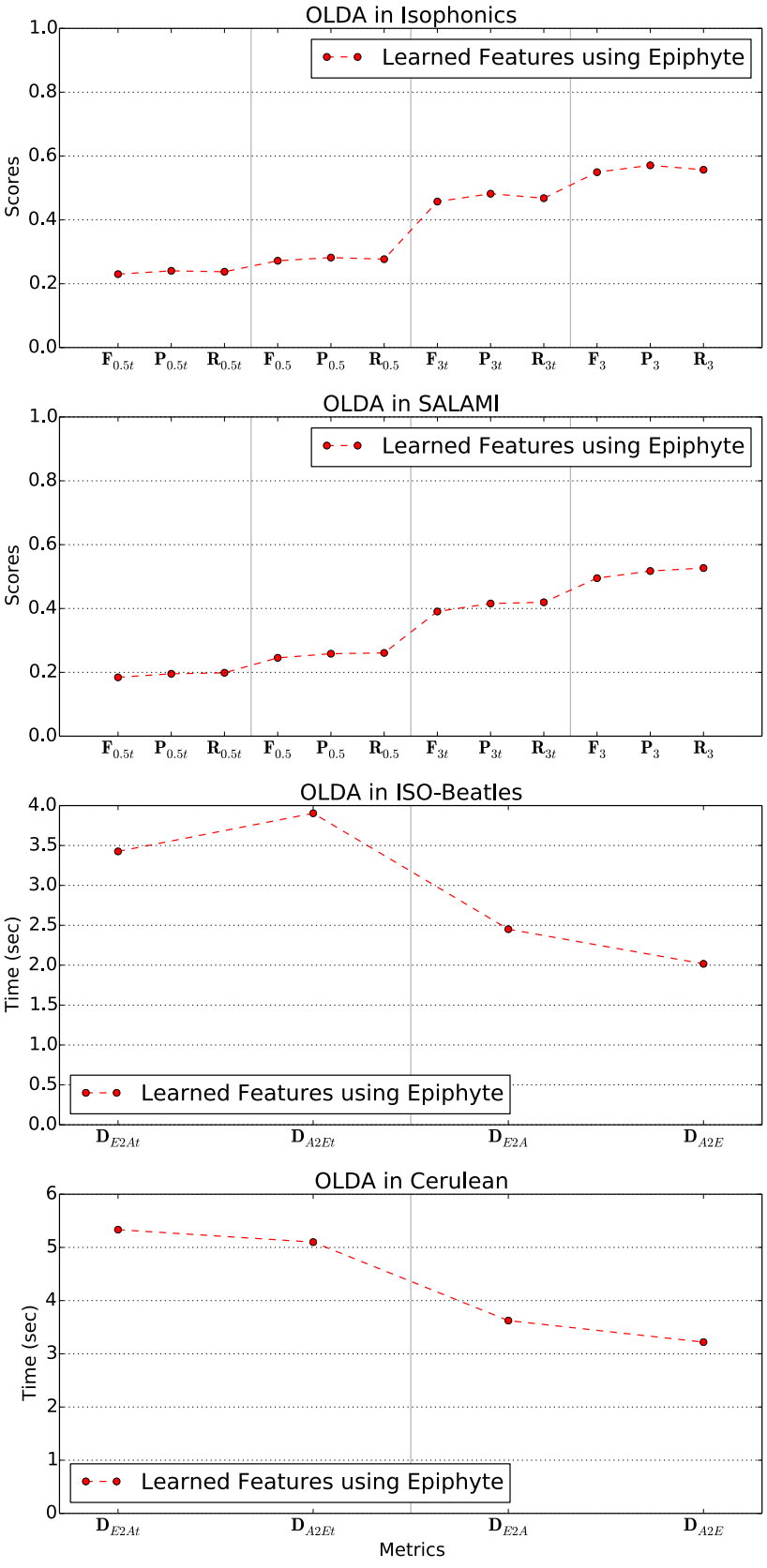
0.5 Ordinal Linear Discriminant Analysis

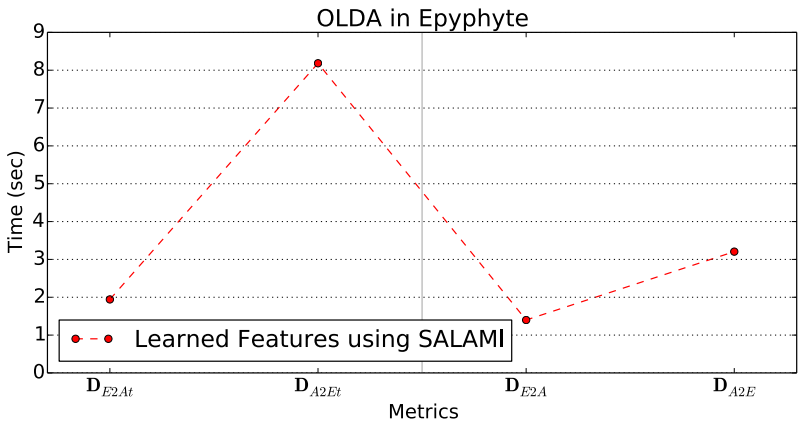
The model used for all datasets except for Epiphyte was trained on the whole Epiphyte dataset. For the Epiphyte dataset, the model used was trained on the entire SALAMI dataset.

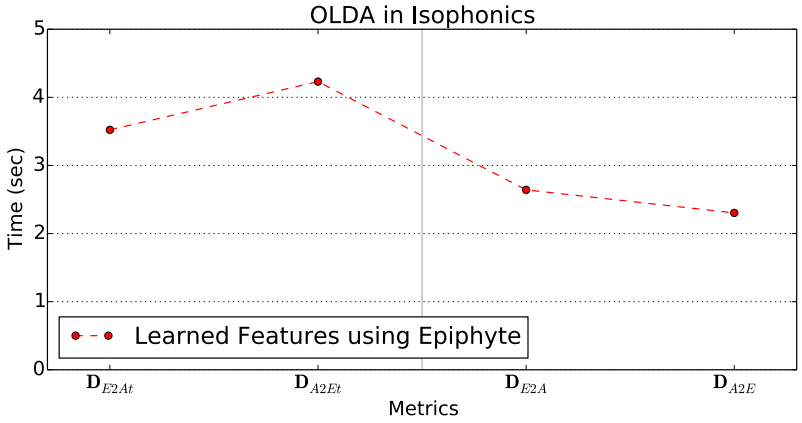


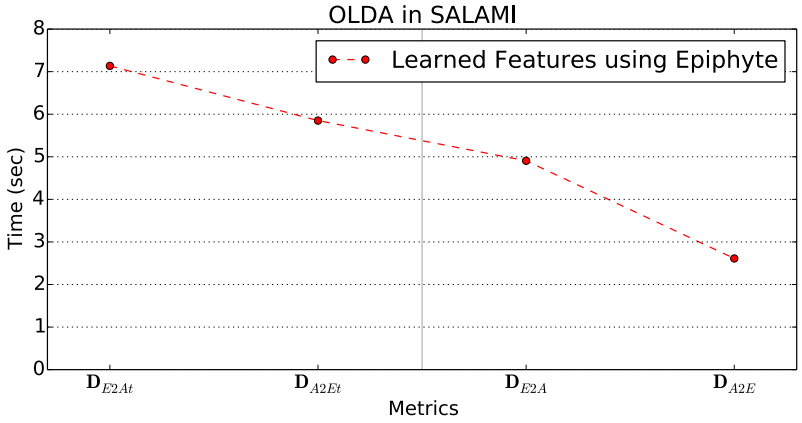


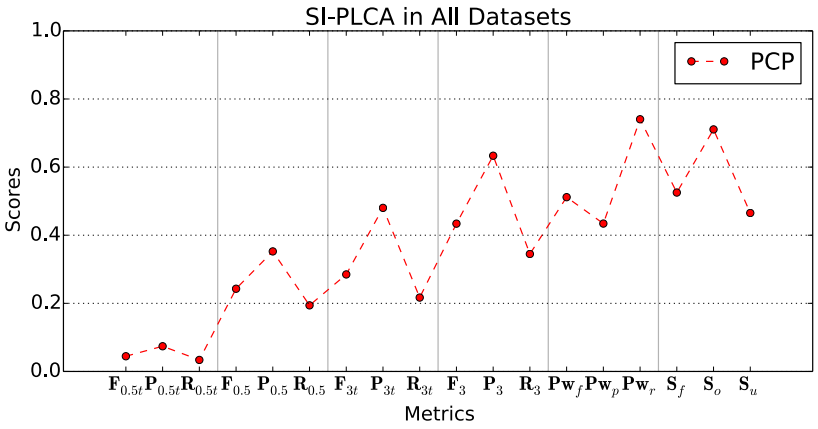


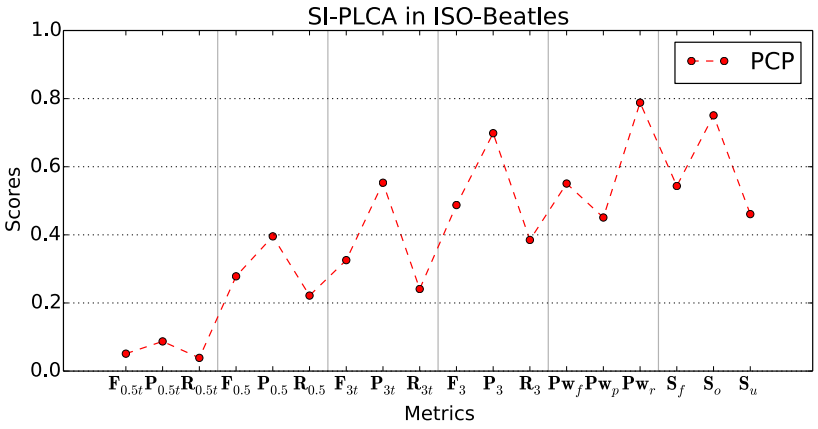

OLDA in Isophonics
Learned Features using Epiphyte
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$
OLDA in SALAMI
Learned Features using Epiphyte
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$
OLDA in ISO-Beatles
Time (sec)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Learned Features using Epiphyte
$\mathbf{D}_{E2At}$ $\mathbf{D}_{A2Et}$ $\mathbf{D}_{E2A}$ $\mathbf{D}_{A2E}$
OLDA in Cerulean
Time (sec)
6
5
4
3
2
1
0
Learned Features using Epiphyte
$\mathbf{D}_{E2At}$ $\mathbf{D}_{A2Et}$ $\mathbf{D}_{E2A}$ $\mathbf{D}_{A2E}$
Metrics

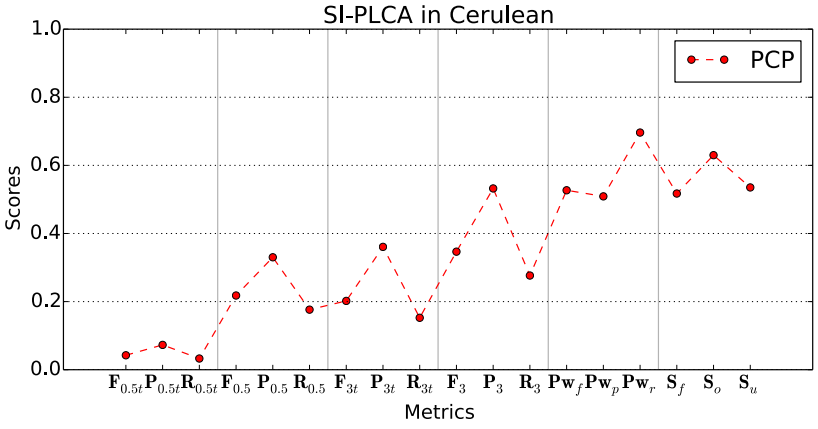

OLDA in Epyphyte
Time (sec)
Learned Features using SALAMI
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Metrics

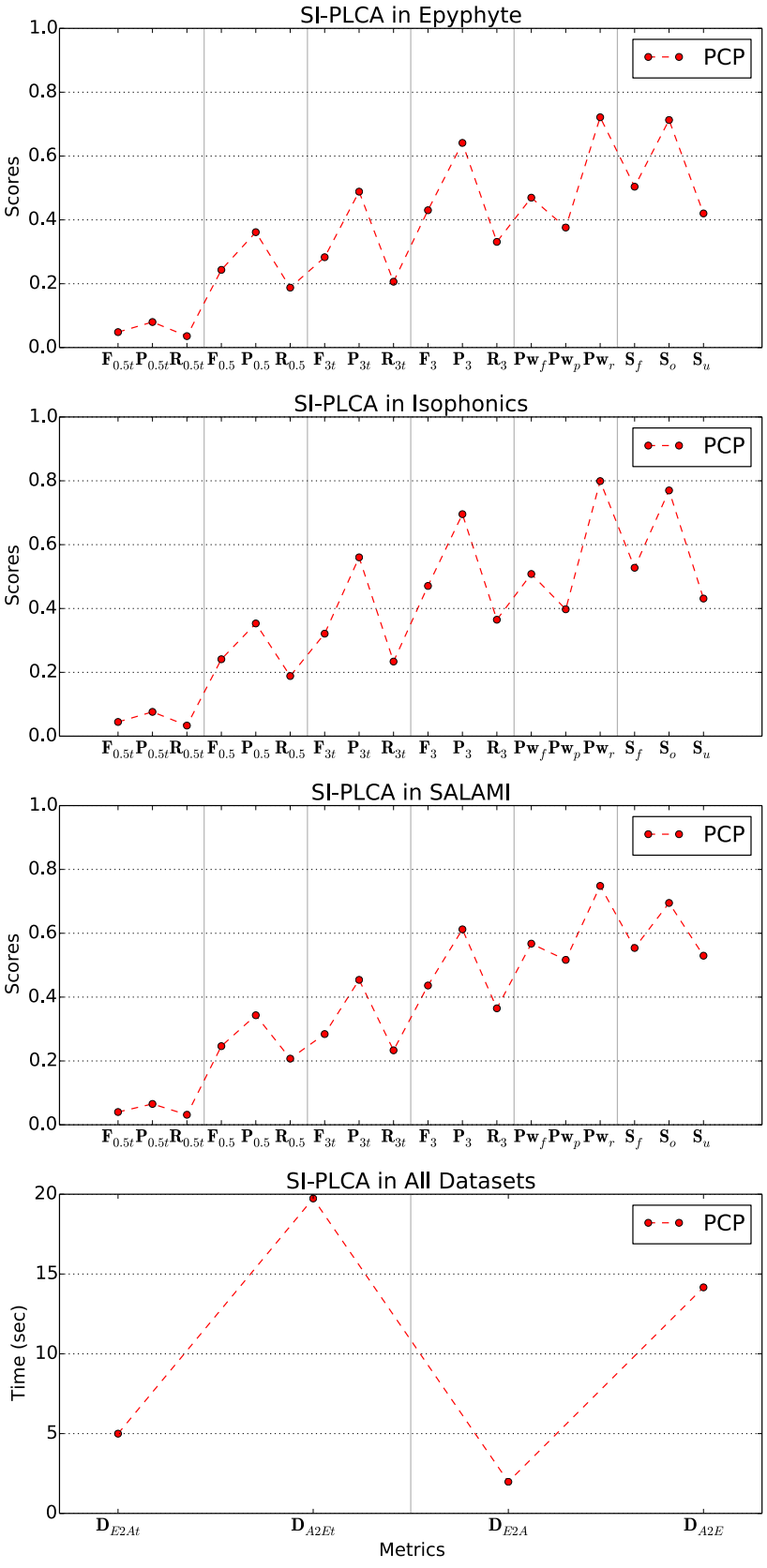

OLDA in Isophonics
Time (sec)
Learned Features using Epiphyte
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Metrics

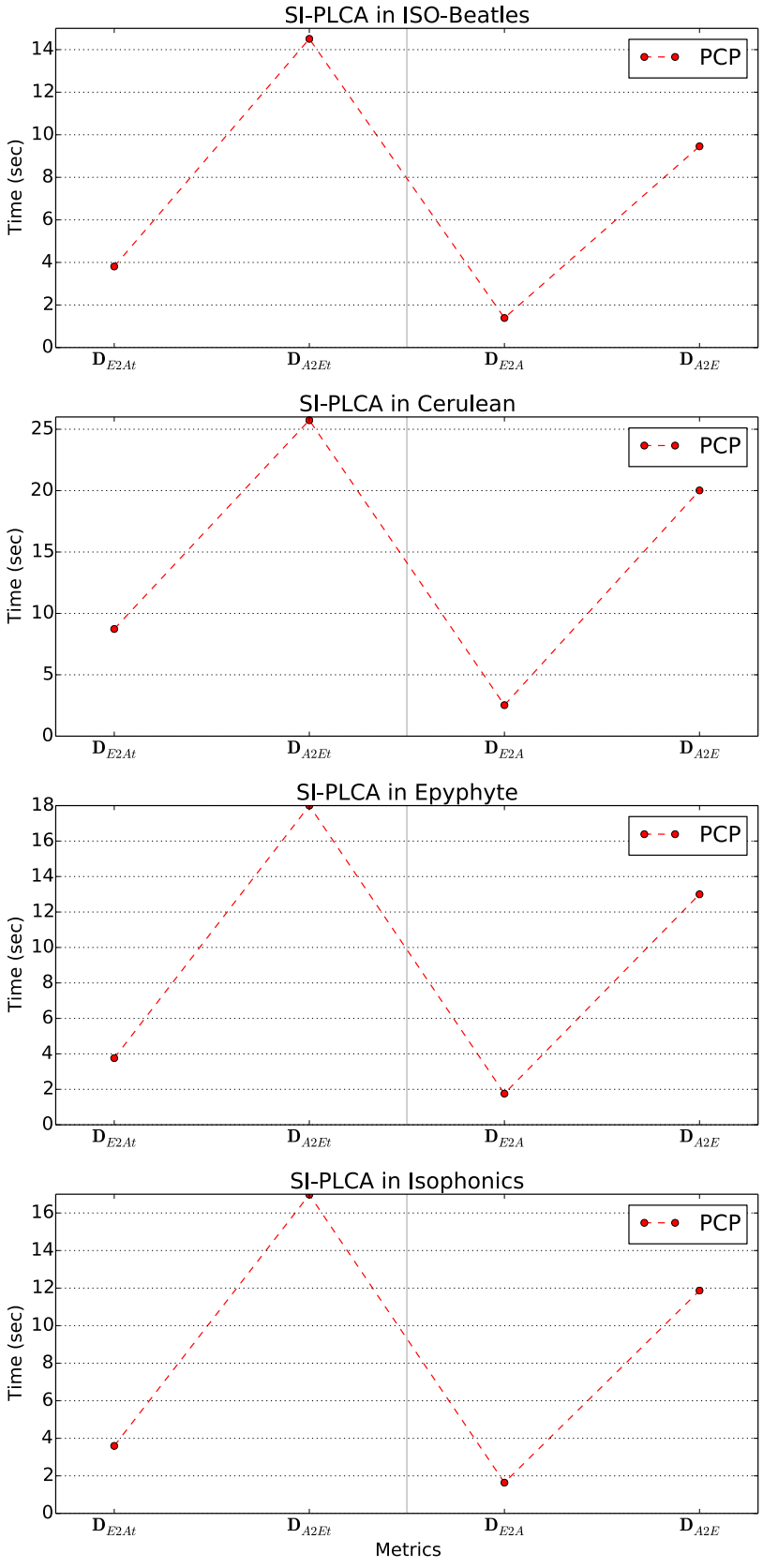

OLDA in SALAMI
Time (sec)
Learned Features using Epiphyte
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Metrics

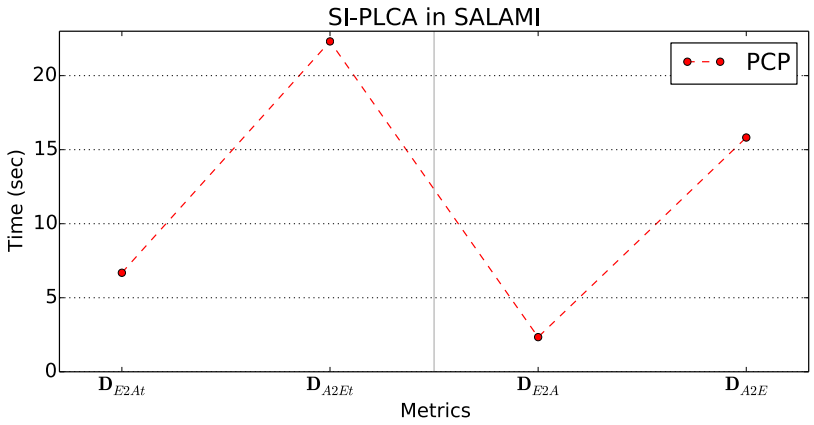

SI-PLCA in All Datasets
PCP
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$ $\mathbf{Pw}_{f}$ $\mathbf{Pw}_{p}$ $\mathbf{Pw}_{r}$ $\mathbf{S}_{f}$ $\mathbf{S}_{o}$ $\mathbf{S}_{u}$
Metrics


SI-PLCA in ISO-Beatles
PCP
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$ $\mathbf{Pw}_{f}$ $\mathbf{Pw}_{p}$ $\mathbf{Pw}_{r}$ $\mathbf{S}_{f}$ $\mathbf{S}_{o}$ $\mathbf{S}_{u}$
Metrics

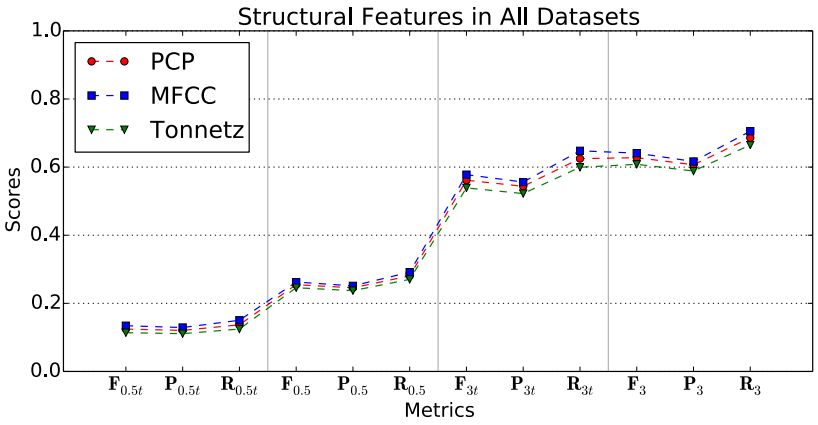

SI-PLCA in Cerulean
PCP
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$ $\mathbf{Pw}_{f}$ $\mathbf{Pw}_{p}$ $\mathbf{Pw}_{r}$ $\mathbf{S}_{f}$ $\mathbf{S}_{o}$ $\mathbf{S}_{u}$
Metrics

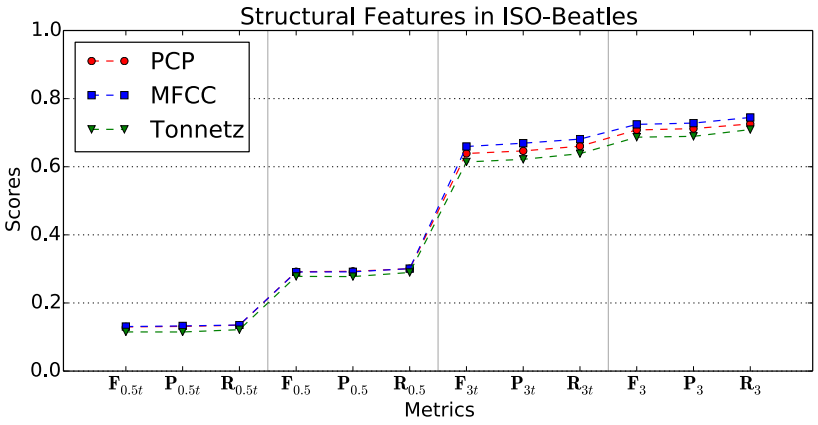

SI-PLCA in Epyphyte
SI-PLCA in Isophonics
SI-PLCA in SALAMI
SI-PLCA in All Datasets
PCP
Scores
Time (sec)
Metrics
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$ $\mathbf{Pw}_f$ $\mathbf{Pw}_p$ $\mathbf{Pw}_r$ $\mathbf{S}_f$ $\mathbf{S}_o$ $\mathbf{S}_u$
$\mathbf{D}_{E2At}$ $\mathbf{D}_{A2Et}$ $\mathbf{D}_{E2A}$ $\mathbf{D}_{A2E}$

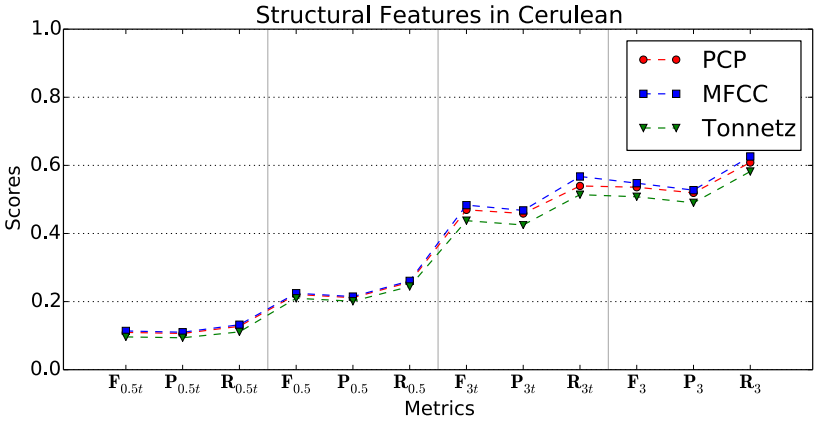

SI-PLCA in ISO-Beatles
PCP
Time (sec)
0
2
4
6
8
10
12
14
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
SI-PLCA in Cerulean
PCP
Time (sec)
0
5
10
15
20
25
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
SI-PLCA in Epyphyte
PCP
Time (sec)
0
2
4
6
8
10
12
14
16
18
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
SI-PLCA in Isophonics
PCP
Time (sec)
0
2
4
6
8
10
12
14
16
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Metrics

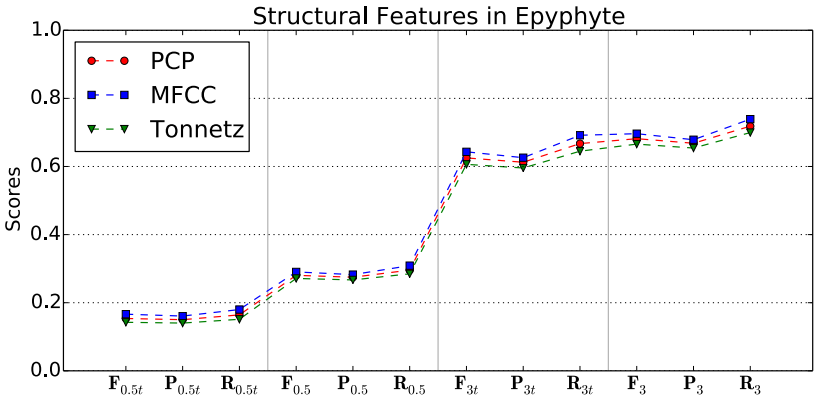

SI-PLCA in SALAMI
PCP
Time (sec)
0
5
10
15
20
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Metrics

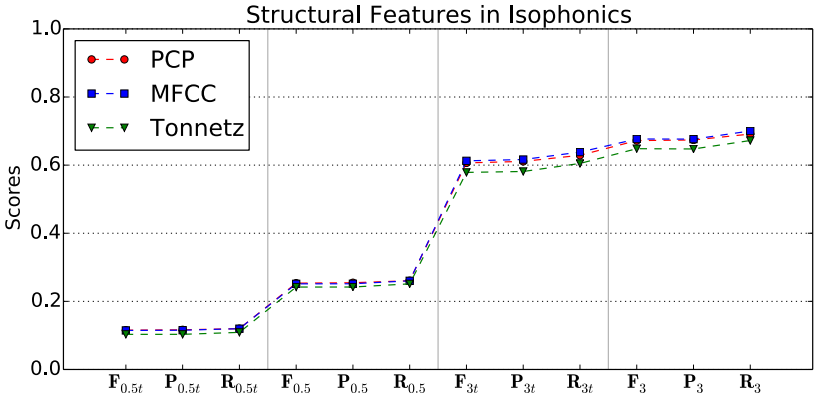
0.7 Structural Features

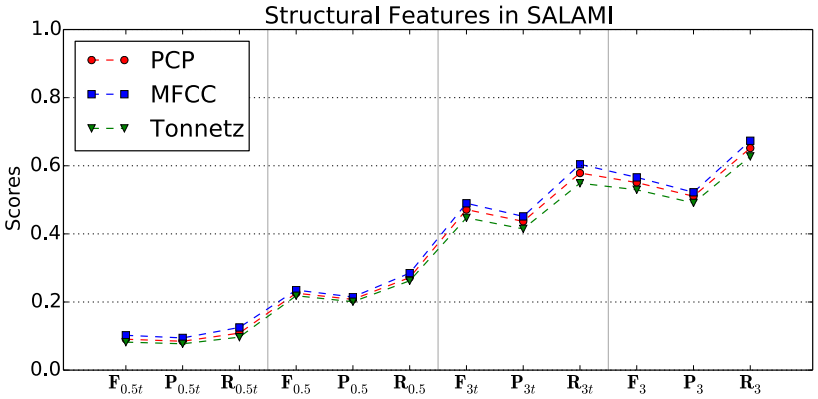


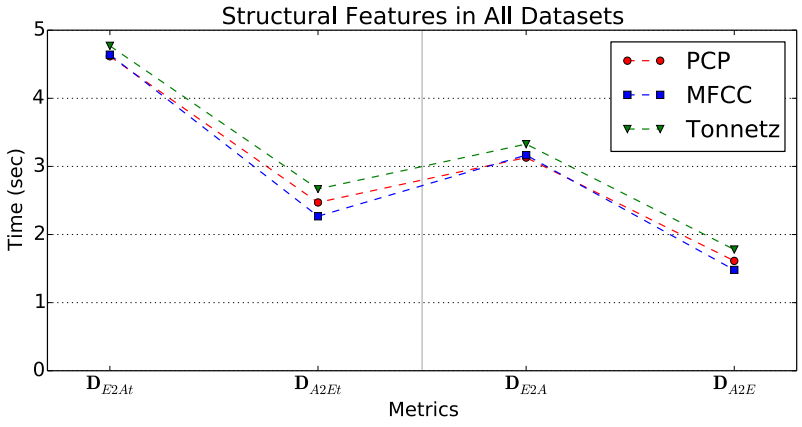


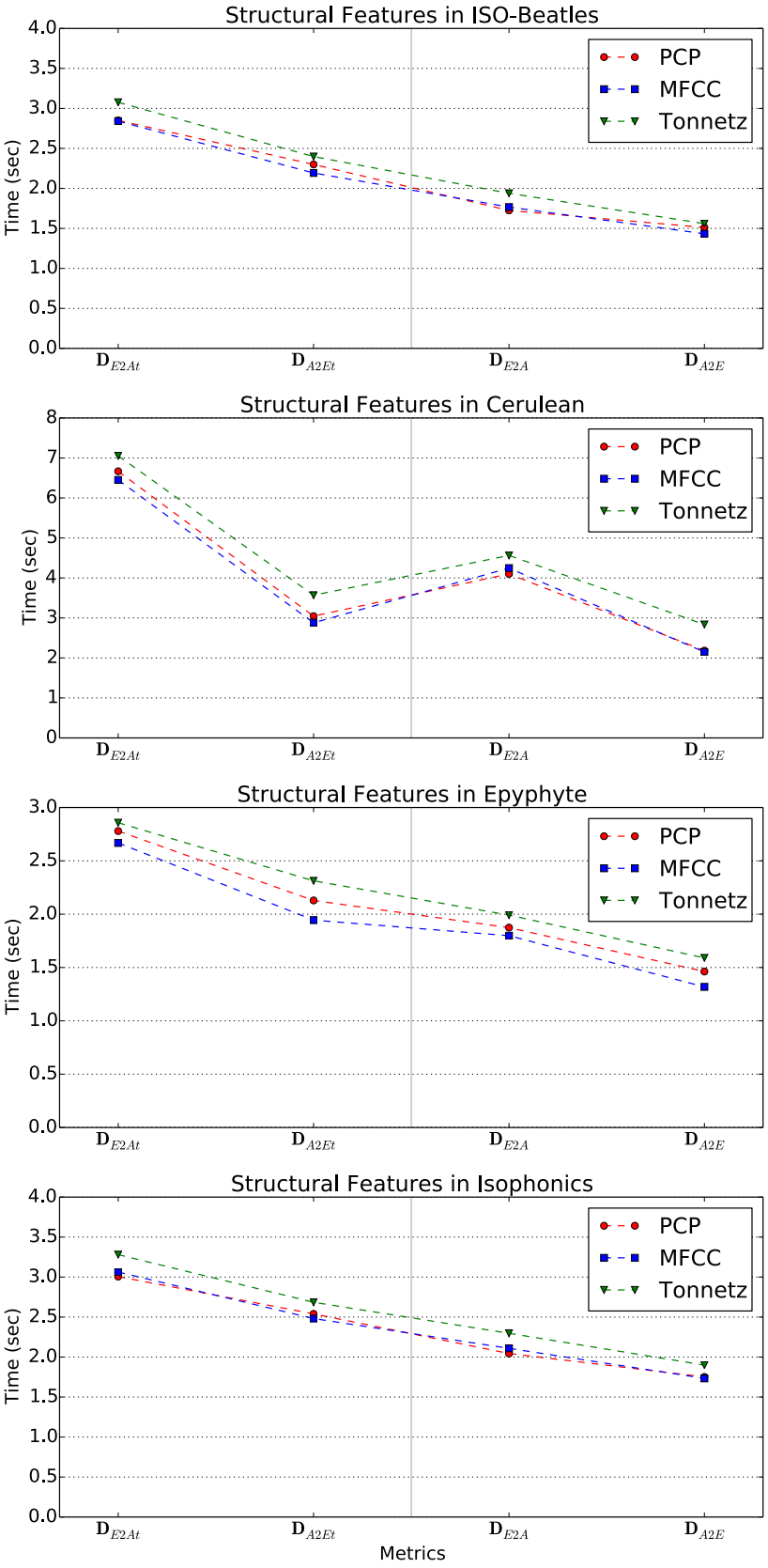


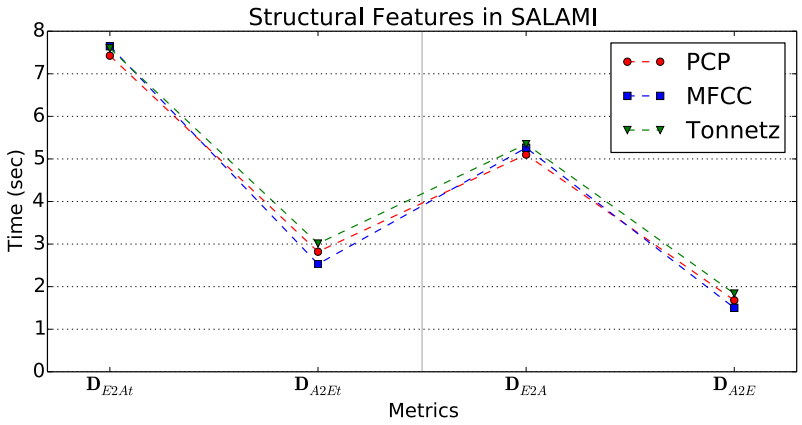

Structural Features in Epyphyte
PCP
MFCC
Tonnetz
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$


Structural Features in Isophonics
PCP
MFCC
Tonnetz
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$


Structural Features in SALAMI
PCP
MFCC
Tonnetz
Scores
1.0
0.8
0.6
0.4
0.2
0.0
$\mathbf{F}_{0.5t}$ $\mathbf{P}_{0.5t}$ $\mathbf{R}_{0.5t}$ $\mathbf{F}_{0.5}$ $\mathbf{P}_{0.5}$ $\mathbf{R}_{0.5}$ $\mathbf{F}_{3t}$ $\mathbf{P}_{3t}$ $\mathbf{R}_{3t}$ $\mathbf{F}_{3}$ $\mathbf{P}_{3}$ $\mathbf{R}_{3}$


Structural Features in All Datasets
PCP
MFCC
Tonnetz
Time (sec)
5
4
3
2
1
0
$\mathbf{D}_{E2At}$ $\mathbf{D}_{A2Et}$ $\mathbf{D}_{E2A}$ $\mathbf{D}_{A2E}$
Metrics


Structural Features in ISO-Beatles
Structural Features in Cerulean
Structural Features in Epyphyte
Structural Features in Isophonics
PCP
MFCC
Tonnetz
Time (sec)
Metrics
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$


Structural Features in SALAMI
PCP
MFCC
Tonnetz
Time (sec)
0
1
2
3
4
5
6
7
8
$\mathbf{D}_{E2At}$
$\mathbf{D}_{A2Et}$
$\mathbf{D}_{E2A}$
$\mathbf{D}_{A2E}$
Metrics