

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Música
Programa de Pós-Graduação em Música

TAIRONE NUNES MAGALHÃES

**IRACEMA: FROM NOTE ONSET DETECTION CHALLENGES TOWARDS
AN AUDIO CONTENT ANALYSIS LIBRARY FOR THE EMPIRICAL STUDY
OF MUSIC PERFORMANCE**

BELO HORIZONTE

2021

Tairone Nunes Magalhães

**IRACEMA: FROM NOTE ONSET DETECTION CHALLENGES TOWARDS
AN AUDIO CONTENT ANALYSIS LIBRARY FOR THE EMPIRICAL STUDY
OF MUSIC PERFORMANCE**

Tese apresentada ao Programa de Pós-Graduação em Música da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do título de Doutor em Música.

Orientador: Mauricio Alves Loureiro

Belo Horizonte

2021

M188i Magalhães, Tairone Nunes.

Iracema [manuscrito]: from note onset detection challenges towards an audio content analysis library for the empirical study of music performance / Tairone Nunes Magalhães. - 2021.
115 f., enc.; il.

Orientador: Maurício Alves Loureiro.

Linha de pesquisa: Sonologia.

Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Música.

Inclui bibliografia.

1. Música - Teses. 2. Performance Musical. 3. Música e tecnologia. 4. Música para clarinete. I. Loureiro, Maurício Alves. II. Universidade Federal de Minas Gerais. Escola de Música. III. Título.

CDD: 789.96



UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE MÚSICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MÚSICA

FOLHA DE APROVAÇÃO

Tese defendida pelo aluno **Tairone Nunes Magalhães**, em 28 de abril de 2021, e aprovada pela Banca Examinadora constituída pelos Professores:

Prof. Dr. Maurício Alves Loureiro
Universidade Federal de Minas Gerais
(orientador)

Prof. Dr. Jose Augusto Mannis
Universidade Estadual de Campinas

Prof. Dr. Hugo Bastos de Paula
Pontifícia Universidade Católica de Minas Gerais

Prof. Dr. Flavio Luiz Schiavoni
Universidade Federal de São João del-Rei

Prof. Dr. Sérgio Freire Garcia
Universidade Federal de Minas Gerais

Dr. Thiago de Almeida Magalhães Campolina
Campolina Toxicologia Computação Nuclear e Áudio

Prof. Dr. Davi Alves Mota
Programa de Pós-Doutorado com Experiência no Exterior
Universidade Federal de Minas Gerais



Documento assinado eletronicamente por **Mauricio Alves Loureiro, Professor do Magistério Superior**, em 30/04/2021, às 12:33, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Sergio Freire Garcia, Professor do Magistério Superior**, em 30/04/2021, às 13:57, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Hugo Bastos de Paula, Usuário Externo**, em 30/04/2021, às 14:42, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Thiago de Almeida Magalhães Campolina, Usuário Externo**, em 30/04/2021, às 15:07, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Davi Alves Mota, Usuário Externo**, em 30/04/2021, às 15:32, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **José Augusto Mannis, Usuário Externo**, em 30/04/2021, às 20:31, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Flávio Luiz Schiavoni, Usuário Externo**, em 03/05/2021, às 13:29, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site

[https://sei.ufmg.br/sei/controlador_externo.php?](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0)

[acao=documento_conferir&id_orgao_acesso_externo=0](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0686591** e o código CRC **18F0AFD5**.

Acknowledgements

The completion of this thesis could not have been accomplished without the support of many great people who helped me in several different ways during the last couple of years. First of all, I want to express my deep gratitude to my research supervisor, Prof. Mauricio Loureiro, whom I first met seventeen years ago (in 2004), in the course “Acoustics and Music”, in which he was the instructor. I have had the chance to work under his supervision on four different occasions over those years, and all those opportunities have been excellent learning experiences. I want to thank him for his enthusiasm, continuous encouragement, and the vast amount of knowledge shared during all those years of my academic life.

Thanks to my fellow labmates from CEGeME for the great happy hours, spirited coffee breaks, and insightful contributions to this thesis. I am very grateful to Rodrigo Borges, Thiago Campolina, Aluizio Oliveira, Davi Mota, Gustavo Machado, and Felipe Barros (*fefo*) for their contributions and long discussions around this work’s subject. I also want to thank Luís Umbelino for making the scores that are included in the thesis. Thanks to Prof. Fernando Braga for giving me the chance to join him as a teaching intern in the course Fundamentals of Audio and Recording III. It was a great opportunity to give lectures on a really fun subject, and I also learned a lot from him. Thanks to Kunumi and my work colleagues for their support during the final months of this text’s writing. The development of this work has been financially supported by CAPES / Brazil (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and CNPq/Brazil (Conselho Nacional de Desenvolvimento Científico e Tecnológico), to which I would like to extend my gratitude.

Lastly, I want to thank my family for the unconditional love and support throughout my Ph.D studies. Finishing a thesis during a pandemic brought difficulties that I could not have imagined at the beginning of the process. I am deeply grateful to my parents, Ernane and Vera, and to my sister, Cinara, who were always there for me when I needed them.

Abstract

The earliest empirical studies on music performance date back to the end of the nineteenth century, when the first mechanical devices capable of recording the actions of pianists on the instrument (key presses) were invented. Since then, many technologies that open up new possibilities for collecting data from musical performances have been invented or developed, including techniques for extracting information directly from audio recordings. These techniques, which have been driven by the fast-paced technological development in computer-related fields over the last decades, are the subject matter of this thesis. We introduce a new software library called *Iracema*, which contains techniques for extracting patterns of manipulation of timing, energy, and spectral content from monophonic audio recordings. In this endeavor, the clarinet is the instrument chosen for the baseline experiments and models, but most of the presented techniques should also work for other monophonic instruments. One of the most critical steps in studying musical performances is the detection of the note onsets because our perception of timing is strongly tied to this variable. We pay special attention to this topic, proposing an interactive web interface for the precise manual annotation of note onsets and conducting an experiment to assess the typical measurement error involved in this kind of task for clarinet recordings. We also propose an annotated dataset of solo clarinet recordings containing approximately 23 minutes of audio and a total of 3551 note onsets. Using this dataset, we train a convolutional neural network to generate a model for automatic note onset detection specifically on clarinet recordings and compare its results to other onset detection models. Finally, we discuss a study case using recordings of a clarinet excerpt by a few different clarinetists to demonstrate the use of the proposed library.

Key-words: empirical musicology, note onset detection, music performance, music information retrieval, machine learning.

Resumo

Os primeiros estudos empíricos em performance musical datam do final do século XIX, quando foram criados os primeiros dispositivos mecânicos capazes de gravar as ações de pianistas no instrumento (o pressionar das teclas). Desde então, várias tecnologias que abrem novas possibilidades de coleta de dados de performances musicais foram inventadas ou aprimoradas, incluindo técnicas de extração de informação a partir do sinal de áudio. Tais técnicas, que se aprimoraram em ritmo acentuado ao longo das últimas décadas, impulsionadas pelo rápido desenvolvimento das mais diversas áreas correlatas à computação, são o foco do presente trabalho. Propomos aqui uma nova biblioteca de software chamada Iracema, que contém técnicas para a extração de padrões temporais, de energia, e conteúdo espectral, a partir de gravações de áudio monofônicas. Escolhemos a clarineta como o instrumento a ser utilizado nos experimentos de referência e modelos propostos, mas a maior parte das técnicas aqui apresentadas pode ser aplicada a outros instrumentos monofônicos. Um dos passos mais importantes no estudo de performances musicais é a detecção dos instantes de *onset* (início) das notas musicais, já que a nossa percepção rítmica (temporal) está fortemente associada a tais instantes. A este assunto dedicamos atenção especial, e propomos uma interface *web* para a anotação manual precisa dos instantes de onset, além de realizar um experimento para avaliar o erro típico de anotação neste tipo de tarefa, para gravações de clarineta. Também propomos uma base de dados anotada contendo aproximadamente 23 minutos de áudio tocados na clarineta, contendo um total de 3551 onsets. Utilizando esta base de dados, treinamos uma rede neuronal convolucional para obter um modelo para detecção automática de onsets especificamente em gravações de clarineta, e comparamos os seus resultados com os de outros modelos. Por fim, exemplificamos e demonstramos o uso da biblioteca proposta por meio de um estudo de caso, envolvendo a análise de gravações de um excerto de uma peça, tocada por vários clarinetistas.

Palavras-chaves: musicologia empírica, detecção de início de nota, performance musical, extração da informação musical, aprendizado de máquina.

List of Figures

Figure 1 – Parts of the clarinet. Adapted from the original image uploaded to Wikipedia by user Ruizo, under the Creative Commons Attribution-Share Alike 1.0 Generic license. Downloaded from https://commons.wikimedia.org	8
Figure 2 – Simplified illustration showing the relationship between the length L of a theoretical closed pipe and wavelengths λ_n of the fundamental mode and its harmonics. The closed end of the pipe coincides with pressure anti-nodes for the odd-numbered harmonic modes. In contrast, it coincides with the pressure nodes for the even-numbered harmonics. Adapted from (WOLFE, 2002).	9
Figure 3 – Illustration representing the waveform of two consecutive notes, with the indication of the note onsets and offsets. The segment corresponding to the note duration extends from the note onset to its corresponding offset. The IOI spans two adjacent note onsets.	14
Figure 4 – Mechanical apparatus for recording the fingering of pianists. Extracted from (BINET; COURTIER, 1895).	16
Figure 5 – Example of graphics produced by the apparatus described in figure 4 during the execution of two excerpts. Extracted from (BINET; COURTIER, 1895).	16
Figure 6 – Pitch and intensity measurements for an excerpt from Ave Maria (Schubert-Wilhelmj) played on the violin. The upper curve in each chart corresponds to the pitch, and also display the nominal durations (according to the score) using musical notation symbols. The bottom curve displays the dynamics of the notes. Extracted from (SEASHORE, 1938).	18
Figure 7 – A Convolutional neural network applied to an image classification task (digits recognition). Extracted from (LECUN et al., 1998).	26
Figure 8 – Scores for the three clarinet excerpts listed in table 2.	33

Figure 9 – Binned scatter plot showing the measurement error of the annotations for the three excerpts listed in table 2. The size of the circle represents the number of times a specific error value occurred, like a histogram, with bin width of 2 ms. The error consists in the difference between each onset annotation and the median.	34
Figure 10 – Convolutional Neural Network for onset detection. Adapted from (SCHLÜTER; BÖCK, 2014).	40
Figure 11 – Diagram showing the core classes of Iracema.	51
Figure 12 – Illustration of a simplified version of the note envelope, composed by three segments: attack, sustain and release.	62
Figure 13 – Illustration showing the areas S_1 and S_2 used to calculate the legato index.	63
Figure 14 – Score for the excerpt of <i>Peter and the Wolf opus 67</i> by Sergei Prokofiev.	64
Figure 15 – Local tempo part 1.	66
Figure 16 – Local tempo part 2.	67
Figure 17 – Local tempo for two performances by the same clarinetist.	67
Figure 18 – Correlation matrix for the local tempi for all the performances.	68
Figure 19 – Legato indexes for performances A to F.	69
Figure 20 – Legato indexes for performances G to K.	70
Figure 21 – Correlation matrix for the legato indexes for all the performances.	71
Figure 22 – Editing an example experiment in the administration interface of Audio Segment Annotator.	81
Figure 23 – The resulting interactive annotation interface for one audio file in an example experiment. It shows the overall waveform on top, followed by the zoomed-in spectrogram and audio waveform. In the zoomed-in waveform, there are two annotated note onset points (red vertical lines), and one vibrato segment (blue box). The black vertical line is the playhead.	82
Figure 24 – Resulting plots showing the waveform, RMS and spectrogram, obtained running the code in listing B.6 for the excerpt <code>stravinsky.wav</code>	86
Figure 25 – Resulting plots showing the estimated pitch and harmonics, obtained running the code shown in listing B.7 for the excerpt <code>stravinsky.wav</code> .	87
Figure 26 – Resulting plot for the features calculated in listing B.8.	88

List of Tables

Table 1 – Basic categories of expressive parameters and some examples of related acoustic parameters. This is not a strict categorization, since some acoustic parameters might pertain to multiple categories.	12
Table 2 – Clarinet excerpts from the dataset <i>clari-onsets-3</i>	32
Table 3 – Absolute error for the manual annotations obtained for the dataset <i>clari-onsets-3</i>	35
Table 4 – Clarinet excerpts from the dataset <i>clari-onsets-50</i>	36
Table 5 – Benchmark results for the dataset <i>clari-onsets-50</i>	45
Table 6 – Benchmark results for the dataset <i>clari-onsets-3</i>	45
Table 7 – Results obtained for the clarinet-specific models on the dataset <i>clari-onsets-50</i> , using 10-fold cross-validation.	47
Table 8 – Results obtained for the clarinet-specific models on the dataset <i>clari-onsets-3</i>	47
Table 9 – Examples of numeric values corresponding to nominal durations in the score.	62
Table 10 – List of performances analyzed, indicating the label associated to the performance, a numeric id corresponding to the performer and the indication of the level of expertise of the clarinetist.	64

Abbreviations and Acronyms

BPM	Beats Per Minute
CNN	Convolutional Neural Network
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
FFT	Fast Fourier Transform
HFC	High Frequency Content
IOI	Inter-Onset Interval
LSTM	Long Short-Term Memory
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
ODF	Onset Detection Function
ReLU	Rectified Linear Unit
RMS	Root Mean Square
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
STFT	Short Time Fourier Transform

Contents

1	INTRODUCTION	1
1.1	Motivation	2
1.2	Objectives	3
1.3	Structure of the Thesis	5
2	THE CLARINET	7
2.1	The Clarinet as an Ideal Closed Pipe	9
2.2	Range	10
2.3	Control Parameters on the Clarinet	10
2.4	Articulation	10
3	EMPIRICAL RESEARCH ON MUSIC PERFORMANCE	11
3.1	Expressiveness and Acoustic Parameters	11
3.2	Note Onset, Offset, and IOI	13
3.3	Empirical Study of the Performance	14
3.3.1	The Pioneers	15
3.3.2	Extracting Information at the Acoustic Level	17
3.3.3	Studies of Clarinet Performance	19
4	MACHINE LEARNING	21
4.1	Deep Learning	21
4.2	Artificial Neural Networks	22
4.3	Supervised Learning	23
4.4	Cross-Entropy Loss	23
4.5	Gradient-Based Optimization	24
4.6	Long Short-Term Memory Networks	25
4.7	Convolutional Neural Networks	26
4.8	Rectified Linear Units	26
4.9	Dropout	27
4.10	K-Fold Cross Validation	27
4.11	Evaluation Metrics	28
5	NOTE ONSET ANNOTATION	30
5.1	Clari-Onsets-3	32

5.1.1	Methods	32
5.1.2	Results	32
5.2	Clari-Onsets-50	35
5.3	Discussion	35
6	NOTE ONSET DETECTION	38
6.1	Methods	38
6.1.1	CNN	39
6.1.2	RNN	41
6.1.3	SuperFlux	42
6.1.4	Training a CNN for Onset Detection on the Clarinet	42
6.1.5	Evaluation Criteria	43
6.2	Results	45
6.2.1	Benchmarking	45
6.2.2	Results for the Clarinet-Specific Model	46
6.3	Discussion	47
6.4	Future Prospects	48
7	IRACEMA	49
7.1	Architecture	50
7.2	Modules and Functionalities	52
7.2.1	Pitch Detection	53
7.2.1.1	Harmonic Product Spectrum	53
7.2.1.2	Expan Pitch Detection Algorithm	54
7.2.1.3	CREPE	54
7.2.2	Harmonics	54
7.2.3	Classic Features	55
7.2.3.1	Peak Envelope	55
7.2.3.2	RMS	55
7.2.3.3	Zero-Crossing Rate	56
7.2.3.4	Spectral Flatness	56
7.2.3.5	High Frequency Content	56
7.2.3.6	Spectral Centroid	57
7.2.3.7	Spectral Spread	57
7.2.3.8	Spectral Skewness	57
7.2.3.9	Spectral Kurtosis	58
7.2.3.10	Spectral Flux	58
7.2.3.11	Harmonic Centroid	58
7.2.3.12	Spectral Entropy	58
7.2.3.13	Spectral Energy	59

7.2.3.14	Harmonic Energy	59
7.2.3.15	Noisiness	59
7.2.3.16	Odd-to-Even Ratio	59
7.2.4	Note Onset Detection	60
7.2.4.1	CNN Model	60
7.2.4.2	Adaptative RMS	60
7.2.4.3	Pitch Change	60
7.2.4.4	Derivative of the RMS	60
7.2.5	Note Envelope Segmentation	61
7.2.6	Expressiveness Features	61
7.2.6.1	Local Tempo	61
7.2.6.2	Legato Index	62
7.3	Case Study	63
7.3.1	Methods	64
7.3.2	Results	65
7.3.2.1	Local Tempo	65
7.3.2.2	Legato Index	68
7.4	Future Prospects	68
8	CONCLUSION	72
	BIBLIOGRAPHY	74
	APPENDIX A AUDIO SEGMENT ANNOTATOR	80
	APPENDIX B CODE EXAMPLES	83
B.1	Loading and Processing Audio	84
B.2	Extracting Features from Audio	84
B.3	Note Segmentation	88

1

Introduction

Just about a century and a few decades ago, any person wishing to listen to a musical piece would need to be present during its performance to fulfill their desire. After all, sound recording and playback devices would only be invented at the end of the nineteenth century. The development of recording and playback technologies radically transformed our culture and relationship with music. As a new industry emerged, people started to establish new ways to consume, practice, and produce music. Over the years, this phenomenon completely reshaped our culture and relationship with music. Nowadays, it has become a ubiquitous element in most people's lives: it is easily accessible and available everywhere, from advertisements and soundtracks to streaming services and interactive systems. Many of the current music streaming platforms now have tens of millions of tracks available, which users can play anywhere they have an internet connection at their disposal. Nonetheless, while this process might seem to have created a disruption between the music and the performance, people still choose to listen to a piece of music not just because of the composition itself but also for its different interpretation possibilities. That is a consequence of the fact that every musical performance is unique¹.

There is a significant amount of information in musical performances that do not depend only on the piece's underlying structure but instead on several other factors, like individual playing style, performance traditions, the mood of the performer, or expressive choices made by them. Other factors are the environment where the performance took place, the instrument played, random manipulations that might occur by chance, etc. In the face of such idiosyncrasies, listeners establish their individual

¹ Repp (1999, p. 239) mentions that “[n]o two performances of the same work are exactly alike, and this is often true even for repeated renditions of the same piece.”

preferences, often favoring a specific rendition of a piece as more compelling, for some reason, than others. This scenario gives rise to many questions regarding the complex process by which a performer shapes a musical composition into its final rendition.

The empirical study of music performance is relatively recent, with the seminal works dating back to the turn of the twentieth century. As stated by Clarke (2004, p. 77), “only once methods had been developed to record either the sounds of performance, or the actions of instruments, was any kind of detailed [empirical] study possible — and so the piano roll, record, magnetic tape, and computer have all played their part at different stages in the short history of empirical studies of performance”. Undoubtedly, the twentieth century’s technological developments progressively extended our capacity to measure relevant information from the performance. Over the last decades, we have witnessed considerable growth in research in the field of music performance. Palmer (1997), Gabrielsson (2003), Goebel, Dixon and De Poli (2005), and Lerch et al. (2019) provide excellent surveys on the topic that indicate this growth. The availability of new tools and technologies that enable the extraction of information from performances has played a pivotal role in this surge. The contributions in audio analysis by researchers from the Music Information Retrieval community were greatly beneficial to the field, as Lerch et al. (2019, p. 1-2) points out, even though most of them were not motivated by the interest in conducting research in music performance, but by other tasks, such as retrieving information from large music databases. Since data acquisition is a fundamental step in the study of music performance, we believe that the continuous development of purpose-oriented tools to extract information from audio recordings of performances will be greatly beneficial for the research on the field. Equally important will be the development of better techniques for obtaining more meaningful representations of musical content, like higher-level descriptors of musical expressiveness in the performance, for example.

1.1 Motivation

At the beginning of my academic life, I joined a multidisciplinary research group focused on empirical research on musical performance called CEGeME (Center for Research on Musical Gesture & Expression), composed by students and researchers from a diverse range of fields spanning Engineering, Computer Science, Physics, Phonology, and Music (LOUREIRO et al., 2019). Although since its foundation the major interest of the group has been the empirical study of music performance, extracting information from performances was always a major challenge faced by most members of the group. Consequently, developing methods of audio content extraction became one of its most prominent lines of research. The first research project dedicated to this purpose

consisted of a software tool called *Expan* (LOUREIRO et al., 2008; CAMPOLINA; MOTA; LOUREIRO, 2009), a *Matlab* toolbox for the extraction of data from performances on the clarinet and other monophonic instruments. I was part of the team that started the development of that tool, which was subsequently employed in many studies conducted by the group. That tool was refactored over the years, but essentially, its functionalities and methods remained the same, except for minor improvements, always performed to suit the needs of specific research projects. Those functionalities were also completely based on Digital Signal Processing, lacking the potential of improvement brought by the field of machine learning. We hypothesised that the performance of the note onset detection methods, in particular, could be greatly improved by using modern machine learning algorithms. This scenario motivated us to propose and develop a new tool from scratch, as a PhD project, this time using the language *Python*, which is freely available and can be easily adapted to run on multiple platforms. Python is also widely used by machine learning researchers, so many frameworks could be easily employed in the development of newer information extraction methods. We decided to call this new tool *Iracema*, named after the novel by the Brazilian writer José de Alencar. It proposes an architecture containing abstractions for easier manipulation of time series and extraction of information from audio. It also introduces newer methods and techniques, moving towards machine learning models in order to improve note onset detection, and towards the extraction of expressive features from audio.

1.2 Objectives

The objective of this work is to develop and evaluate computational techniques for audio content analysis, specifically designed to support research projects on music performance that rely on monophonic recordings (recordings with a single voice, or melodic line) as source of data. Such techniques seek to extract information that is meaningful for the representation/parametrization of musical expression in the performance. In this endeavor, the clarinet is the instrument used for the baseline experiments and models, but in theory, the presented techniques should work well for other monophonic instruments. Since detecting the instants of note onset is greatly important for the parametric study of music performance, we devote a large part of our efforts to obtain a good onset detection model for the clarinet. The main product of the work in this thesis is the library *Iracema*, which contains the implementation of all the algorithms and models for extracting information from music recordings addressed in the text.

The sound produced by the musician during a performance cannot be thought to contain the whole performance. Lately, many studies have been focusing on the musician's body movement, his interaction with the audience, as well as other phenomena,

which cannot be investigated only by analyzing the sound. While these non-acoustic aspects are undisputedly greatly important, they are out of the scope of this work. The discussions and the definitions, tools, and techniques here presented will deal exclusively with the acoustic component of music performance, employing digital audio recordings as source material.

While there are several examples of scientific works that use polyphonic recordings as a source of investigation (REPP, 1992; BERNAYS; TRAUBE, 2014; KOREN; GINGRAS, 2014), we chose to limit our scope to monophonic recordings². In many musical instruments, the excitation that produces sound happens only during a short interval at the beginning of a note (i.e., the plucking of strings in a guitar or a hammer hitting the strings of a piano). Contrastingly, in woodwind and brass instruments, the player continuously feeds energy into the system, employing high-pressure air from his lungs. Therefore, due to the dynamic control that the player has over the acoustic properties of the sound, a single note will contain over its duration a substantial amount of expressive information, e.g., timbral manipulations or dynamic intensity variations. It is harder to extract this kind of information from polyphonic music signals such as a full orchestral recording than from signals of a single source, especially when one cannot afford to lose any relevant expressive information pertaining to individual instruments/performers. For example, to analyze the timbral manipulations that a single clarinetist performs in some specific notes on a full orchestral recording, it would be necessary to isolate the information pertinent to that single clarinet. This approach would probably depend on a highly sophisticated source separation algorithm, which would have to be absolutely precise in retrieving only the data belonging to that clarinetist from a highly complex stream containing all the other instruments. This level of precision is still unfeasible, taking into account that any loss would have an impact on the subsequent timbral analysis. Hence, to study polyphonic compositions, the individual voices that constitute them must be recorded and analyzed separately and processed as monophonic signals. This simplification enables a more thorough investigation of attributes that would otherwise be very hard to analyze.

Given this context, we decided to choose the clarinet as the baseline instrument for the discussion in this thesis; so the methods and models will all be designed for this particular instrument. Based on previous experience extracting data from clarinet recordings, we know that its note onsets are particularly difficult to extract due to the typically soft note attacks that the instrument produces. So we will direct much of our attention to obtaining a robust method for clarinet onset detection, which we believe will be able to generalize, to some extent, to other monophonic instruments, specially woodwind and brass instruments.

Thus, the main objectives of this thesis are: (1) devising a robust onset detection

method for the clarinet; (2) implementing a software library (Iracema) containing methods for estimating acoustic parameters such as note onset, intensity, pitch, note envelope, duration, articulation, and spectral content from clarinet recordings; (3) employing the implemented methods in a brief case study to demonstrate the use of the library.

1.3 Structure of the Thesis

One of the key ideas behind the structure we chose to adopt in this thesis is to keep the chapters as independent as possible. Ideally, a chapter should be able to exist, as much as possible, as a self-contained discussion. The first chapter is the introduction of the thesis and is followed by three chapters (2-4) containing its theoretical foundation and literature review. The next three chapters (5-7) contain an internal structure comprising, each, their own methodological discussions and results. Chapter 7, in particular also includes references to previous studies that proposed or described the methods implemented in the library Iracema. Lastly, chapter 8 comprises the concluding remarks of the thesis.

Chapter 1 – Introduction The current chapter addresses the contextualization, motivation, objective, and scope of the current work. It also provides information about the structure of the thesis.

Chapter 2 – The Clarinet Provides a brief description of the clarinet, its acoustic properties, etc.

Chapter 3 – Empirical Research on Music Performance Defines concepts that are pertinent to the empirical study of music performance, like the acoustic parameters related to musical expressiveness, and the concept of note onset. Furthermore, provides a brief literature review on studies that are considered historically important in the field or relevant for this thesis.

Chapter 4 – Machine Learning Introduces Machine Learning concepts and techniques that are employed in this thesis.

Chapter 5 – Note Onset Annotation Discusses the process of manually annotating onsets and introduces a specialized interface for manual segmentation developed for this thesis. It also describes the process of annotation of onsets for two clarinet datasets, one of which was annotated five times, to obtain a more reliable estimation of the onset the annotation error.

² Monophonic recordings, containing a single melodic line, but not necessarily monophonic compositions. A monophonic clarinet part in a polyphonic orchestral composition could be studied in isolation, for example.

Chapter 6 – Note Onset Detection Discusses methods for automatic onset detection and compare the results from a few benchmarking methods to a specific model for clarinet onset detection, trained on our own clarinet datasets.

Chapter 7 – Iracema Introduces Iracema, a Python library for audio content analysis developed during the course of this work, which seeks to support research in music performance with methods for note segmentation and extraction of features from audio. Additionally, presents a case study using Iracema to analyze recordings of performances on the clarinet.

Chapter 8 – Conclusion Concluding remarks.

2

The Clarinet

The clarinet is a single-reed woodwind instrument with a roughly cylindrical bore. It is usually made of wood (but there are also instruments made of plastic or metal) and is equipped with a system of silver-plated keys used to open and close the toneholes distributed along the pipe. Its origins can be traced back to around 1700, when Johann Christoph Denner invented it in Nuremberg, Germany. Two predecessors of the clarinet were: the *chalumeau*, which had a range of a twelfth in the fundamental register of the clarinet, and the *baroque clarinet*, which produced higher notes, corresponding to the upper register of the clarinet. The clarinet combined them into a single instrument, capable of producing a wide pitch range (LAWSON, 2000, p. 11). Although the term clarinet can be used to refer to a family of instruments of different sizes and pitches, it will be used in this text to refer specifically to the more common *B♭* or *A* soprano clarinets.

The parts of the clarinet are shown in figure 1. The mouthpiece is a small tube with a flattened end onto which the reed is fixed, using the ligature. A narrow opening is left between the reed's tip and the mouthpiece. The clarinetist blows into the instrument the air from his mouth cavity, providing an approximately steady flow of high-pressure air. The reed, interacting with the bore, which acts as a resonator, will convert this energy into oscillations and sound waves (WOLFE, 2018). By opening and closing the toneholes, the clarinetist modifies the length of the wave in the tube, producing different pitches. The mechanism of the keys allows the clarinetist to produce combinations of open/closed toneholes that would be beyond the reach of his fingers if he had to use his finger pads. The bell is the flared end of the clarinet and is necessary to improve tone quality for the lower notes of the register. The barrel connects the mouthpiece to the

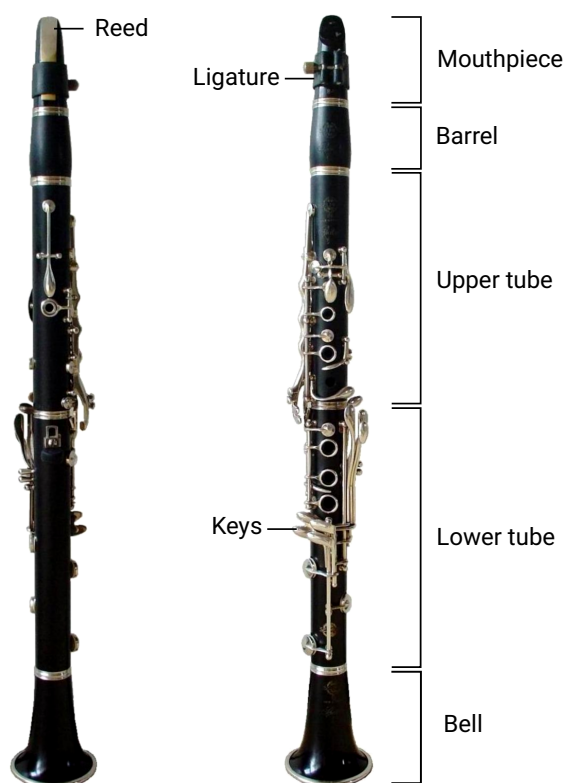


Figure 1 – Parts of the clarinet. Adapted from the original image uploaded to Wikipedia by user Ruizo, under the Creative Commons Attribution-Share Alike 1.0 Generic license. Downloaded from <https://commons.wikimedia.org>.

body of the instrument. Its position can be adjusted to modify the instrument's tuning.

The clarinet was subject to several upgrades over its history, the most noticeable ones being probably the changes in the keywork. In the late eighteenth century, keys were added to extend the actions of the index and little fingers of both hands, for opening and closing toneholes that would be out of the reach of the clarinetist's fingers. During the nineteenth century, a few different reasons motivated additional changes, as mentioned by Shackleton (1995, p.25): "keys were added to enable complex chromatic passages to be played more fluently and/or with better intonation"; also, keywork "was designed to render the tone of adjacent notes more even" and "to enable the instrument to play more loudly". In the late nineteenth century, more keys were added to facilitate trilling in specific notes. Improvements in the padding were important for the development of the instrument since older clarinets had air leak issues that hindered the addition of new keys. Therefore, good padding was crucial to avoid leakage, as the number of keys increased. Most modern clarinets use a system of keys called *Boehm*, devised in the middle of the nineteenth century by Hyacinthe Klosé and Auguste Buffet, *jeune*. Another system called *Oehler*, which uses different fingerings, was developed by Oskar Oehler in Berlin, and is also used by modern players, mostly in Germany and Austria.

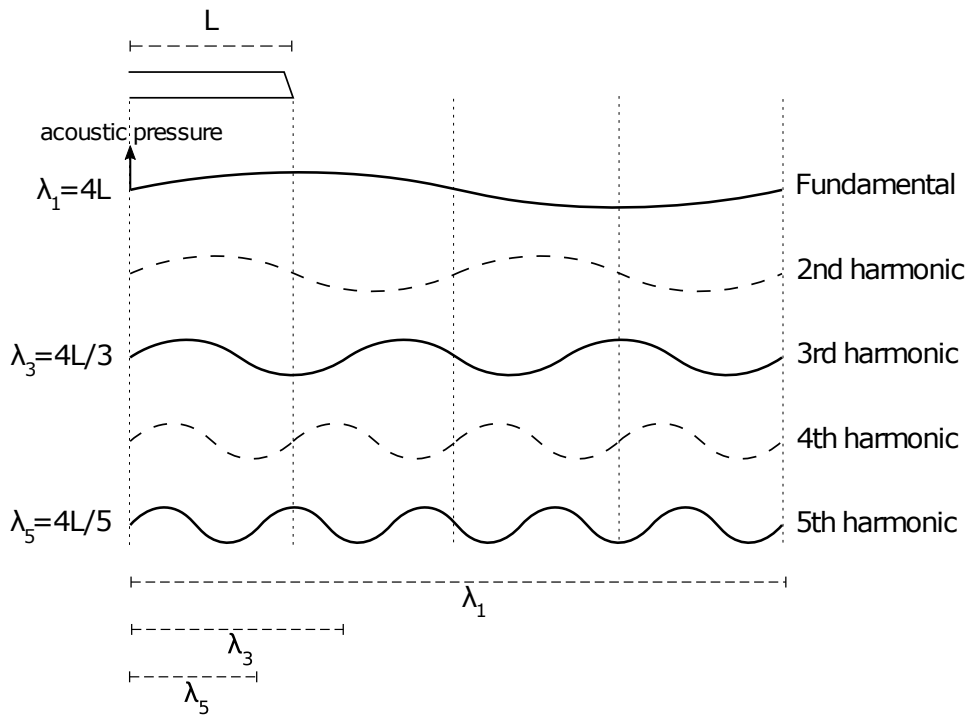


Figure 2 – Simplified illustration showing the relationship between the length a theoretical closed pipe L and wavelengths λ_n of the fundamental mode and its harmonics. The closed end of the pipe coincides with pressure anti-nodes for the odd-numbered harmonic modes. In contrast, it coincides with the pressure nodes for the even-numbered harmonics. Adapted from (WOLFE, 2002).

2.1 The Clarinet as an Ideal Closed Pipe

The clarinet's cylindrical bore is open at the bell's end and almost completely closed at the mouthpiece. The acoustic pressure varies dynamically inside the bore, except at the open end, where the acoustic pressure is close to the atmospheric pressure (pressure node). The maximum variation in pressure occurs in the mouthpiece (pressure anti-node). These two boundary conditions allow a stationary wave inside the bore with the wavelength of its fundamental mode λ_1 equal to four times the length of the pipe (L). For an ideal cylinder closed at one end and open at the other, the available vibration modes correspond to odd integer multiples of the fundamental (see figure 2). As a consequence, the even harmonics in the lower notes of the clarinet have less energy than the odd ones. Also, “the same fingering for a particular note in the first register will produce a note sounding a twelfth above it in the second register (one octave plus a fifth)” (EGOZY, 1995, p. 23).

2.2 Range

The clarinet has an extensive range of almost four octaves, being the instrument with the largest range among the woodwinds. Its lowest note is D_3 (≈ 146.8 Hz)¹, and its nominal highest note Bb_6 ($\approx 1,864.7$ Hz), but depending on the instrument and skill of the player, it is possible to produce even higher notes. The range of the clarinet can be split into three registers:

- Lowest register (chalumeau): D_3 to Ab_4
- Higher register (clarion): A_4 to Bb_5
- Altissimo register: B_5 to Bb_6

2.3 Control Parameters on the Clarinet

The clarinetist has a large number of parameters at his disposal that will modify the sound produced in the instrument. Besides changes in the key fingerings, “pressure in the mouth can be varied, so can the bite force, the position at which the lip presses on the reed, and sometimes the configuration of the vocal tract” (WOLFE, 2002). These control parameters will usually affect multiple sound properties; for example, “blowing pressure and lip force both affect each of loudness, pitch, and spectrum” (WOLFE, 2018, p.56). The vocal tract’s resonances are especially important at high frequencies and can be used to modify pitch and timbre or to control multiphonics.

2.4 Articulation

Articulation is an important expressive parameter in clarinet performances, and is related to the characteristics of the transitions between successive notes. Pàmies-Vilà, Hofmann and Chatziioannou (2018) showed that it is achieved by a combination of actions related to tonguing techniques (mostly achieved by interrupting the reed vibration with the tongue) and blowing actions. Note transitions performed with relatively steady air support tend to be perceived as *legato*. Conversely, transitions that exhibit a significant decay in the energy of the first note followed by a sharp attack in the second tend to be perceived as more articulated.

¹ All the notes in the text are specified in terms of concert pitch, labeled according to the numbering defined by the American Standard Pitch Notation (ASPN). Since the clarinet is a transposing instrument, for a Bb clarinet, a concert D_3 will be written as E_3 on the score (two semitones higher than the actual concert pitch).

3

Empirical Research on Music Performance

This chapter introduces a few concepts and provides a brief survey on the study of music performance, focusing particularly on the measurement of performance. First we will discuss the acoustic parameters of expressiveness in music performance and the definition of note onset. Then techniques and methodological approaches employed by researchers will be reviewed and discussed, starting with references to the work of the pioneer researchers in the field. Subsequently, we will shift our focus to studies that employ measurement techniques strictly on the acoustic level, finishing with studies that deal specifically with the clarinet.

3.1 Expressiveness and Acoustic Parameters

During a musical performance, the musician is able to manipulate a wide range of acoustic parameters that influence how listeners perceive it. According to Sloboda (2000, p. 398):

The expressive component of music performance is derived from intentional variations in performance parameters chosen by the performer to influence cognitive and aesthetic outcomes for the listener. The main expressive parameters available to performers are those of timing (both in note-onset and note-offset), loudness, pitch and timbre (sound quality).

Table 1 shows a few examples of performance parameters related to each of the categories mentioned by Sloboda.

category	related acoustic parameter
timing	note onset time
	note duration
	articulation ¹
	tempo
loudness	RMS envelope
timbre	spectral centroid
	logarithm of the attack time
	spectral flux
	harmonic spectral deviation
pitch	intonation
	vibrato ²
	portamento

Table 1 – Basic categories of expressive parameters and some examples of related acoustic parameters. This is not a strict categorization, since some acoustic parameters might pertain to multiple categories.

An interpretation of a piece consists in one specific configuration of those parameters. Since they are continuously variable, the multiple possibilities of interpretation lead to a combinatorial explosion (REPP, 1999, p. 239). On the other hand, “there are significant constraints on this variety, deriving both from the musical structure of a given work and from performance conventions that define what expressive actions are aesthetically pleasing within that structure”. Such conventions are not strict though, and they can change significantly in different cultural contexts or musical genres. As mentioned by Sloboda, “what would be considered appropriate for Chopin would be completely unacceptable for Mozart”. Moreover, the parameters available for manipulation will differ from one instrument to another. On the piano, for example, the performer cannot produce a continuous pitch shift, since the scale is fixed in discrete tonal steps. On the violin, on the other hand, a performer can easily do a soft and continuous glissando to connect two adjacent notes.

Many studies on music performance rely on the capability of extracting from the audio signal a good estimation of such parameters, or at least, of obtaining a set of features that correlate to them. To compare individual differences on the vibrato performed by different musicians, for example, a good pitch detection method is fundamental, as well as a good method to extract vibrato depth and rate. To investigate how a musician performs a rubato, it is necessary to employ a robust method to identify

¹ Articulation is also related to changes in energy over time, so it is also related to the category loudness.

² Vibrato is primarily associated to modulations in pitch, but it also produces modulations in timbre and loudness.

the note onsets, to subsequently estimate the tempo. Although in some situations it is possible to annotate this kind of information manually (if provided with the right visualization and annotation tool), it consists in a laborious task, so automatic methods are highly desirable. Automatic methods also open up new opportunities for dealing with larger datasets and for developing models for evaluation, categorization and analysis of musical performances.

3.2 Note Onset, Offset, and IOI

In a rough definition, the note onset represents the instant at which a musical note starts. However, there are a few different perspectives that lead to slightly different definitions of the onset. According to Vos and Rasch (1981, p. 323), the *physical onset* (also called *acoustic onset*) can be defined as the instant at which the generation of the acoustic stimulus starts. Differently, the *perceptual onset* is the instant the stimulus is first perceived by the listener. The physical onset precedes the perceptual onset, and their time difference is due to the fact that acoustic events typically follow a gradual increase in amplitude, so the listener will only be able to perceive it when it reaches a certain sound level. These two definitions differ from the *mechanical onset*, which is based on the measurement of actions on the instrument by means of sensors, capturing the instant “the instrument is triggered to make a sound” (LERCH, 2012). This is the case for the *note-on* events measured in MIDI instruments.

The relationship between the notes represented in a musical score and the note onsets in a performance is not biunivocal. The rendition of a piece might contain omissions, substitutions or additional notes. Repp (1996) reported a considerable amount of such inaccuracies in a study that analysed a MIDI database of piano performances by ten graduate student pianists. Most “intrusions and nearly all substitutions seemed contextually appropriate” in the pieces, and only 38% of the errors identified in the database were properly reported in an experiment, by listeners who were themselves pianists. Moreover, in music scores, there are “some ambiguous definition of the notes, such as is common for trills and other ornaments” (MAZZOLA, 2012). It’s not uncommon that the number of note onsets in the performance will be different from the number of notes written in the score.

We define segmentation as a task consisting in determining the boundaries of certain events or structures within an audio signal. In a speech signal, identifying where a phoneme starts and ends is an example of a segmentation task. In musical audio, the most important and widely studied type of segmentation task is note segmentation. The boundaries where a note lies can be defined by two points, the note *onset* and the *offset*, which respectively, correspond to the instants the note starts and ends.

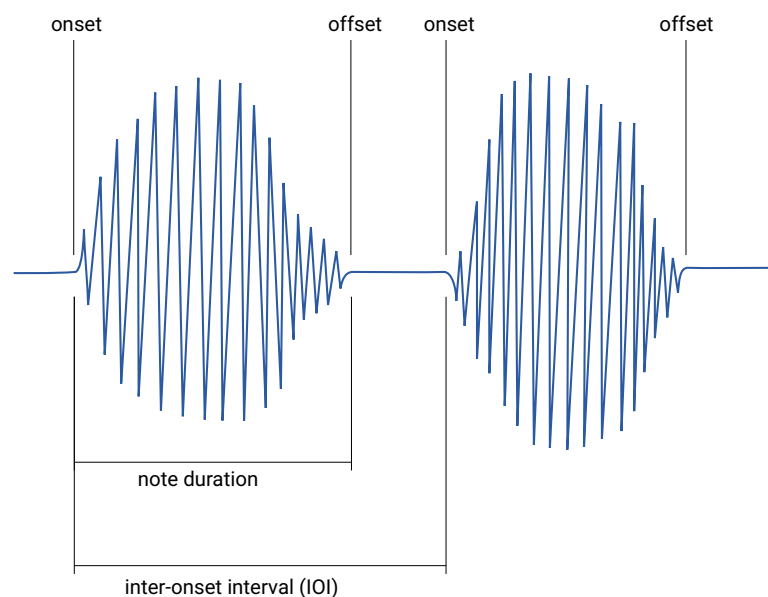


Figure 3 – Illustration representing the waveform of two consecutive notes, with the indication of the note onsets and offsets. The segment corresponding to the note duration extends from the note onset to its corresponding offset. The IOI spans two adjacent note onsets.

The detection of the onset is regarded as more important than the offset, owing to the fact that our perception of timing is more strongly tied to this variable. Since in music one of the most important expressive parameters is timing, the detection of note onsets is essential for the empirical study of the performance. The segment that spans two successive note onsets is called *inter-onset interval* (IOI) (see figure 3). While varying the length of the IOI in a passage will certainly affect the perception of timing, changing the position of the offset, within the limits of the IOI, is not likely to affect it. Instead, the perception of articulation will be influenced by this change.

3.3 Empirical Study of the Performance

Gabrielsson (2003) mentions that while reviewing empirical performance research up to the mid-1990's for a previous publication (GABRIELSSON, 1999), he had found around 500 works covering many different aspects of the performance, which he categorized into the following topics: performance planning, sight reading, improvisation, feedback in performance, motor processes, models of music performance, physical factors in performance, psychological and social factors, performance evaluation and measurements of performance. According to him, around 222 of those publications concerned the measurement of performance. Since then, this number has grown considerably. On the following sections we will focus on the most prominent or historically relevant publications, as well as the ones that are considered important in the context

of this thesis.

3.3.1 The Pioneers

The first empirical experiments on music performance were conducted by Binet and Courtier (1895) with the development of a mechanical apparatus to capture the strength employed by the pianist when pressing a key on the instrument (figure 4). It consisted of a system of rubber tubes placed beneath the keys of the piano, connected to a cylindric graphical recorder. Whenever a pianist pressed a key on the instrument, an air pulse would propagate inside the rubber tubes, moving a needle that would draw, in paper rolls, a curve representing the instant and intensity of the key press (figure 5). This enabled the researchers to analyse trills, accents and dynamic variations executed by pianists. Their study showed that when playing an accented note in a passage, there was a tendency to dettach it from the previous note and bind it to the next note. Accented notes also tended to have their duration prolonged. It is quite remarkable to realise that the authors actually expected the execution of a performance to closely match the nominal values from the score:

Binet and Courtier claimed for their system that it would enable the identification of faults in piano performance, that is to say deviations from the nominal values represented in the score – according to which each beat has the same duration, each crotchet lasts two quavers, and so forth. This is the most extreme form of musical textualism imaginable, according to which the point of performance is literally to reproduce the score. (COOK, 2013)

Since the beginning, most empirical studies in music performance have been conducted using keyboard instruments, for a couple of different reasons, including: their mechanism can be easily adapted to record keys presses; the physical separation between the player and the instrument enables the recording of measurements without affecting the playability of the instrument; and their percussive character makes them a good choice for the study of rhythmic skills (CLARKE, 2004, p. 78). Just a few years after the studies by Binet and Courtier, Ebhardt (1898) used electromechanical devices to investigate timing patterns during scale playing. Sears (1902) used a reed organ equipped with mercury electrical contacts to record the performances of church hymns by four different players, and investigated the durations of excerpts, measures, notes, accents and note overlaps. He reported a wide range of “personal difference” among the performances by different subjects.

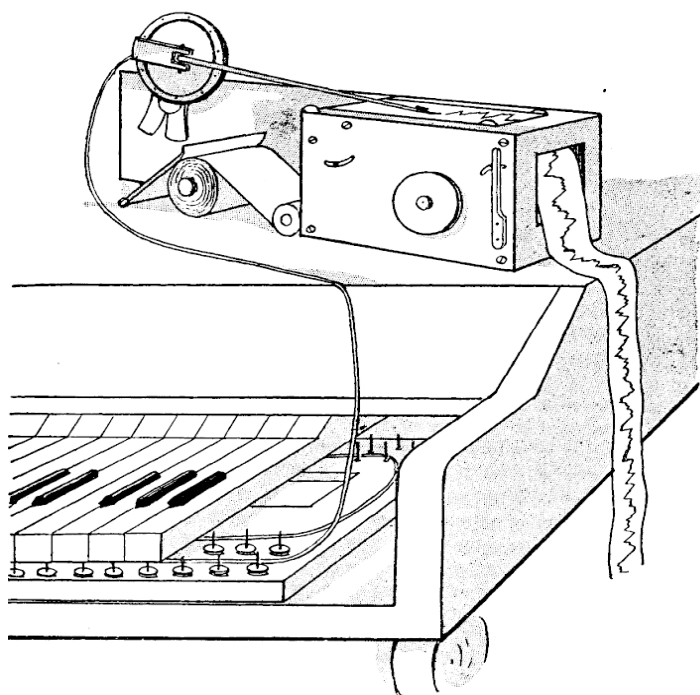


Figure 4 – Mechanical apparatus for recording the fingering of pianists. Extracted from (BINET; COURTIER, 1895).

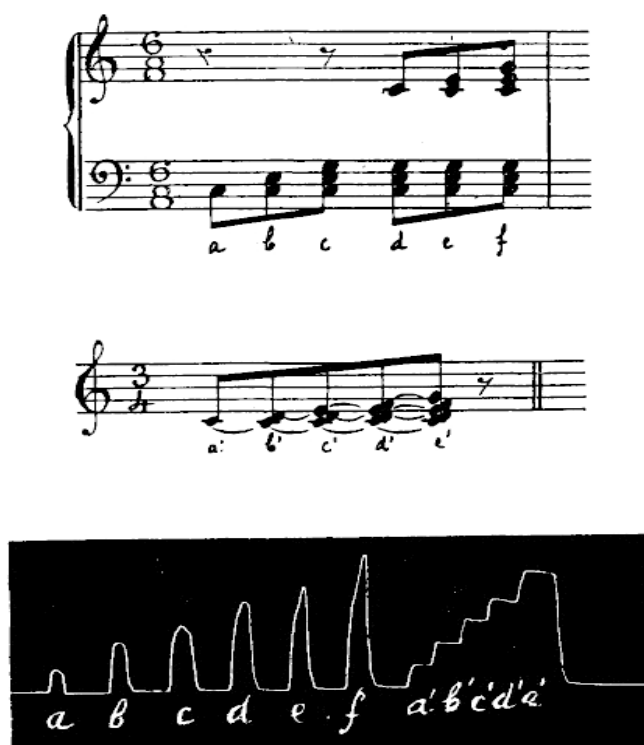


Figure 5 – Example of graphics produced by the apparatus described in figure 4 during the execution of two excerpts. Extracted from (BINET; COURTIER, 1895).

During the 1920's and 1930's, Carl E. Seashore made big contributions to music psychology, acoustics, psychoacoustics, and also music performance. He and his colleagues collected and summarized a large amount of information from performances in different instruments. Apart from the *Piano Camera*³ (SEASHORE, 1938), which improved significantly the potential for data collection on the piano, he also conducted studies on timing, intensity, pitch and timbre on the violin and the singing voice. Seashore proposed a formalization of musical expressiveness in terms of systematic deviations of acoustic parameters from a reference, proposing the score as the norm. His results revealed a surprising consistency in the deviations observed over different performances of a score, by the same player. In the words of Clarke (2004, p. 78), Seashore's works represent "the earliest extensive and systematic empirical work on performance, and identified many of the issues that have remained the preoccupations of subsequent research".

3.3.2 Extracting Information at the Acoustic Level

The first attempts to extract information from the performance at the acoustic level were probably made by Seashore, with the Tonoscope (SEASHORE, 1916), a piece of equipment which, in his own words "will transform the vibrations of voice or instrument to visual configurations on a scale that indicates the actual pitch of any note down to an accuracy of a fraction of a vibration". In the subsequent years, in collaboration with other researchers, he studied the vibrato in terms of pitch and amplitude modulation, in several publications that are summarized in (SEASHORE, 1938). Figure 6 shows an example of the resulting pitch and intensity measurements obtained for a violin recording in a chart that the author calls "the violin performance score".

After Seashore's retirement, there was a hiatus in music performance research, until its resurgence, around the 1960's. In that decade, the Swedish musicologist Ingmar Bengtsson started to study musical rhythm and was soon joined by Alf Gabriellson (GABRIELSSON, 1999). They conducted studies on Viennese waltzes and Swedish folk tunes, employing a technique called *oscillogram filming* to extract timing information directly from the sound (BENGTSSON; GABRIELSSON; THORSRN, 1969; BENGTSSON, 1974; BENGTSSON; GABRIELSSON, 1977). For the waltzes, they identified systematic timing deviations for the three beats in a measure, following a "short-long-intermediate" pattern, with the anticipation of the second beat. Systematic timing deviations were also identified in the Swedish folk tunes studied. Povel (1977) used gramophone recordings

³ It consisted of a piano with a series of optical shutters connected to its hammers and pedals, which would expose a moving film to light during their movement. The system enabled measuring note onset, offset, and intensity.

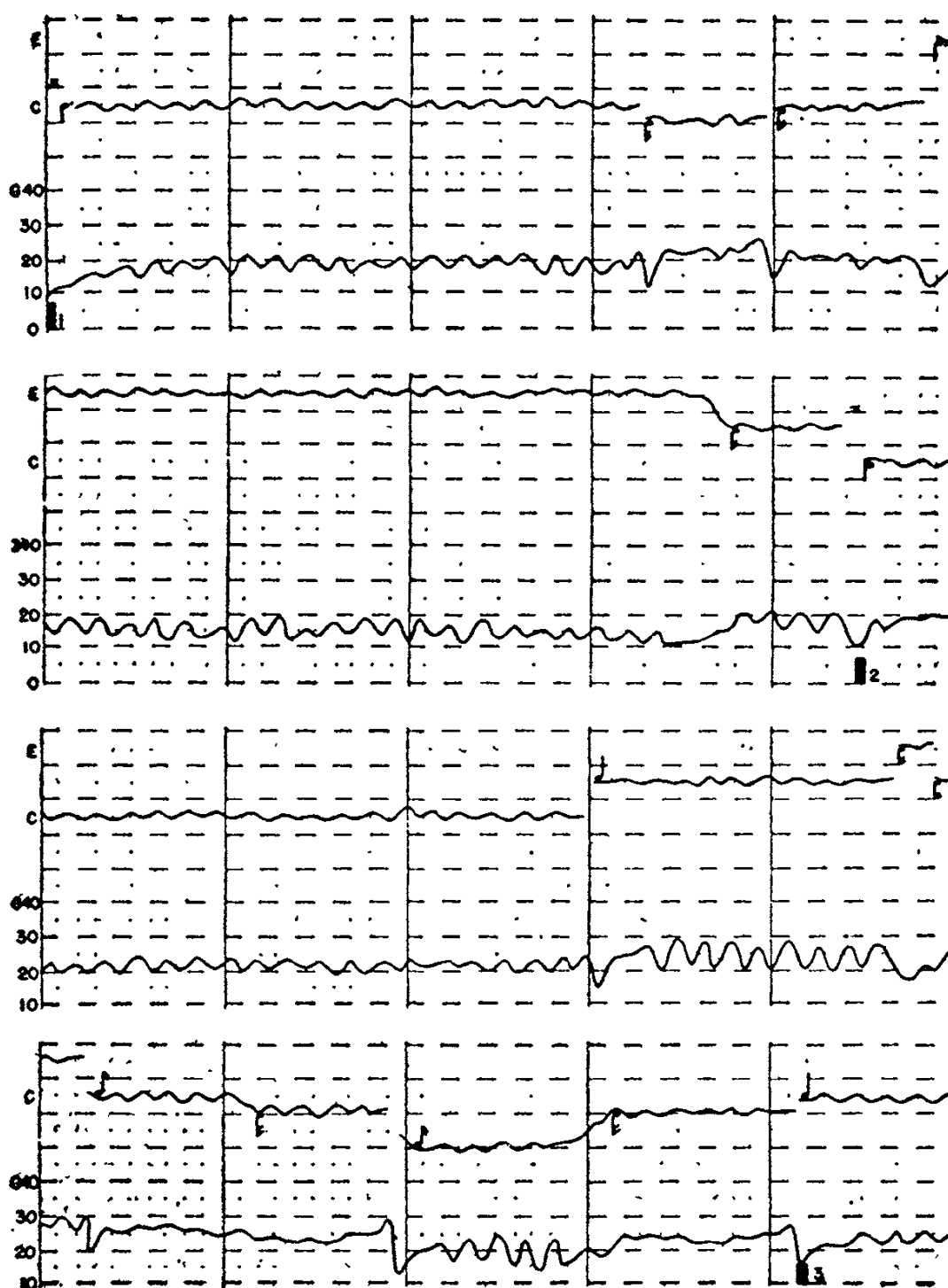


Figure 6 – Pitch and intensity measurements for an excerpt from Ave Maria (Schubert-Wilhelmj) played on the violin. The upper curve in each chart corresponds to the pitch, and also display the nominal durations (according to the score) using musical notation symbols. The bottom curve displays the dynamics of the notes. Extracted from (SEASHORE, 1938).

to study the temporal manipulations performed by three professional musicians on Prelude No. 1, Book 1, in C Major, BWV 846 by J. S. Bach, employing oscillograms to annotate the onset of the notes “by eye”. He reported that he was able to determine the note onsets in the recordings with great precision, within 1-2 ms. Bengtsson and Gabrielsson (1980), Gabrielsson, Bengtsson and Gabrielsson (1983) employed a specialized equipment for analyzing monophonic audio sequences, to compare systematic temporal variations in different recordings of monophonic performances of several melodies on the flute, clarinet, and piano. The note onsets were manually estimated using the output of the equipment, comprised of pitch and energy curves. They reported a measurement precision of ± 5 ms. These studies did not mention any criteria for estimation of the measurement error.

During the 1990’s Bruno Repp conducted a series of studies on piano performances, investigating different aspects of the performance on the instrument. In some of these studies (REPP, 1990; REPP, 1992; REPP, 1998; REPP, 1999) he analysed the timing of commercial recordings from several pianists, and the measurements were performed manually, by means of a computer interface. In (REPP, 1990, p. 252), he performed the annotations for one of the studied recordings twice, and used them to estimate the error. He reported a mean absolute discrepancy of 12 ms between the estimated onsets. In (REPP, 1992), the author identifies differences and commonalities in the expressive temporal manipulation patterns executed by several pianists on the piece *Träumerei*, *Kinderszenen* op. 15, No. 7 by Robert Schumann. He demonstrates common patterns that are related to structural characteristics of the composition, and differences that are attributed to individual variation. The idiosyncratic temporal characteristics of the performances from two legendary pianists, Alfred Cortot and Vladimir Horowitz, are objectively demonstrated in the article.

3.3.3 Studies of Clarinet Performance

Most empirical studies on clarinet performances rely on the analysis of timing manipulations on the instrument, requiring a method for estimation of note onsets. Barthet et al. (2010) analysed mechanical and expressive performance excerpts from Bach’s *Suite No. II* and Mozart’s *Quintet for Clarinet and Strings* and investigated the effect of the performer’s expressive intentions on acoustic parameters such as timing, timbre, dynamics and pitch. They observed a strong effect of the expressive intention on timbre, timing and dynamics. In a subsequent study, Barthet et al. (2011) investigated the effects of the same acoustical parameters of listener’s preferences, using an analysis-by-synthesis to transform the previously recorded performances.

Mota (2012) studied the synchronization of five acoustic parameters and its rela-

tionship with the musician's body movement. These parameters included normalized inter-onset interval (IOI), logarithm of the attack time and legato index, all of which relied on the estimation of the note onset times. The calculated acoustic parameters were compared to data extracted using a motion capture system. The results showed that the participants tend to synchronize the acoustic parameters better when playing along their own recordings than recordings of other musicians. In (MOTA, 2017), the author investigates further the human-to-human interaction and synchronization in ensemble performance using excerpts played in unison by two clarinetists. The note onset times are used for determining the synchronization accuracy between two clarinetists, a leader and the follower. One of the hypotheses investigated by the study was whether rhythmic consistence (from a leader clarinetist) influence the quality of the synchronization between two musicians. The study showed evidence that point towards this hypothesis, although it mentions that the results were not conclusive.

Additionally to timing manipulations, (TEIXEIRA; LOUREIRO; YEHIA, 2018) used the audio RMS and Spectral Centroid (see sections 7.2.3.2 and 7.2.3.6) as a rough estimate of loudness and timbre manipulations, to investigate possible correlations between such data and the clarinetist's movement. They found evidence that supports "the hypothesis of a musical significance in the ancillary gestures of musicians, closely related to their [the musicians'] sounded expressive intentions".

All these studies employed semi-automatic techniques for note onset detection, which normally require manually fixing a large percentual of the annotations generated automatically by a method. Mota (2017) makes the following remark regarding the importance of accurate measurements of note onsets:

One of the key points for studying the synchronization in musical ensembles is the ability to detect note onsets with accuracy. To the present, analysis of timing and synchronization in musical performance studies has been limited by the precision of note detection methods. [...] [I]n recent years, the majority of works dealing with synchronization are tied to the use of MIDI interface devices (e.g. electronic keyboards) leaving aside acoustic instruments and voice.

These remarks emphasize the importance of reliable and temporally accurate onset detection methods for the empirical study of musical performances on the clarinet.

4

Machine Learning

Machine learning algorithms have become widely adopted over the past few decades. They are now present in virtually every field, handling a wide range of different tasks, such as credit approval, fraud detection, image classification, speech synthesis, medical diagnosis, protein folding, among many others. It concerns algorithms that enable computers to learn from input data instead of being explicitly programmed to perform a given task. The following definition was proposed by Mitchell (1997), “[a] computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks T , as measured by P , improves with experience E ”. To solve a given task using a machine learning algorithm, one must provide some input data from which the program will learn and define a performance measure to evaluate how good the model’s predictions are.

4.1 Deep Learning

Over the last decade, a class of techniques called deep learning became a game-changer, solving problems that conventional machine learning techniques could not handle, or at least, were not so successful. Although the term deep learning is frequently associated with artificial neural networks with multiple layers, it is not restricted to learning techniques that are neurally inspired. “It appeals to a more general principle of learning multiple levels of composition” (GOODFELLOW; BENGIO; COURVILLE, 2016).

Deep learning allows computational models that are composed of multi-

ple processing layers to learn representations of data with multiple levels of abstraction. [...] Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. (LECUN; BENGIO; HINTON, 2015)

Before the advent of deep learning, most techniques used in music information retrieval were based in two stages: (1) feature extraction to transform audio into a more meaningful representation (usually music features that represent timbre, rhythm, harmonicity, dynamics and other kinds of information); followed by (2) a classifier, trained to perform the desired task (DIELEMAN; SCHRAUWEN, 2014). Now the approach has shifted towards strategies that substitute the handcrafted feature extraction by learned features, since the most relevant aspects of the feature extraction can be optimized by a neural network, based on the objective of the task being learned.

4.2 Artificial Neural Networks

Artificial neural networks are directed graphs in which each node corresponds to a neuron (also called unit) and the edges correspond to the connections between neurons. Each connection between neurons has a weight associated to it, which is a parameter learned during the training of the network. The most basic type of neural network is called feedforward neural network, or multilayer perceptron. In this type of network, each neuron consists in a weighted sum of inputs, with an added constant value called bias. One layer of such network contains multiple neurons, each one having their own weights and biases. Also, the value output by each neuron is usually transformed by an activation function, which is usually a nonlinear function. In this type of network, the outputs of a layer will connect only to the inputs of the next immediate layer, without any feedback connections. Also, a neuron does not have the ability to store any internal state, so its output depend only on its current input. Feedforward neural networks define a basis from which several different types of neural networks were developed.

At the beginning of the training process, the parameters of the network are set according to the initialization strategy, which can be, for example, assigning them a random value according to a gaussian distribution. The objective of the learning algorithm is to learn a set of parameters that enables the network to produce the desired outputs, based on the given training examples. To achieve this, the outputs of the network will be estimated for each training example, and its loss (or error) will be calculated based on the values of the target and the prediction, using a previously

chosen loss function. This error will be propagated backwards through the network, and will be used by the optimizer to update the values of the parameters of the network.

4.3 Supervised Learning

Supervised learning algorithms learn to perform a given task based on patterns extracted from the input $x^{(m)}$, and its associated labels $y^{(m)}$ which describe the expected output of the model, for each training example m in a training set of size M . A supervised learning algorithm analyzes the input training data to learn a mapping $h_{\theta}(x^{(m)})$ that approximates to the expected output $y^{(m)}$. After the training, this function can be used to map new unseen examples. The labels are frequently difficult to collect automatically and require a human annotator, but the term supervised learning also applies when they are collected using an automatic method (GOODFELLOW; BENGIO; COURVILLE, 2016).

4.4 Cross-Entropy Loss

A loss function measures how far a model's predictions are from the target values. Typically, during training, the learning algorithm will iteratively update the neural network parameters, seeking to minimize the loss. Choosing the appropriate loss function for a given problem is crucial for obtaining satisfactory results.

The cross entropy is a convex lost function which is a frequent choice for calculating the loss in classification problems. In a binary classification problem, for each training example m in a training set of size M , it is defined as

$$J(\theta) = -\frac{1}{m} \sum_{m=0}^{M-1} y^{(m)} \log(h_{\theta}(x^{(m)})) + (1 - y^{(m)}) \log(1 - h_{\theta}(x^{(m)})), \quad (4.1)$$

in which $x^{(m)} \in \mathbb{R}^D$ is a D-dimensional input vector, $y^{(m)} \in \{0, 1\}$ is the expected output (target), and $h_{\theta}(x^{(m)}) \in \{x \mid 0 < x < 1\}$ is the actual output produced by the model using its current internal parameters.

For multi-class classification, where the number of classes K is larger than 2, the model produces K outputs, each corresponding to the probability of one of the classes. In this case, we modify the notation slightly, adding a subscript k to the targets and outputs. So $y_k^{(m)} \in \{0, 1\}^K$ is a vector with the expected (target) outputs $(y_0^{(m)}, y_1^{(m)}, \dots, y_{K-1}^{(m)})$; and $h_{\theta}(x^{(m)}) \in \{x \mid 0 < x < 1\}^K$ represents the actual outputs produced by the model. In

this scenario, a separate loss is calculated for each class label and then summed up:

$$J(\theta) = -\frac{1}{m} \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} y_k^{(m)} \log(h_{\theta,k}(x^{(m)})). \quad (4.2)$$

Since frequently in multi-class classification it is important that outputs of the model sum up to 1.0, it is common to normalize the outputs using the function softmax:

$$\sigma(z_k) = \frac{e^{z_k}}{\sum_{j=0}^{K-1} e^{z_j}}, \quad (4.3)$$

where z_k for correspond to the z -th output of the model.

4.5 Gradient-Based Optimization

During the training process, a machine learning algorithm updates the model's internal parameters, typically seeking to minimize the loss produced by the model. In a neural network, these parameters are the weights and biases of the network. The optimizer algorithm is the key piece in this process, since it will define the updating criteria for each parameter of the network, based on the loss produced on the output of the model, the current parameters of the network and the hyperparameters of the optimizer itself. The hyperparameters must be previously chosen by the programmer, before training the model, to control some aspect of the behaviour of the learning algorithm. For example, the learning rate is a scalar hyperparameter that controls how large is the update performed to the model parameters, in each update step. In simpler terms, it controls the speed of the learning process.

In gradient-based optimization, the learning process occurs by estimating the impact of small variations in the model parameters on the loss. This estimation is made by calculating the gradient of the loss function with respect to the model parameters (LECUN et al., 1998, p. 3). In each iteration, the parameters θ of the model are updated according to the formula

$$\theta \leftarrow \theta - \alpha \frac{\partial J(\theta)}{\partial \theta} \quad (4.4)$$

in which α is the chosen learning rate.

In a neural network, the parameters θ that will be updated correspond to the weights and biases of each neuron in the network. The update in those parameters are performed by the backpropation algorithm, which first updates the parameters of the last layer in the network and calculates the error, which is subsequently propagated to

the preceding layer, to update its parameters. This process is repeated in a layer-wise fashion, until all the parameters of the network have been updated.

In batch gradient descent, the parameters of the network are updated using the calculated gradient for all the training examples in the train set, in a single step. The epoch¹ and the step have the same duration. In stochastic gradient descent (SGD), the examples from the training set are randomly presented, and the update rule uses the calculated gradient for each train example. The number of steps in a single epoch correspond to the number of training examples in the train set. Mini-batch gradient descent, or mini-batch stochastic gradient descent, is a compromise between the two approaches, and also consists in randomly presenting the training examples, but the updates are not so granular. Instead of calculating the gradient for each individual example, it calculates it for a mini-batch of examples of a predetermined size. Mini-batch SGD mitigates the typical noise in SGD, but is more computationally efficient than batch gradient descent.

4.6 Long Short-Term Memory Networks

The output of a Recurrent Neural Network (RNN) does not depend only on its current input, but also on the state of the network in previous time steps. Long Short-Term Memory (LSTM) networks are a special type of RNN capable of learning long-term dependencies (standard RNNs are quite limited in the range of context they can access). LSTMs are very powerful for dealing with sequential data like text, speech, video or music.

The data flows through the layers of an LSTM, interacting with its internal state/memory. The LSTM can add or remove information to the memory using structures called gates. There are three gates in a LSTM: the *forget gate* uses a linear transformation between the current input and the output from the last time step, followed by a sigmoid activation function, to remove or keep information in the memory cell. The *input gate* also uses a transformation (linear + sigmoid) to decide which values will be updated in the memory cell, while a third transformation (linear + tanh) creates a vector of values that could be added to the memory cell. These two transformations are combined (through element-wise multiplication) and added to the memory. Lastly, the *output gate* uses a transformation (linear + sigmoid) to choose what information from the memory the LSTM will output.

¹ Each epoch corresponds to one complete pass through the entire training dataset.

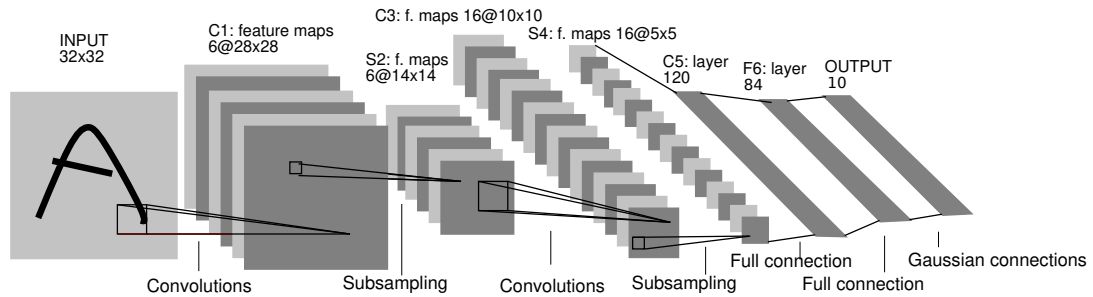


Figure 7 – A Convolutional neural network applied to an image classification task (digits recognition). Extracted from (LECUN et al., 1998).

4.7 Convolutional Neural Networks

Convolutional neural networks were introduced in the 1990's but only recently they became more widespread, due to a work in which deep CNNs outperformed the state of the art in an image classification challenge (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). CNNs are very powerful when applied to types of data for which topological relations should be taken into account, such as images and audio². The discrete convolutions employed in CNNs consist in a type of linear transformation that preserve this notion of ordering. CNNs are sparse (only a few input units contribute to a given output unit) and reuse parameters (the same weights are shared in multiple locations of the input) (DUMOULIN; VISIN, 2016).

Convolutions consist in applying kernel filters (sliding windows) to data. In the case of images, two-dimensional kernels are usually applied (for example, a 3×3 kernel multiply its values element-wise with the original image, and then sum them up), and the operation is repeated by sliding the window over the whole matrix. Figure 7 shows an example of a typical CNN architecture for image classification. For each layer of a CNN, convolutions are applied to the input, followed by a nonlinear activation function. Multiple kernel filters can be used in each layer of the network, generating feature maps as a result of the convolution operations. The CNN will learn the parameters for each kernel during the training phase based on the objective of the network. Max-pooling layers (subsampling) are usually applied after convolutional layers, in order to reduce dimensionality without losing the most salient information.

4.8 Rectified Linear Units

A rectified linear unit (ReLU) is a non-linear activation function, which is defined as $f(z) = \max(0, z)$, and has recently become widely used in neural networks. The

² Pixel ordering is important in images, while temporal information is important in audio.

difference between them and a linear unit, is that it outputs zero for half of its domain. They have been widely adopted because they are a good alternative to sigmoid and tanh activation functions, one of the reasons being the fact that they avoid a problem called vanishing gradient, which hinders the learning process in neural networks.

4.9 Dropout

There are many different regularization techniques in machine learning, which are useful to avoid overfitting. Overfitting happens when a model with high capacity captures the variation of the data as if that variation represented the essence of the data. What will frequently happen in this situation is that the model will not be able to generalize well to unseen examples, thus performing significantly worse on new data than it did on the data used for training.

Dropout is a powerful stochastic regularization strategy that can be applied to a broad family of models. It consists in randomly omitting each unit of the model with a given probability. “This prevents complex co-adaptations in which a feature detector is only helpful in the context of several other specific feature detectors. Instead, each neuron learns to detect a feature that is generally helpful for producing the correct answer given the combinatorially large variety of internal contexts in which it must operate” (HINTON et al., 2012).

4.10 K-Fold Cross Validation

To validate the results of a trained model, it is important to have a separate validation set, containing data that was not used during the training of the model. This is because the model will learn from patterns in the training data, and its ability to produce good results with new data (in other words, to generalize) has to be evaluated using data it has not seen before. However, when the volume of data available for training is relatively small, the quality of the resulting model tends to degrade, so reducing the training data even further might become unfeasible. At the same time, to have a reliable estimation of how the model will behave with unseen data, it is important to have a validation set with a sufficient number of samples. A technique called k -fold cross validation handles this problem by splitting the entire dataset into k subsets (folds) and repeating the training process k times. Each time, a different fold is chosen as the validation set, and the resulting models are evaluated separately. This avoids sacrificing samples of the training set to obtain a reliable evaluation of the model. However, it is necessary to train the model k times, increasing the total training time.

4.11 Evaluation Metrics

Choosing the appropriate evaluation metric is a critical step to assess the performance of the model. In a binary classification task, the classification model must predict if a given example belongs or not to a single given class³. If it does, it is a *positive* example, and if not, a *negative* example. There are four possible outcomes for a prediction made by the model:

- Correct predictions
 - true positive (*TP*);
 - true negative (*TN*).
- Incorrect predictions
 - false positive (*FP*);
 - false negative (*FN*).

The most basic metric for classification tasks is the accuracy, which consists in dividing the number of correctly predicted examples by the total number of examples.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.5)$$

But when dealing with unbalanced datasets⁴, the accuracy is unreliable as an evaluation metric because it may be biased towards the overrepresented class. In such cases, the metrics *recall* and *precision* are frequently used in conjunction, since they are less sensitive to class imbalance. Recall measures the ratio between the correctly predicted positive examples and the actual number of positive examples (based on the ground truth).

$$\text{recall} = \frac{TP}{TP + FN} \quad (4.6)$$

Precision measures the ratio between the correctly predicted positive examples and the total number of examples predicted as positive.

$$\text{precision} = \frac{TP}{TP + FP} \quad (4.7)$$

³ A simple and classic example: a classification model that has to decide if any given photograph is a photo of a cat or not. So the class in the case is *cat*. A *positive* prediction means the example is classified as a cat, while a negative example as not a cat.

⁴ When one or more target variables (classes) are overrepresented in a dataset (in other words, when the class distributions are not uniform), the dataset is said to be unbalanced. This problem is also known as a class imbalance.

If it is important for the model to reach a good balance between precision and recall, the F1-score can be used, since it summarizes the performance of the classifier with a single number (GOODFELLOW; BENGIO; COURVILLE, 2016). It consists in the harmonic mean of the two.

$$f1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (4.8)$$

5

Note Onset Annotation

The automatic detection of note onsets in audio recordings is a difficult task, and frequently the methods available might not be robust enough to rely upon. So in some circumstances it might be necessary to resort on manual annotation. But given the current state of machine learning, wouldn't it be possible to develop automatic models of audio segmentation that are as robust as manual annotations? In any case, to obtain robust segmentation models using machine learning, it is necessary to provide a dataset with labeled target values, because this is a typical task for supervised learning algorithms. In this chapter we will discuss the process of manual annotation of two clarinet onset datasets created specifically for this study. We will also take the opportunity to assess the annotation error for the task. In chapter 6 we will use those datasets for training/evaluating a neural network for clarinet onset detection.

A few prominent studies have employed manual methods to annotate note onsets in recordings (BENGTSSON; GABRIELSSON; THORSRN, 1969; POVEL, 1977; GABRIELSSON; BENGTSSON; GABRIELSSON, 1983; REPP, 1990; REPP, 1992), however the measurement error is barely discussed, if at all. Repp (1990) and Repp (1992) make rapid analyses of the measurement error in piano recordings but does not delve further on the subject. This scenario led us to question whether manual onset annotations are precise enough for researchers studying musical performances, specially on the clarinet, since its note attacks can be very soft, making it harder to define the instant of the onset. We only found a single study that sought to analyze note onset annotation error (DAUDET; RICHARD; LEVEAU, 2004). It comprised a few different instruments, which reported an average timing difference of 10.5 ms among annotations performed by three subjects, in both monophonic and polyphonic audio. One of the excerpts in the study was a clarinet recording of 30 seconds, for which the average timing difference

was slightly larger, 13.6 ms.

As mentioned by Cartwright et al. (2017, p.1), “[a]udio annotation is key to developing machine listening systems”. They conducted a study that investigated the effects of visualizations in the quality of audio annotations for soundscapes. Their results showed that “the spectrogram visualization enables annotators to identify sounds more quickly”. They proposed a nice tool for audio annotation which we considered adopting for our experiments. However, it was not suitable for our specific needs, because it did not provide an option to set a fixed resolution (in milliseconds per pixel) for the audio visualization. The interface would resize the visualization to occupy all the available space on the screen, which would lead to different resolutions depending on the length of the audio file, compromising the necessary precision for annotating note onsets.

Therefore, we decided to develop our own tool, called *Audio Segment Annotator*, which consists in a web application for multi-categoric annotation of points and segments, with an integrated administration interface for experiment management. This tool seeks to provide a fine temporal resolution of 2 milliseconds for annotating audio, in a user interface with a steep learning curve and straightforward controls. The draggable screen elements and few keyboard shortcuts are easy to learn; in fact, for a user that is already familiar with audio-related software, it takes no more than two or three minutes to learn how to use the application. It provides visualizations of the audio waveform and its spectrogram, synchronized with respect to the time axis. The fact that it is developed as a web application makes it suitable for crowdsourcing, enabling researchers to obtain large corpuses of annotations in a relatively short time. See appendix A for more details.

As mentioned in section 3.2, there are a few different definitions of note onset, each of them corresponding to a slightly different instant at the beginning of a note. Any attempt to determine one of those particular instants precisely involves a series of challenges. To obtain the perceptual note onset, the subject cannot be exposed to any other stimulus than the sound. The visual information provided by a waveform or spectrogram will certainly affect the judgement of the listener regarding the perceptual onset, biasing it towards the physical onset. On the other hand, any attempt to annotate the physical onset would be influenced upon the presentation of the recording itself, biasing the subject’s judgement towards the perceptual onset. We considered a few possible solutions to annotate perceptual onsets, like asking the subject to perform tapping or to push a mechanical button synchronized with the onset events, or even by playing back the audio together with controllable short click sounds that the subject would be supposed to synchronize with the note onsets. However, those approaches would pose their own methodological challenges, and would also make the annotation process a lot more time-consuming, making the annotation of larger corpuses of data

excerpt	file	length (s)	onsets
Mozart <i>Clarinet Quintet</i> Kv. 588 in A Major, 1st Movement – Main Theme (bars 118-124)	mozart_q_m1	14.52	31
Mozart <i>Clarinet Quintet</i> Kv. 588 in A Major, 4th Movement – First Variation (bars 17-24)	mozart_q_m4	16.88	53
Stravinsky – <i>Petrushka Suite – Dance Of The Peasant And The Bear</i>	stravinsky	15.43	42

Table 2 – Clarinet excerpts from the dataset *clari-onsets-3*.

unfeasible. Therefore, we decided to take a more pragmatic approach, avoiding to deal with these specific onset definitions in our annotation experiments.

5.1 Clari-Onsets-3

5.1.1 Methods

In this experiment we performed redundant onset annotations for the three clarinet excerpts listed in table 2, seeking to obtain: (1) a reliable estimation of the note onset time, based on multiple measurements; and (2) an estimation of the measurement error for the task. The three audio files contained a total of 126 note onsets. Their corresponding scores are shown in figure 8. These excerpts had been previously recorded by two different clarinetists, and were taken from a larger database from our research group. They were all recorded in our laboratory, in a room with very little reverberation. The recordings were chosen seeking to achieve a diversity of musical material, with notes in different registers of the instrument, melodies containing both short and long intervals, notes with soft and sharp attacks, and different dynamics.

Using the *Audio Segment Annotator*, we repeated the annotation procedure five times, in annotation sessions that took place in three different occasions. We obtained a total of 630 onset annotations.

5.1.2 Results

For this particular dataset, by making repeated measurements of the note onsets and calculating the median onset time for each note, we obtained a more reasonable estimation of the onset time. We were also able to estimate the error involved in this kind of measurement, by calculating the time difference between each annotated onset (from different sessions) and the median.

in A

(a) Mozart *Clarinet Quintet* Kv. 588 in A Major, 1st Movement – Main Theme (bars 118-124)

in A **Allegro**

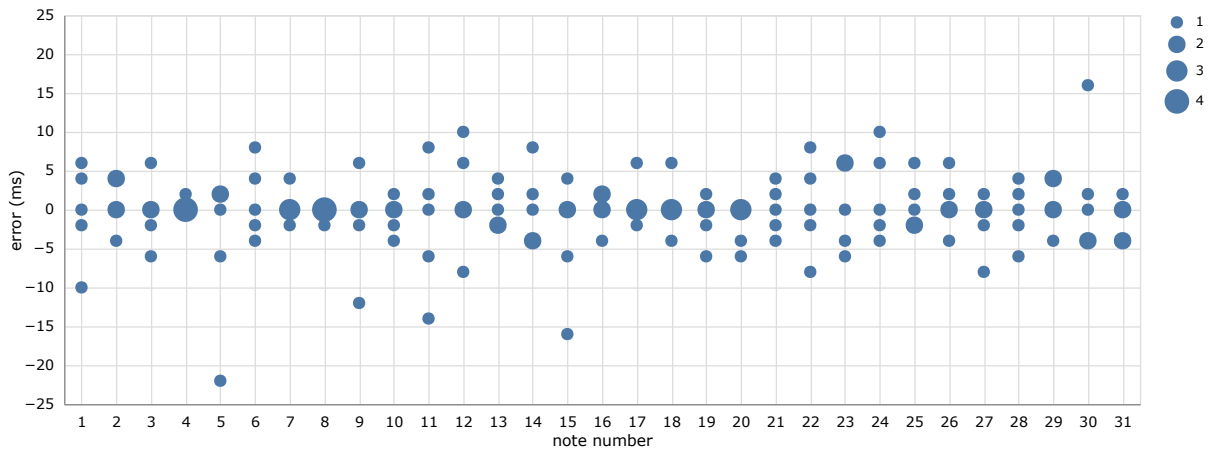
(b) Mozart *Clarinet Quintet* Kv. 588 in A Major, 4th Movement – First Variation (bars 17-24)

Sostenuto
in Sib

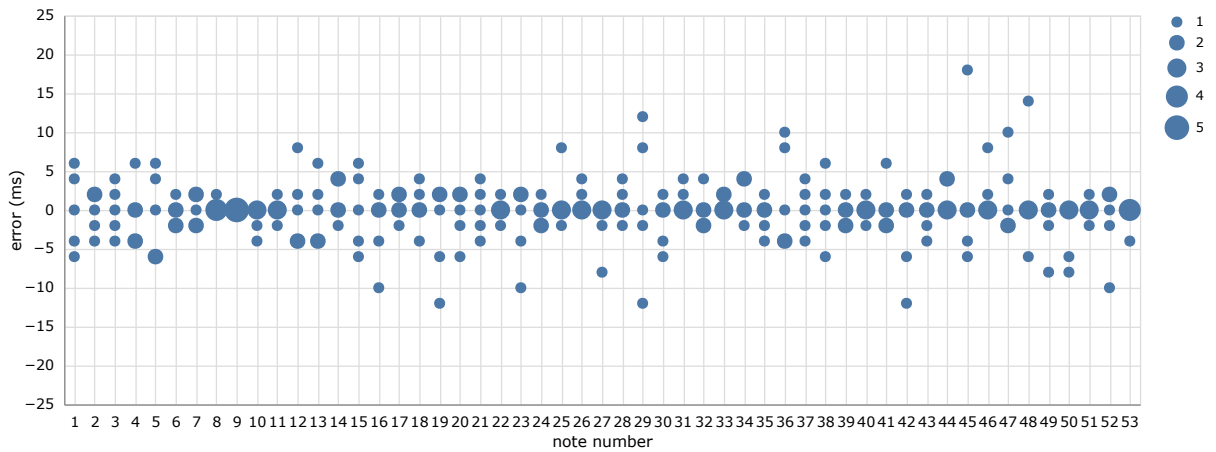
(c) Stravinsky – *Petrushka Suite* – *Dance Of The Peasant And The Bear*

Figure 8 – Scores for the three clarinet excerpts listed in table 2.

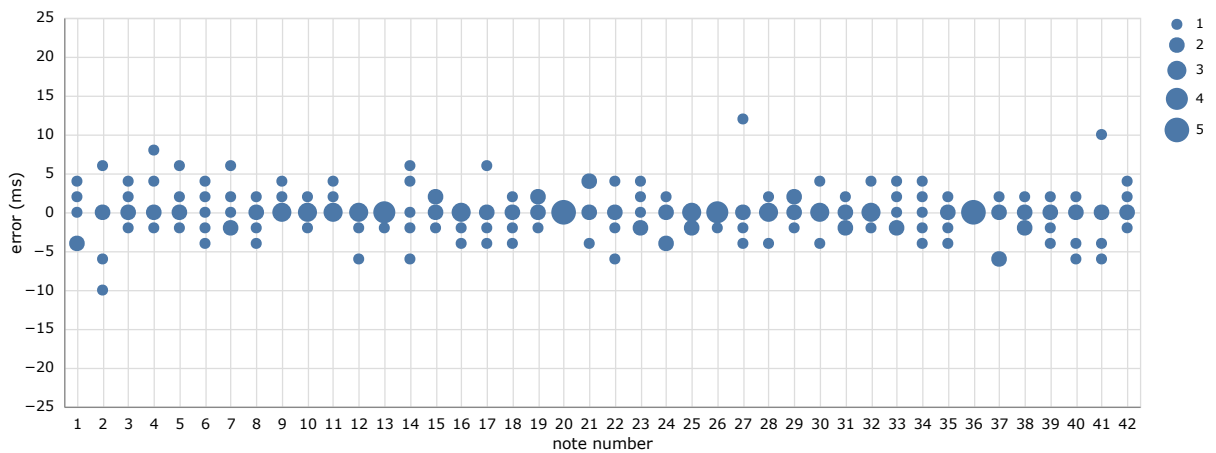
Figure 9 shows the error for each note from the three excerpts, while table 3 summarizes them, grouping by absolute error ranges. For 96.5% of the onset annotations the absolute error is less than or equal to 8 ms. The type of acoustic material in a recording affects how difficult it is to annotate the note onsets. As the difficulty increases, the magnitude of the annotation error is also likely to increase. The excerpt that showed the larger dispersion in the annotation error was `mozart_q_m1`. This is the excerpt that contains the softest attacks, which makes the precise definition of a note onset point more challenging. On the other hand, the excerpt `stravinsky` contains sharp high pitched attacks in *fortissimo*, which are considerably easier to annotate, and hence its annotation error tend to be lower.



(a) Mozart *Clarinet Quintet* Kv. 588 in A Major, 1st Movement



(b) Mozart *Clarinet Quintet* Kv. 588 in A Major, 4th Movement



(c) Stravinsky – *Petrushka Suite* – *Dance Of The Peasant And The Bear*

Figure 9 – Binned scatter plot showing the measurement error of the annotations for the three excerpts listed in table 2. The size of the circle represents the number of times a specific error value occurred, like a histogram, with bin width of 2 ms. The error consists in the difference between each onset annotation and the median.

absolute error (ms)	count	percentage	cumulative percentage
0 to 4 ms	541	85.9	85.9
4 to 8 ms	67	10.6	96.5
8 to 12 ms	16	2.5	99.0
12 to 16 ms	4	0.6	99.7
16 to 20 ms	1	0.2	99.8
> 20 ms	1	0.2	100.0

Table 3 – Absolute error for the manual annotations obtained for the dataset *clari-onsets-3*.

5.2 Clari-Onsets-50

Most datasets available for note onset detection do not focus on monophonic recordings or do not contain a significant amount of clarinet onsets. Therefore, we decided to create a larger dataset composed exclusively of clarinet recordings. This will enable us to train a specific onset detection model for clarinet (see chapter 6). This dataset is composed by 50 excerpts of solo clarinet recordings, totaling 23 minutes and 6 seconds of audio, with a mean duration of 27.7 seconds. Table 4 lists the audio files included in the dataset. Half of those recordings were made in our laboratories, while the other half was obtained from commercial recordings from several different albums and clarinetists. They comprised multiple genres, including classical, jazz, and contemporary pieces. The recordings are very diverse with respect to level of reverberation, distance of the microphone, background noise, etc. This was intentional, since a varied dataset can lead to a model that is able to generalize better. The excerpts were all annotated by myself, this time without repetition, over several weeks. The annotation sessions for this dataset started a couple of weeks after finishing the annotations for *clari-onsets-3*. As a result, a total of 3551 note onsets were obtained for the 50 excerpts.

5.3 Discussion

It is very natural to think of an onset as the precise instant when there is an increase in acoustic energy corresponding to a note start. However, based in our experience annotating onsets (aided by the visual feedback of a spectrogram and a waveform) it is worth mentioning that frequently the start of a note is highly ambiguous, and the task of defining the onset as a single point becomes quite challenging. In the clarinet, in particular, some common ambiguous situations are: (1) in a few situations of two successive notes of different pitches, the onset of the frequency components of the second note do not correspond to the point where the overall energy start to increase,

file	length (s)	onsets	file	length (s)	onsets
tumbledown_1	28.05	58	tumbledown_2	32.14	62
beethoven_1	27.35	119	beethoven_2	33.68	74
brahms_1	37.44	60	debussy_1	39.65	40
debussy_2	43.09	35	debussy_3	25.29	18
debussy_4	18.19	56	excerpt_1	18.60	93
excerpt_2	15.49	34	excerpt_3	20.47	20
fast_excerpt_1	08.84	61	fast_excerpt_2	23.38	160
inschrift_2	22.48	45	inschrift_3	38.98	56
jazz_1	15.24	69	jazz_2	29.45	133
korsav_1	16.53	51	long_notes_1	40.48	24
messiaen_1	25.05	1	mimic_1	26.01	37
mimic_2	27.95	88	mimic_3	25.14	30
mimic_4	30.95	24	mozart_1	16.79	38
mozart_2	16.35	38	mozart_3	15.90	33
mozart_4	23.59	121	preludi_1	38.45	54
preludi_2	39.69	50	prokof_1	21.46	47
scheherazade_1	29.18	72	scheherazade_2	15.07	51
scheherazade_3	14.47	97	sonata_in_f_major	41.64	72
stille_nacht_1	27.55	23	stille_nacht_2	27.80	23
staccato_exercise_1	36.76	237	staccato_exercise_2	25.88	155
study_in_g_major	29.93	136	tchaikovsky_1	36.35	29
tchaikovsky_2	44.92	34	solo_clarinet_1	33.49	42
solo_clarinet_2	29.99	63	solo_clarinet_3	23.16	22
solo_clarinet_4	25.58	126	solo_clarinet_5	36.17	160
solo_clarinet_6	26.25	157	solo_clarinet_7	39.94	223

Table 4 – Clarinet excerpts from the dataset *clari-onsets-50*.

based on the waveform visualization. Instead, the spectrogram shows that the harmonic content of the second note starts to exist before the typical increase in the waveform envelope. (2) Blowing noise precede the start of a note, and since there is no clear pitch or harmonic energy present at the time, it becomes unclear if it should be treated as the onset, or if the subsequent pitched content should be prioritized. (3) The transition between successive notes can be sometimes so smooth and subtle that it sounds like a progressive transformation, consisting in an interval, instead of a single point. This kind of situation will manifest more often in recordings with a lot of reverberation, which will increase the annotation difficulty significantly. (4) Even in single notes without any superposition of energy caused by reverberation, when the note attack is too soft, it is difficult to determine one precise point that corresponds to the instant of the note onset.

The situations mentioned above led us to consider one possible future improve-

ment for this kind of annotation experiment: instead of treating the onsets as points, they could actually be treated as time intervals, at least in situations in which the annotator is uncertain about the location of the onset. In fact, at this point we believe that treating the onset as a point is a convenient (and useful) simplification, when in fact, frequently it cannot be so precisely defined.

6

Note Onset Detection

To obtain a specific note onset detector for clarinet recordings, we implemented the neural network architecture proposed by Schlüter and Böck (2014), which consists of a CNN applied to time-frequency representations of an audio signal, and trained this network on a clarinet dataset. In the original paper, the author had trained it on a dataset containing both monophonic and polyphonic recordings played on various instruments and covering multiple musical genres. We anticipated that since it is trained on mixed data, the original model would not perform as well as it could on clarinet recordings, due to the specific characteristics of the sound produced by the instrument, which are underrepresented in the training data. To verify this, in section 6.2.1 we test and evaluate this model for clarinet recordings using the datasets we introduced in chapter 5. We also evaluate two other onset detection methods from the library (RNN and SuperFlux) in the same section. In section 6.1.4 we show the results of our implementation of the CNN model, this time training the model using clarinet recordings exclusively.

6.1 Methods

Many studies on automatic onset detection have been published since the beginning of the 2000s. The first methods used mostly DSP techniques to estimate the onsets (DUXBURY; SANDLER; DAVIES, 2002; BELLO et al., 2005; COLLINS, 2005; ZHOU; REISS, 2007; THORNBURG; LEISTIKOW; BERGER, 2007). Recently, the majority of the studies have employed machine learning techniques, which tend to achieve better results on the task. Typically, an onset detection method will attempt to generate a

time series that provides an estimation of the probability of onset over an audio signal. This time series is usually called *onset detection function* (ODF). Then, a peak-picking algorithm will be used to estimate the actual onset times from the ODF.

In sections 6.1.1 to 6.1.3 we describe three methods for note onset detection, of which the first two are based on neural networks. These methods are evaluated to detect clarinet onsets in section 6.2.1, using models trained on a mixed dataset. Our own implementation of the CNN model (section 6.1.1), trained on a clarinet dataset, is evaluated in section 6.1.4.

In this chapter we test and evaluate three onset detection models from the library *madmom* (BÖCK et al., 2016) on clarinet recordings, specifically. Two of those models are based on neural networks, a CNN (section 6.1.1) and an RNN (section 6.1.2), and were previously trained by the author of the library on a generic dataset. The other method (section 6.1.3) uses strictly DSP techniques and is based on spectral flux. We also test and evaluate a few models trained by ourselves using a clarinet dataset, specifically. These models use basically the same CNN architecture from *madmom*'s model, along with a few minor modifications, to investigate if they lead to a better result. With these clarinet-specific models we can check whether the specific training data leads to a better note onset detection model for the instrument.

6.1.1 CNN

This method was proposed by Schlüter and Böck (2014) and is based on a convolutional neural network applied to mel-scaled spectrograms. The advantage of this representation is that it uses a logarithmic scale for the frequency channels instead of a linear one. This logarithmic scale models better the human perception of pitch, allowing the spectrogram to have a better perceptual pitch resolution using a relatively small number of bins, thus reducing the model's memory demand.

The input of the model consists of three stacked 80-band mel spectrograms with logarithmically scaled magnitudes, each calculated using a different window length: 1024 (23 ms), 2048 (46 ms), and 4096 (93 ms) samples. They are calculated using a hop length of 441 samples, which corresponds to a frame resolution of 10 ms. The input of the network consists of a group of 15 contiguous spectral frames centered on the frame to be classified. So each example processed by the network corresponds to a context of 150 ms. The onset annotations (which consist of time points) are converted to target values, which are defined for each input example, using the criteria described below.

- If there is an onset annotation within the 10 ms time window corresponding to the central frame for a given input, the target value is assigned to 1 (onset).

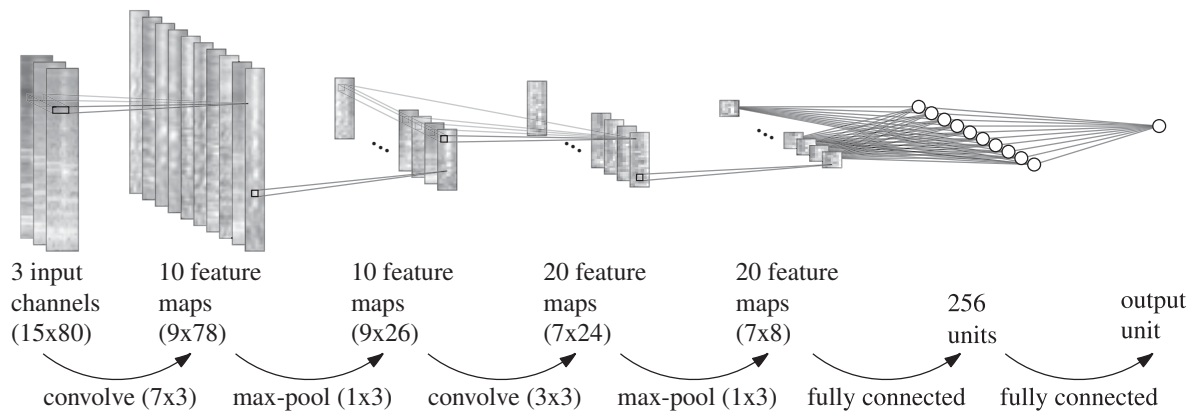


Figure 10 – Convolutional Neural Network for onset detection. Adapted from (SCHLÜTER; BÖCK, 2014).

- Since the annotation error might be a bit larger than this window, each frame immediately before or after a frame that was set to 1 in the step above is also set to 1, giving a margin for the network to learn from annotations that are not precise. These samples are also weighted with a factor of 0.25 during training.
- The targets for any other frames are defined to 0 (no onset).

Basically, the architecture of the network consists of two convolutional layers (2D convolutions) followed by an intermediary fully-connected layer, and the output layer, which is also fully-connected. Figure 10 shows an illustration of the network architecture, which is composed by the following layers:

- input: 15 frames x 80 bands;
- 2D convolutional layer: filters of 7 frames x 3 bands, computing 10 feature maps;
- max-pooling over 3 bands with no overlap;
- 2D convolutional layer: filters of 3 x 3, computing 20 feature maps;
- max-pooling over 3 bands with no overlap;
- fully connected layer with 256 units;
- fully connected layer with a single unit (output).

All the layers use rectified linear unit activation functions. Dropout is applied to the inputs of the fully connected layers for regularization. The network is optimized using gradient descent, minimizing binary cross-entropy loss. It is trained for 100 epochs, using a starting learning rate of 1.0, which is multiplied by 0.995 at the end of

each epoch. The initial momentum is 0.45, and it is linearly increased to 0.9, between epochs 10 and 20. The mini-batches used for each training step consist in 256 examples.

The network produces output values between 0 and 1, representing the frame-wise probability of onset along the audio signal. To obtain the onset points, first, this time series is convoluted with a Hamming window of 5 frames, which smooths it a bit. Then a peak-picking method is used to extract the local peaks higher than a given threshold value. This threshold is chosen by varying its value and picking the one that produces the highest F-score. The instants of the peaks correspond to the onsets detected by the model.

6.1.2 RNN

The model proposed by Eyben et al. (2010) uses as input a stacked array of features consisting of: (1) a mel spectrogram calculated using a window length of 1024 samples (23 ms); a mel spectrogram computed using a window length of 2048 samples (46 ms); (2) the positive first-order difference between two successive mel spectrogram frames at the 1024 samples window length; (3) the positive first-order difference at the 2048 samples window length. The positive first-order difference is calculated by applying a half-wave rectifier function $H(x) = \frac{x+|x|}{2}$ to the difference between two consecutive mel spectrogram bands. A total of 40 mel bands are used in this model, achieving 160 stacked input features for each time step. The input features are calculated using a sliding window with a hop length of 441 samples (10 ms). Each hop performed by the sliding window corresponds to one time step in the model.

The model consists of a bidirectional long short-term memory network (LSTM) applied to the described input features. Bidirectional recurrent networks incorporate future context into a network by adding an extra layer for each hidden layer, which will process the input sequence backward, while the other layer processes it forwards. It produces a non-causal model that outputs values that depend on both the past and future time steps. The network contains 6 hidden layers, 3 of them for processing the input forwardly and 3 backwardly.

In this network architecture, each layer contains 20 LSTM units. The network's output layer has two units, which are normalized to sum up to 1.0, using the softmax function. The outputs represent the probabilities of onset and no onset. The authors reported that they used this two-output approach because the results using a single output were not as successful.

To obtain the final onset times, a peak picking method is used to detect the local maxima just for the output corresponding to the onset class. The algorithm only detects peaks that are above a threshold θ , which is proportional to the median activations

produced in the output, constrained by $0.1 \leq \theta \leq 0.3$:

$$\theta^* = \lambda \times \text{median}(a_0(0), \dots, a_0(N - 1)) \quad (6.1)$$

$$\theta = \min(\max(0.1, \theta^*), 0.3), \quad (6.2)$$

where $a_0(n)$ is the output of the network for the onset class and n -th frame; and λ is a scaling factor chosen to maximize the F-score based on the results obtained for the validation set.

6.1.3 SuperFlux

One of the most popular DSP-based methods for note onset detection is based on calculating the spectral flux (defined in section 7.2.3.10), based on the idea that note onsets are accompanied by rapid spectral changes in the audio signal. The *SuperFlux* method, proposed by Böck and Widmer (2013), consists of adding some improvements to the spectral flux method. Instead of calculating the bin-wise difference between consecutive spectral frames, it adds a trajectory-tracking stage to the method to avoid high values in the ODF. It also combines phase information, using a technique called local group delay, to reduce the impact of amplitude variations that occur in steady tones in the output of the method. These changes seek to reduce false positives caused by *vibrato* and *tremolo*.

6.1.4 Training a CNN for Onset Detection on the Clarinet

Machine learning algorithms learn to perform tasks based on patterns that are extracted from input data. Therefore, the quality of the training dataset impacts on the quality of the final model. Having a good dataset that captures the characteristics of “real world” data is key to obtaining a model that can generalize. Although it is important to avoid biases in the dataset that may negatively impact the model’s generalization capacity, in some circumstances, a model might benefit from specific data that is biased towards the expected final purpose of the model.

We chose to implement and train the same CNN architecture mentioned in section 6.1.1 using the largest of our datasets, *clari-onsets-50*, to obtain a clarinet-specific model. We also experimented with three other slightly different versions of the same network to test which one would achieve the best performance. The experiments were all conducted using 10-fold cross-validation, using 8 folds for training, 1 for validation,

and 1 for testing. We used the validation set to optimize the models and reported the results on the test set. Those models are detailed below.

- *CNN-Clari*: uses the same network proposed by (SCHLÜTER; BÖCK, 2014), with just a few hyperparameter adjustments. The learning rate is set to 0.1, and the weights of the frames that precede and succeed the annotated onset are set to 0.4. These hyperparameters were adjusted by testing different values and evaluating the result on the validation set.
- *CNN-Clari-Gau*: uses the same network as *CNN-Clari*, but the targets are set modeling a sharp Gaussian curve in the region around the frame that corresponds to an onset, with a standard deviation of 0.5, and 5 frames of length.
- *CNN-Clari-BN*: adds batch normalization layers after the convolutional and fully connected layers of the network used in *CNN-Clari*. Batch normalization “is a method of adaptive reparametrization” which seeks to “standardize only the mean and variance of each unit in order to stabilize learning”.
- *CNN-Clari-Hi*: this model uses 120-band mel spectrograms as input, instead of the 80 bands used in the other models, and employs a network with a larger capacity than the former ones, containing three convolutional layers instead of two, and two fully connected layers before the output layer, instead of just one. It also increases the number of feature maps in the convolutional layers, and all the convolutional layers are followed by a batch normalization layer.

Our motivation for the experiment *CNN-Clari-Gau* was to test whether modeling the onset targets as sharp Gaussians would result in an improvement in the model recall, based on the fact that manual onset annotations are prone to temporal errors that might be larger than the chosen frame resolution of 10 ms. This Gaussian would extend the margin for annotation errors a bit further than just up to the neighbour frames, so we wanted to check if this approach would help the model to learn, and improve the final results. The experiment *CNN-Clari-BN* just added batch normalization layers to the network. We wanted to investigate if this technique could help the learning process a bit further. The last experiment, *CNN-Clari-Hi* tested if a model with a higher capacity and higher input dimensionality could improve the final result of the model.

6.1.5 Evaluation Criteria

To evaluate a note onset model, we need to define criteria to decide whether each onset prediction is correct or not. Thus, the predictions must be compared and matched to the onset annotations (ground truth). For this purpose, we specified an argument ω ,

which is a tolerance window (defined in milliseconds) centered on each annotated value. To be considered valid, a prediction must fall within the defined tolerance window around the onset annotation. The annotated onset times are matched to the predictions using the following criteria:

1. first we construct a distance matrix $\Delta_{a,p}$ containing all possible distances between each annotation a and prediction p ;
2. we search this matrix and find the smallest distance $\delta = \min(\Delta_{a,p})$ between an annotation and a prediction, and set the corresponding indexes to a' and p' ;
3. if δ is smaller than $\omega/2$, the annotation a' and prediction p' will be marked as a valid match (true positive), and the row a' and column p' will be discarded from the distance matrix;
4. if there are still distances smaller than $\omega/2$ left in the table, steps 2 and 3 are repeated; else, the search for matches is finished.

At the end of the process described above, we mark all the annotations and predictions that were not matched as false negatives and false positives, respectively. Using this criterion, when two onset predictions fall within the tolerance window defined for a single annotation, only the closest one is accounted as a true positive, while the other is considered a false negative.

With those values, we can calculate the metrics precision, recall, and F-score (section 4.11), to evaluate the results of an onset detection method. All the results presented in this chapter show evaluation metrics calculated using two different values for the tolerance window ω : 20 ms and 50 ms. Since these tolerance windows are centered on the target onset point, an onset detection that is within their limits will correspond to a maximum distance of 10 ms and 25 ms, respectively.

The methods discussed in this chapter are evaluated using the two clarinet datasets we created: *clari-onsets-3* (section 5.1) and *clari-onsets-50* (section 5.2). Although *clari-onsets-3* is a small dataset, its temporal precision is considerably better, because the final targets are estimated using the median values from repeated annotation experiments.

method	$\omega = 50$ ms			$\omega = 20$ ms		
	f-score	recall	precision	f-score	recall	precision
CNN-Böck	0.861	0.851	0.872	0.605	0.597	0.614
RNN-Böck	0.781	0.779	0.784	0.459	0.456	0.461
SuperFlux	0.718	0.669	0.775	0.379	0.353	0.410

Table 5 – Benchmark results for the dataset *clari-onsets-50*.

method	$\omega = 50$ ms			$\omega = 20$ ms		
	f-score	recall	precision	f-score	recall	precision
CNN-Böck	0.948	0.944	0.952	0.709	0.706	0.712
RNN-Böck	0.887	0.905	0.870	0.685	0.698	0.672
SuperFlux	0.862	0.889	0.836	0.469	0.484	0.455

Table 6 – Benchmark results for the dataset *clari-onsets-3*.

6.2 Results

6.2.1 Benchmarking

The methods mentioned in sections 6.1.1 to 6.1.3 were employed to extract the onsets from the datasets *clari-onsets-3* and *clari-onsets-50*. The implementations used in these benchmark experiments were obtained from the library *madmom* (BÖCK et al., 2016). The author of the library previously trained the CNN and RNN methods on a mixed dataset, known as the Böck dataset, containing 102 minutes of audio and 25,927 onsets composed by multiple instruments and musical genres, with both polyphonic and monophonic recordings. We will refer to the CNN and RNN methods trained on the Böck dataset as *CNN-Böck* and *RNN-Böck*.

The results for the three methods on the dataset *clari-onsets-50* are shown in table 5. CNN-Böck performed better than the other methods with respect to all the metrics. It achieved an F-score of 0.861 for a tolerance window of 50 ms. Considering that the F-score reported in the original paper using a mixed dataset was 0.903¹ (SCHLÜTER; BÖCK, 2014), the model showed a reasonable generalization capacity, even though the result was a bit lower. For the tolerance window of 20 ms, the results obtained were considerably lower for all the methods. The SuperFlux, in particular, showed the most drastic degradation in the F-score: from 0.718 in the mixed dataset to 0.379 in the clarinet dataset.

For the dataset *clari-onsets-3*, the results obtained for every model were consid-

¹ It is worth mentioning that the results in the original paper the results were obtained using 8-fold cross-validation.

erably better, which indicates that the examples in this dataset were easier to classify. Again, CNN-Böck performed better than the other methods with respect to all the metrics, achieving an F-score of 0.948 for a tolerance window of 50 ms. This time the F-score was higher than the value reported in the original paper for the mixed dataset (0.903).

The results obtained by the CNN model in both datasets were far superior to the other methods. In the original paper that proposed the model, Schlüter and Böck (2014) also reported better results using the CNN on their dataset, but the difference between the models is more subtle than the differences we observed.

6.2.2 Results for the Clarinet-Specific Model

The results obtained with each of the clarinet-specific models (section 6.1.4) for the dataset *clari-onsets-50* are shown in table 7. This table reports the mean metrics obtained from 10-fold cross-validation. The best result was obtained using *CNN-Clari*, resulting in an F-score of 0.954 for a 50 ms tolerance window and 0.720 for 20 ms. The F-score for the model *CNN-Clari-BN* was just marginally worse. In terms of precision, this was the best model, for both tolerance windows. The results obtained for the models *CNN-Clari-Gau* and *CNN-Clari-Hi* were slightly worse, achieving F-scores of 0.947 and 0.949, respectively.

As shown in table 8, the results obtained for the dataset *clari-onsets-3* were considerably better than for *clari-onsets-50*, similarly to the results obtained with the benchmark models. These results were obtained from the final models, which we trained using the entire dataset *clari-onsets-50*. For this particular dataset, the model *CNN-Clari-BN* reached the highest F-scores for the 50 ms tolerance window (0.996). However, the model *CNN-Clari* performed better for the 20 ms window (0.908). The model *CNN-Clari-BN* reached a recall of 1.000 for the 50 ms window but dropped to 0.873 for the 20 ms one.

It may be mentioned that we did experiments with several hyperparameter combinations for the four models reported in this section. We chose the hyperparameters employed in these models based on the best F-score obtained for the validation set. We also experimented with a few different deeper network architectures, but they all tended to overfit and did not produce better results than *CNN-Clari*.

The results obtained using *CNN-Clari* on the dataset *clari-onsets-3* were uploaded to <https://taironemagalhaes.github.io/phd-thesis-audio/onset-detection/>. We made them available as audio files, in which the clarinet recordings are played along with short clicks corresponding to the note onset predictions generated by the model.

method	$\omega = 50$ ms			$\omega = 20$ ms		
	f-score	recall	precision	f-score	recall	precision
CNN-Clari	0.954	0.946	0.962	0.720	0.712	0.728
CNN-Clari-Gau	0.947	0.938	0.958	0.717	0.708	0.727
CNN-Clari-BN	0.953	0.942	0.965	0.719	0.709	0.730
CNN-Clari-Hi	0.949	0.939	0.959	0.712	0.702	0.722

Table 7 – Results obtained for the clarinet-specific models on the dataset *clari-onsets-50*, using 10-fold cross-validation.

method	$\omega = 50$ ms			$\omega = 20$ ms		
	f-score	recall	precision	f-score	recall	precision
CNN-Clari	0.988	0.984	0.992	0.908	0.905	0.912
CNN-Clari-Gau	0.984	0.984	0.984	0.881	0.881	0.881
CNN-Clari-BN	0.996	1.000	0.992	0.870	0.873	0.866
CNN-Clari-Hi	0.984	0.992	0.977	0.882	0.889	0.875

Table 8 – Results obtained for the clarinet-specific models on the dataset *clari-onsets-3*

6.3 Discussion

Training a CNN model on a clarinet-specific dataset, we were able to achieve a result that is significantly better than with a model trained on a mixed dataset. The modifications we tested to the CNN model seeking to improve the model performance did not improve the results obtained with the original model, although the model with batch normalization layers surpassed it on a few metrics.

We noticed that the clarinet-specific onset detection models tended to miss the onset for notes with extremely long attack times (soft notes), producing false negatives. In the annotation experiments (chapter 5) we reported that it was more difficult to determine the exact position of the onset for such notes. We suspect that those false negatives might be related to the annotated data’s imprecision for soft notes. Also, in a few situations where there were two consecutive notes with the same pitch, the model tended to miss the second note’s onset.

In the clarinet datasets used, there is a significant amount of breathing sounds, background noises, crackling sounds from the chair, and sometimes even whisper sounds. It is interesting to notice that the network was able to learn to ignore those sounds completely, which is an excellent result, since they would not correspond to note onsets in the vast majority of the clarinet repertoire (although they might, in contemporary compositions).

Regarding the dataset *clari-onsets-3*, for which we obtained results that are

considerably better when compared to *clari-onsets-50*, it is worth emphasizing that it is a small dataset, having a total duration that corresponds to only 3.4% percent of the latter, and there is certainly a bias related to the fact that its recordings were all made in a laboratory, under controlled conditions, in a room with very low reverberation, by only two clarinetists. The low reverberation is certainly a factor that makes it a lot easier for any method to detect the onsets since it minimizes the superposition of energy of consecutive notes.

6.4 Future Prospects

Although we did some preliminary experiments training the onset detection model using additional semi-synthetic data, so far, the results have not been satisfactory. That data was created using a MIDI keyboard controller to record monophonic melodies on a commercial clarinet sampler software. The *note on* events of the MIDI protocol were used to generate the onset annotations, and those were used in our training experiments. We attempted to use this semi-synthetic data in the CNN model using two different strategies: (1) employing them on a pre-training phase and (2) mixing them in the training data. So far, none of these approaches improved the results of the model. As a matter of fact, they produced slightly worse results than the models trained without them. Yet, this is something that still needs further investigation. We also did some experiments applying data augmentation techniques to the train data (such as pitch-shifting, time-stretching, gain adjustment, and addition of noise). However, in our preliminary experiments using these techniques, the results of the model got slightly worse. This is, again, something that needs to be explored further.

7

Iracema

The main product of this thesis is a Python package for audio content analysis aimed at the empirical research on music performance called *Iracema* (version 0.2.1). It provides functionalities for extracting patterns of manipulation of duration, energy, and spectral content from monophonic audio (MAGALHAES; BARROS; LOUREIRO, 2020). Its development was motivated by research projects conducted at CEGeME¹ (LOUREIRO et al., 2019), and was strongly inspired by a previous Matlab tool developed by the group, called *Expan* (CAMPOLINA; MOTA; LOUREIRO, 2009), which has not been released for public use. Iracema is licensed under the GNU General Public License v3.0, and its source code can be freely obtained at <<https://github.com/cegeme/iracema>>. The documentation of the API is available at <<https://cegeme.github.io/iracema>>. This chapter will provide an overview of the proposed architecture of the system, some technical characteristics and basic functionalities. To check some code examples for the library, refer to appendix B.

Iracema uses *NumPy* arrays for storing and manipulating data, providing a new level of abstraction on top of such objects (WALT; COLBERT; VAROQUAUX, 2011). It also wraps some functionalities from other libraries, such as *SciPy* (JONES et al., 2001), *librosa* (MCFEE et al., 2015), *CREPE* (KIM et al., 2018), *resampy* (MCFEE, 2016; SMITH, 2020), and *audioread* (SAMPSON, 2020), to provide methods with a more natural interface for audio content extraction.

¹ <http://musica.ufmg.br/cegeme/>

7.1 Architecture

Software architecture refers to the set of structures needed to reason about a system. These structures are comprised of software elements, relations among them, and properties of both elements and relations (BASS; CLEMENTS; KAZMAN, 2013). This section will discuss some important aspects of Iracema’s architecture and offer an overview of the elements that compose the core functionalities of the library.

Audio content analysis systems rely on manipulation of dynamic data, i.e., data that represents an attribute’s changes over time. Thus, *time series* is a fundamental element in Iracema’s architecture. The starting point for any task performed by the system is the *audio time series*, from which other kinds of time-related data are extracted. The transformation of time series into other time series, to obtain more meaningful representations of the underlying audio, is a common behavior of audio content analysis systems, to perform *feature extraction*. The implementation of such extractors usually depends on some recurrent types of operations, like applying sliding windows to a series of data, which is an example of an *aggregation* operation.

Frequently, it is necessary to deal with a specific excerpt of a time series, such as a musical phrase or a single musical note. There is another important element in the architecture, called *segment* which is defined as the interval between two separate *points*, and can be used to easily delimit such excerpts in a time series, independently of its sampling rate. A user may sometimes specify the limits of the segments they want to study, if they already know where these points are located. However, most of the time, users expect the system to identify those limits automatically, a common kind of task in audio content extraction, known as *segmentation*.

Some aforementioned elements, like audio, time series, points, and segments were implemented as classes, since they have intrinsic attributes (e.g., the samples of the time series, and the start/end of the segments) and behavior (e.g., generating time vectors in time series or calculating indexes in segments). Figure 11 shows those classes in a diagram. The class `Audio` inherits the functionalities from `TimeSeries`, and add some specific behaviors (such as loading wave files). The classes `Point`, `Segment`, `PointList` and `SegmentList` provide a handy way to extract corresponding excerpts from time series of different sampling rates, since it performs all the necessary index conversion operations to extract data that coincide with the same time interval.

Feature extraction methods or classes take time series objects as input and output another time series object. For example, the initializer method of the class `STFT` takes as input an `audio` object, a `window_size`, and a `hop_size`, and generates a time series in which each sample contains the bins of the FFT within one analysis window. Another

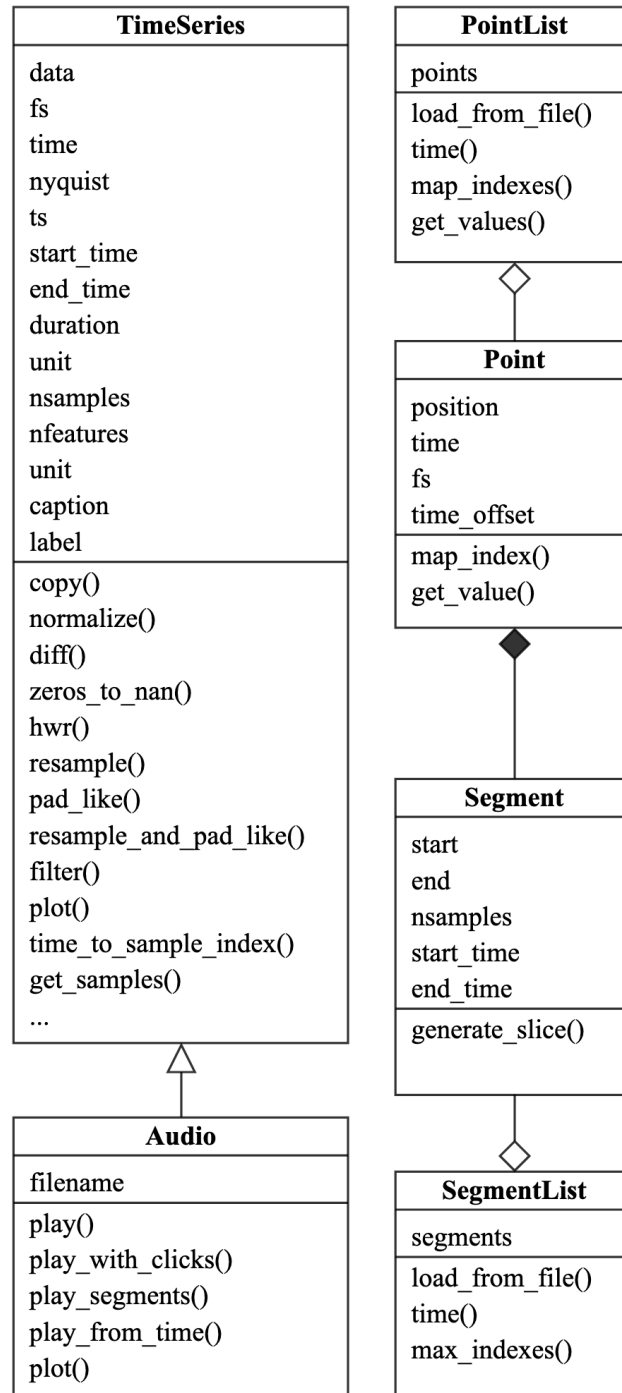


Figure 11 – Diagram showing the core classes of Iracema.

example, the method `spectral_flux` (section 7.2.3.10) takes an STFT time series as input and generate another time series containing the calculated spectral flux.

Segmentation methods take `time_series` objects as input and output a list of segments or points. Then, these segments can be used to easily extract excerpts from time series objects, using square brackets notation (the same operator used in Python to perform indexing/slicing operations).

7.2 Modules and Functionalities

These are the modules that compose Iracema, and their respective functionalities:

- `core.timeseries`: contains the definition of the class `TimeSeries`.
- `core.audio`: contains the definition of the class `Audio`.
- `core.segment`: contains the definition of the classes `Segment` and `SegmentList`.
- `core.point`: contains the definition of the classes `Point` and `PointList`.
- `spectral`: contains the definition of the classes that implement frequency-domain analysis methods, like the STFT, spectrogram, mel spectrogram and constant-Q transform.
- `pitch`: a few different models for pitch detection (detailed in section 7.2.1).
- `harmonics`: a model for extracting harmonic components from audio.
- `segmentation`: methods for automatic audio segmentation.
- `features`: contains the implementation of several feature extractors (detailed in sections 7.2.3 and 7.2.6).
- `plot`: contains several different methods for plotting time series data.
- `aggregation`: contains some common aggregation methods that can be useful for implementing feature extractors.
- `io`: subpackage containing IO methods, for loading/writing files, playing audio, etc.
- `util`: subpackage containing some useful modules for unit conversion, DSP, windowing operations, etc.

The core modules implement the core classes of the library. To make these classes easily available for users, they are all already imported into the namespace `iracema`. For instance, the class `Audio` is available in the module `iracema.core.audio`, but it can be accessed using the shorter name `iracema.Audio`.

7.2.1 Pitch Detection

According to the ANSI standard 1994, “Pitch is that attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from low to high. Pitch depends mainly on the frequency content of the sound stimulus, but it also depends on the sound pressure and the waveform of the stimulus”. Two pitch detection methods have been implemented in Iracema, and there is an extra method that wraps a model from the external library CREPE.

7.2.1.1 Harmonic Product Spectrum

Measures the maximum coincidence for partials, based on successive down sampling operations on the frequency spectrum of the signal (CUADRA, 2001). If the signal contains harmonic components, then it should contain energy in the frequency positions corresponding to the integer multiples of the fundamental frequency. So by down-sampling the spectrum by increasing integer factors ($1, 2, 3, \dots, R$) it is possible to align the energy of its harmonic components with the fundamental frequency of the signal.

Then the original magnitudes of the spectrum are multiplied by its downsampled versions. This operation will make a strong peak appear in a position that corresponds to the fundamental frequency. The HPS calculates the maximum coincidence for harmonics, according to the equation

$$Y(\omega) = \prod_{r=1}^R |X(\omega r)| \quad (7.1)$$

where $X(\omega r)$ represents one spectral frame and R is the number of harmonics to be considered in the calculation. After this calculation a simple peak detection algorithm is used to obtain the fundamental frequency of the frame.

Our implementation modifies this approach by adding an offset of 1 to the magnitude spectrum of the signal before applying the product shown in the equation above. This makes the algorithm more reliable in situations where some harmonics have very little or no energy at all.

7.2.1.2 Expan Pitch Detection Algorithm

Based on the algorithm implemented in the tool Expan (CAMPOLINA; MOTA; LOUREIRO, 2009), this method consists, for each spectral frame, in choosing the n highest local peaks of the magnitude spectrum (above a certain minimum relative threshold) as potential candidates, and estimating the corresponding harmonics for each candidate, using the criteria specified in section 7.2.2. Then, the harmonic energy is calculated for each candidate (section 7.2.3.14). The candidate with the highest harmonic energy is chosen as the pitch frequency for each spectral frame.

7.2.1.3 CREPE

This pitch detection method is based on a deep convolutional neural network operating directly on the time-domain waveform (KIM et al., 2018). It was developed by the Music and Audio Research Laboratory, at the New York University, and is available in *Iracema* as a wrapper over their published Python package. It uses six convolutional layers connected to a densely connected output layer. The output produced by this layer is a 360-dimensional vector, from which the pitch estimate is calculated deterministically. The frequencies that correspond to the output follow a logarithmic scale, and cover six octaves in intervals of 20-cents. The model was trained on two large datasets containing synthesized audio, for which it is possible to have very precise target annotations. The method produces excellent pitch estimations (according to the authors, state-of-the-art as of 2018), but it is highly demanding in terms of computational power when set to its maximum capacity. Lowering the capacity of the model (it is one of its input arguments) it is possible to find a good compromise between accuracy and processing time.

7.2.2 Harmonics

The harmonic model uses as input the pitch curve extracted for a given audio signal and its STFT. It calculates the expected position of the theoretical harmonics of the signal using integer multiples of the fundamental frequency. Then, a local search for peaks is performed around this theoretical position, within a tolerance interval. This step is important to account for some inharmonicity, which is quite frequent in higher harmonic components.

7.2.3 Classic Features

This section contains the definition several classic feature extractors that are typically used in the *Music Information Retrieval* (MIR) field, and are included in the library Iracema. We use the following conventions in the definitions:

- an audio signal is denoted by $x(n)$, where $n \in \mathbb{N}$ is the sample index;
- $X(k)$ is the result of the DFT applied to the signal $x(n)$, where $k \in \mathbb{N}$ is the index of the spectral bin;
- the last definition is useful to simplify definitions that apply to each spectral frame separately; alternatively, we may denote the DFT by $X(k, t)$, where t represents the index spectral frame over the temporal dimension;
- the amplitudes of the harmonic components of a signal are denoted by $A(h)$ where $h \in \mathbb{N}$ is the number of the harmonic component ($A(1)$ is the fundamental frequency);
- μ_s and σ_s are the mean and standard deviation of s .

7.2.3.1 Peak Envelope

The peak envelope consists in the peak absolute values of the signal amplitude, within each step of a sliding window.

$$\text{PE} = \max(|x(n)|), 1 \leq n \leq L \quad (7.2)$$

Where $x(n)$ is the n -th sample of a window of length L .

7.2.3.2 RMS

The RMS envelope consists in the root mean square of the signal amplitude, calculated within the aggregation window. For a sliding window of length L :

$$\text{RMS} = \sqrt{\frac{1}{L} \sum_{n=1}^L x(n)^2} \quad (7.3)$$

7.2.3.3 Zero-Crossing Rate

The zero-crossing rate of a time series consists in the number of times the signal crosses the zero axis per second. It gives some insight on the noisiness character of a sound. In noisy or unvoiced signals, the zero-crossing rate tends to reach higher values than in periodic or voiced signals.

$$ZC = \frac{1}{2L} \sum_{n=1}^L |\text{sgn}[x(n)] - \text{sgn}[x(n-1)]| \quad (7.4)$$

Where

$$\text{sgn}[x(n)] = \begin{cases} 1, x(n) \geq 0 \\ -1, x(n) < 0 \end{cases} \quad (7.5)$$

And $x(n)$ is the n -th sample of a window of length L .

7.2.3.4 Spectral Flatness

The spectral flatness gives an estimation of the noisiness / sinusoidality of an audio signal (for the whole spectrum or for a frequency range). It can be used to determine voiced / unvoiced parts of a signal (PARK, 2004). It is defined as the ratio between the geometric mean and the arithmetic mean of the energy spectrum:

$$\text{SFM} = 10 \log_{10} \left(\frac{\left(\prod_{k=1}^N |X(k)| \right)^{\frac{1}{N}}}{\frac{1}{N} \sum_{k=1}^N |X(k)|} \right) \quad (7.6)$$

7.2.3.5 High Frequency Content

The high frequency content (HFC) function produces sharp peaks during attacks or transients (BELLO et al., 2005) and might be a good choice for detecting onsets in percussive sounds.

$$\text{HFC} = \sum_{k=1}^N |X(k)|^2 \cdot k \quad (7.7)$$

Alternatively, it can be calculated using the spectral amplitude instead of the energy:

$$\text{HFC} = \sum_{k=1}^N |X(k)| \cdot k \quad (7.8)$$

7.2.3.6 Spectral Centroid

The spectral centroid is a well known timbral feature that is used to describe the brightness of a sound. It represents the center of gravity of the frequency components of a signal (PARK, 2010).

$$\text{SC} = \frac{\sum_{k=1}^N |X(k)| \cdot f_k}{\sum_{k=1}^N |X(k)|} \quad (7.9)$$

7.2.3.7 Spectral Spread

The spectral spread represents the spread of the spectrum around the spectral centroid (PEETERS et al., 2011), (LERCH, 2012).

$$\text{SSp} = \sqrt{\frac{\sum_{k=1}^N |X(k)| \cdot (f_k - \text{SC})^2}{\sum_{k=1}^N |X(k)|}} \quad (7.10)$$

Where SC is the spectral centroid for the frame.

7.2.3.8 Spectral Skewness

The spectral skewness is a measure of the asymmetry of the distribution of the spectrum around its mean value, and is calculated from its third order moment. It will output negative values when the spectrum has more energy below the mean value, and positive values when it has more energy above the mean. Symmetric distributions will output the value zero (LERCH, 2012).

$$\text{SSk} = \frac{2 \cdot \sum_{k=1}^N (|X(k)| - \mu_{|X(k)|})^3}{N \cdot \sigma_{|X(k)|}^3} \quad (7.11)$$

7.2.3.9 Spectral Kurtosis

The spectral kurtosis is a measure of the flatness of the distribution of the spectrum around its mean value, compared to a Gaussian distribution. For Gaussian distributions, its value is equal to zero. For flatter distributions it outputs negative values, while for sharply-peaked distributions it outputs positive values.

$$\text{SKu} = \frac{2 \cdot \sum_{k=1}^N (|X(k)| - \mu_{|X(k)|})^4}{N \cdot \sigma_{|X(k)|}^4} \quad (7.12)$$

7.2.3.10 Spectral Flux

The spectral flux measures the amount of change between successive spectral frames. There are different methods to calculate the spectral flux across the literature. For now we have implemented the one proposed by (DIXON, 2006).

$$\text{SF} = \sum_{k=1}^N H(|X(t, k)| - |X(t-1, k)|) \quad (7.13)$$

Where $H(x) = \frac{x+|x|}{2}$ is the half-wave rectifier function, and t is the temporal index of the frame.

7.2.3.11 Harmonic Centroid

The harmonic centroid represents the center of gravity of the amplitudes of the harmonic series. It is basically the same concept as the Spectral Centroid, but applied to the harmonic partials of a signal.

$$\text{HC} = \frac{\sum_{k=1}^H A(k) \cdot f_k}{\sum_{k=1}^H A(k)} \quad (7.14)$$

Where $A(h)$ represents the amplitude of the h -th harmonic partial.

7.2.3.12 Spectral Entropy

The spectral entropy is based on the concept of information entropy from Shannon's information theory (MATHWORKS, 2021). It measures the unpredictability of

the given state of a spectral distribution.

$$\text{SE}_{\text{py}} = - \sum_k^N P(k) \cdot \log_2 P(k) \quad (7.15)$$

Where

$$P(i) = \frac{|X(i)|^2}{\sum_j^N |X(j)|^2} \quad (7.16)$$

7.2.3.13 Spectral Energy

Spectral energy is the sum of the energies of all spectral components from the FFT of the signal.

$$\text{SE} = \sum_{k=1}^N |X(k)|^2 \quad (7.17)$$

7.2.3.14 Harmonic Energy

The harmonic energy is the sum of the energies of the harmonic partials of a signal.

$$\text{HE} = \sum_{k=1}^H A(k)^2 \quad (7.18)$$

7.2.3.15 Noisiness

The noisiness represent how noisy a signal is (values closer to 1), as oposed to harmonic (values close to 0). It is the ratio of the noise energy to the total energy of a signal (PEETERS et al., 2011).

$$\text{Ns} = \frac{\text{SE} - \text{HE}}{\text{SE}} \quad (7.19)$$

7.2.3.16 Odd-to-Even Ratio

The OER is the odd-to-even ratio among the harmonics of an audio signal (PEETERS et al., 2011). This value will be higher for sounds with predominantly odd

harmonics, such as the clarinet.

$$\text{OER} = \frac{\sum_{h=1}^{H/2} A(2h-1)^2}{\sum_{h=1}^{H/2} A(2h)^2} \quad (7.20)$$

7.2.4 Note Onset Detection

7.2.4.1 CNN Model

This is the main note onset detection method provided by the library, which consists in a clarinet-specific model trained using the dataset *Clari-onsets-50*, discussed in detail in chapter 6.

7.2.4.2 Adaptative RMS

This method was proposed by (De Poli; MION, 2006), and consists on the calculation of two RMS curves: one with a short window length, and another with a large window length. These curves will intersect each other along the audio signal, and the onsets will tend to occur in the valleys of the interval between two intersections, when the values for the RMS calculated with the shorter window are smaller than the values calculated for the other curve. So the difference between these curves can be calculated and used as an onset detection function (ODF). The peaks of this curve are extracted to obtain the points corresponding to the note onsets.

7.2.4.3 Pitch Change

This method is based on the detection of changes in the pitch values for adjacent frames. It calculates the ratio between adjacent frames to generate an ODF. The peaks in this curve will correspond to instants with fast changes in pitch. It is necessary to establish a threshold to consider a peak in the ODF as an onset. The cons of this method are: (1) it is highly dependent on a good pitch detection method, possibly with a good post processing method for smoothing the pitch curve; and (2) it usually fails to detect consecutive notes of the same pitch.

7.2.4.4 Derivative of the RMS

Since note onsets usually result in energy increases in the audio signal, the derivative of the RMS can be used to estimate the note onsets. It can be used as the ODF, since the peaks in this will correspond to instants of rapid variation in the energy of

the signal. This method tends to produce false negatives in legato phrases, since there might be little or no energy variation in the audio signal.

7.2.5 Note Envelope Segmentation

The typical note envelope model used in sound synthesis usually divides the note into four segments: attack, decay, sustain, and release (ADSR). Although this model is useful for synthesis purposes, when analyzing real recordings of acoustic instruments some of these segments might be highly ambiguous and difficult to define or detect, or might not even exist. We adopt a simplified definition for analyzing the note envelope, discarding the decay segment, to obtain a ASR envelope. Figure 12 shows an illustration representing the ASR envelope.

To segment the notes in an audio signal in terms of these three segments (ASR), we must first obtain the note onsets using one of the available methods (section 7.2.4). Then, for each IOI, the following criteria is used to obtain the other three points of the note envelope:

- the spectral flux (using the positive first-order difference implementation, defined in section 7.2.3.10) is calculated for the entire IOI segment, and its peak value is chosen as the beginning of the release;
- the peak value of the RMS between the onset and the beginning of the release is chosen as the end of the attack;
- the first derivative pitch curve is estimated and used to determine the offset point, within the segment starting at the beginning of the release and ending at the onset of the next note: if the derivative of the pitch curve exceeds 50 cents (half semitone), this point is chosen as the offset, else the onset of the next note is chosen as the offset point.

7.2.6 Expressiveness Features

7.2.6.1 Local Tempo

The local tempo consists in a granular estimation of the musical tempo, calculated for each IOI segment in an excerpt. To calculate it, we must first obtain the duration of the IOIs.

$$\text{IOI}(n) = \text{onset}(n + 1) - \text{onset}(n) \quad (7.21)$$

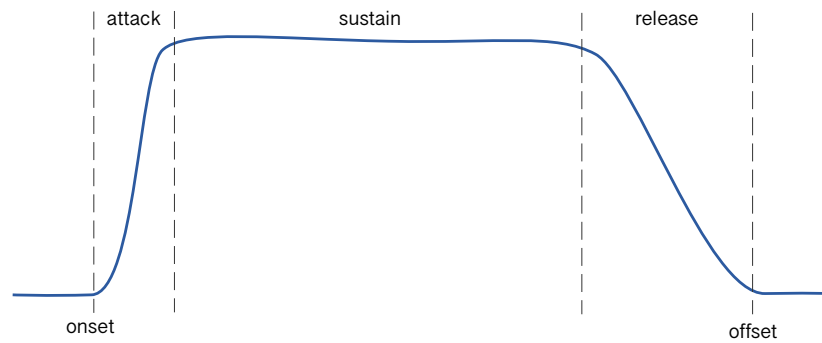


Figure 12 – Illustration of a simplified version of the note envelope, composed by three segments: attack, sustain and release.

note	duration
♩	4
♪	2
♫	1
♬	0.5

Table 9 – Examples of numeric values corresponding to nominal durations in the score.

In the equation above, $\text{onset}(n)$ is the onset time of the n -th note. To normalize the durations of the IOIs, we divide them by their nominal durations, $\text{ND}(n)$.

$$\text{IOI}_{\text{normalized}}(n) = \frac{\text{IOI}(n)}{\text{ND}(n)} \quad (7.22)$$

The nominal duration of the n -th IOI is obtained directly from the the musical score. It consists in the notated duration corresponding to the start of the IOI plus the duration of the rests (if there are any) between that note and the subsequent note. In table 9 we show some examples of values corresponding to the nominal durations specified by the score.

Finally, for any time signature with a quarter note as the beat unit, we can use the following formula to calculate the local tempo $\text{LT}(n)$ (in BPM):

$$\text{LT}(n) = \frac{60}{\text{IOI}_{\text{normalized}}(n)} \quad (7.23)$$

7.2.6.2 Legato Index

An articulated note transition is characterized by a plunge in energy in the beginning of the release of the first note, followed by an ascent until the end of the attack of the second note. This definition of the legato index assumes that a perfect legato

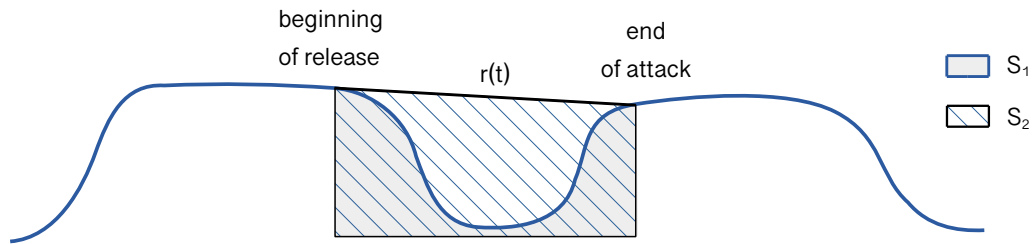


Figure 13 – Illustration showing the areas S_1 and S_2 used to calculate the legato index.

consists in a linear change in energy during the note transition, from the beginning of release to the end of attack. By estimating how much the energy declines below this legato baseline, it is possible to estimate the quality in terms of legato/articulation. For a given note transition, the legato index is given by

$$LI = \min\left(\frac{S_1}{S_2}, 1\right) \quad (7.24)$$

$$S_1 = \sum_{t=r_i}^{a_{i+1}} RMS(t) \quad (7.25)$$

$$S_2 = \sum_{t=r_i}^{a_{i+1}} r(t) \quad (7.26)$$

Where r_i is the index corresponding to the start of the release of the note i and a_{i+1} the start of the attack of the subsequent note. $r(t)$ is defined as a straight line between r_i and a_{i+1} . This index was adapted from the definition proposed by (MAESTRE; GÓMEZ, 2005). The areas S_1 and S_2 are shown in figure 13. S_1 correspond to the area under the energy curve, and S_2 the area under the legato base line connecting the energy values at the beginning and the end of the transition.

7.3 Case Study

To demonstrate the practical use of some content extraction methods studied in this thesis, we conducted a brief case study of clarinet performances using the library Iracema.

label	clarinetist	level of expertise
A	1	professional
B		
C	2	student
D	3	professional
E	4	student
F	5	professional
G	6	student
H	7	professional
I	8	student
J	9	professional
K	10	student

Table 10 – List of performances analyzed, indicating the label associated to the performance, a numeric id corresponding to the performer and the indication of the level of expertise of the clarinetist.

Moderato

in A

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47

Figure 14 – Score for the excerpt of *Peter and the Wolf opus 67* by Sergei Prokofiev.

7.3.1 Methods

We chose to analyze an excerpt of *Peter and the Wolf opus 67* (the cat theme), by Sergei Prokofiev (shown in figure 14). These excerpts were obtained from our research group’s database, and consist in a total of 11 recordings by 10 different clarinetists, 5 professional players and 5 students. The recordings are available at <https://taironemagalhaes.github.io/phd-thesis-audio/case-study/>. Two of these recordings were made by a single professional clarinetist. Table 10 shows the label used to refer to each performance (A to K), the label representing the clarinetist (1 to 10), and their level of expertise.

First of all, we extracted the note onsets using the specialized clarinet note onset detection model obtained in chapter 6. All the note onsets in the recordings

were correctly identified by the model, so there were no false negatives. However, the model produced 23 false positives, which were removed manually for this experiment (obtaining a total of 517 note onsets). No performance had any extra or missing notes, so the relationship between the performed note onsets in the recordings and the notes in the score were all biunivocal. This allowed us to directly compare these performances using features that are estimated on a per-note basis (or per-transition basis). We analysed those excerpts using the features local tempo (section 7.2.6.1) and legato index (section 7.2.6.2).

7.3.2 Results

This section shows the results obtained using the features local tempo and legato index for all the recorded excerpts of *Peter and the Wolf opus 67*.

7.3.2.1 Local Tempo

The local tempi were calculated for all the recordings using the estimated onset times. We plotted those values in figures 15 and 16. The excerpt was split in two parts in these graphs to make the results easier to analyze. The first part is shown in figure 15 (IOIs 1 to 23) and the second in figure 16 (IOIs 24 to 46). Also, the performances were split into three different axes to reduce cluttering.

There are accent marks in the score on notes 39, 41, and 43, indicating that the performer must emphasize them. There are also legato marks connecting those notes to their following notes. As can be seen in figure 16, there is a clear zig-zag pattern in the local tempi for those notes, intercalating longer IOIs (odd numberings) with shorter IOIs (even numberings). This is probably due to the fact that accented notes tend not just to be played stronger, but also to have their durations prolonged. This is a typical pattern that has been reported in other studies, including the seminal one by Binet and Courtier (1895). Yet, in many of these performances the zig-zag pattern starts not at note 39 but note 35 (this is easily noticeable in performances A, B, C and D), although these notes have no explicit accent marks. This might be explained by the melodic pattern built by an upward third leap followed by a downward second, notes 35–36 and 37–38, which are repeated at notes 39–40, 41–42 and 43–44, going down stepwise of a second, with an accent at the first note of each leap. It seems that the players anticipate these accents at notes 35–38 even though no accent is marked.

The performances executed by the same clarinetist (A and B) were very consistent, judging by the similarity between the two curves. Figure 17 shows those performances in a single graph, to make this similarity easier to observe.



Figure 15 – Local tempo part 1.

We computed a correlation matrix for those performances using the local tempo values. The results are shown in figure 18. This result confirms the high consistency between the two performances by the same clarinetist (A and B), which exhibited the highest correlation coefficient among all the pairs of performances (0.83). By comparing performances A and B to performances by other clarinetists, we observed higher correlation coefficients for F (0.67 and 0.56), H (0.65 and 0.68) and K (0.7 and 0.61). The performances F and H were executed by professional clarinetists, and K by a student. The correlation coefficients between the performance D and most other performances were small, suggesting that this performance might be the most distinctive one in terms of timing manipulations.

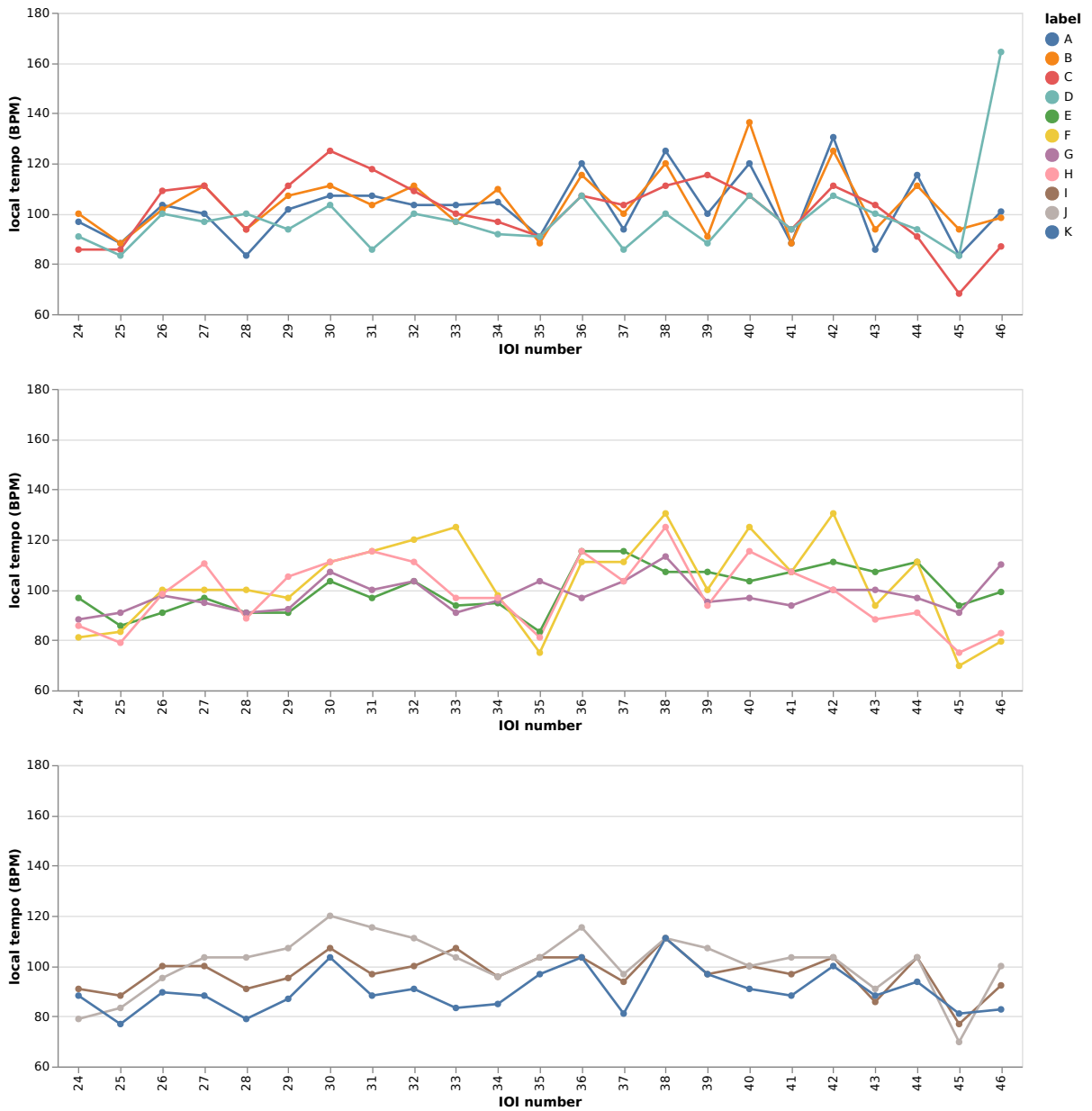


Figure 16 – Local tempo part 2.

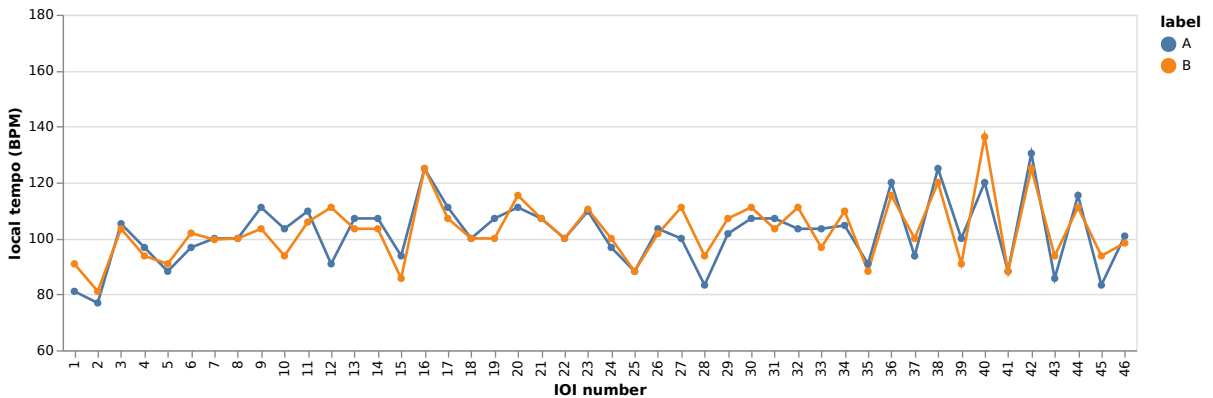


Figure 17 – Local tempo for two performances by the same clarinetist.

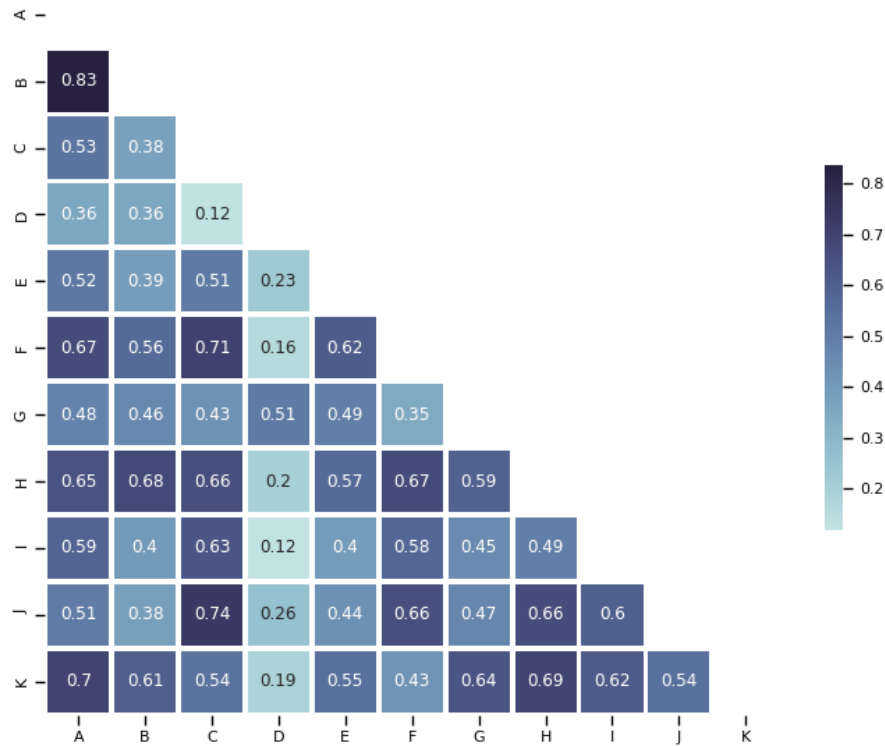


Figure 18 – Correlation matrix for the local tempi for all the performances.

7.3.2.2 Legato Index

We calculated the legato index for all the note transitions in the recordings. The results obtained are shown in figure 19 and figure 20. The score indicates that the following transitions must be played in staccato: 1-10, 17-18, 21-22, and 24-33. In contrast, the transitions 15, 19, 35-37, 39, 41, 43, and 46 must be played in legato. The results indicate that the legato index is able to characterize these transitions reasonably well, since its values tend to be higher for the notes with legato marks and lower for notes with staccato marks.

The correlation matrix computed using the legato indexes for all the performances is shown in figure 21. The highest correlation occurred between the performances A and H (0.84), both professional clarinetists. The second largest value occurred between the performances A and B (0.79), both from the same clarinetist. The correlation values tended to be higher between performances by professional clarinetists, with the exception of the performances J (professional) and K (student).

7.4 Future Prospects

At the time of writing Iracema contains an effective note onset detector for clarinet recordings. Although preliminary tests seem to indicate that this model pro-

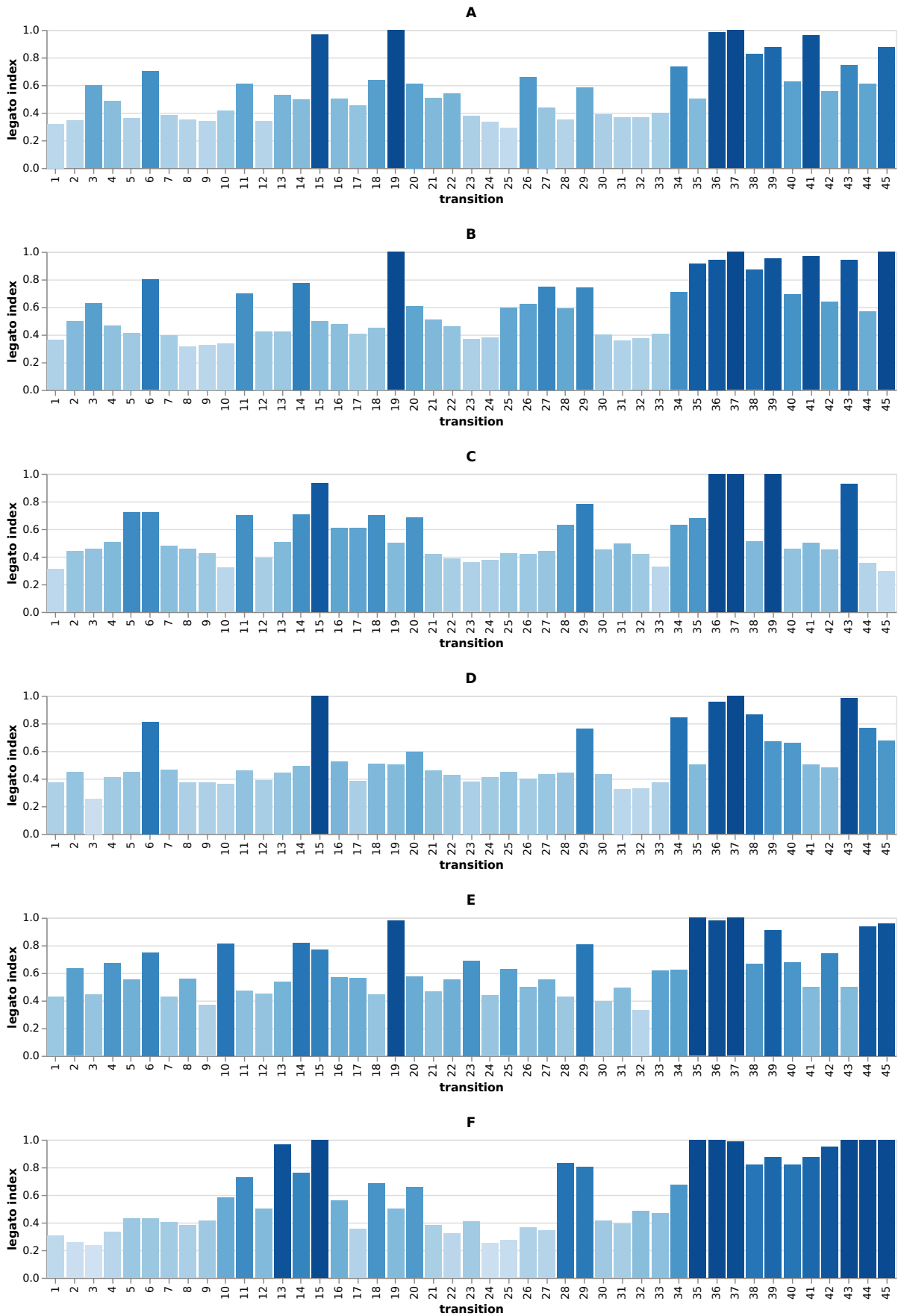


Figure 19 – Legato indexes for performances A to F.

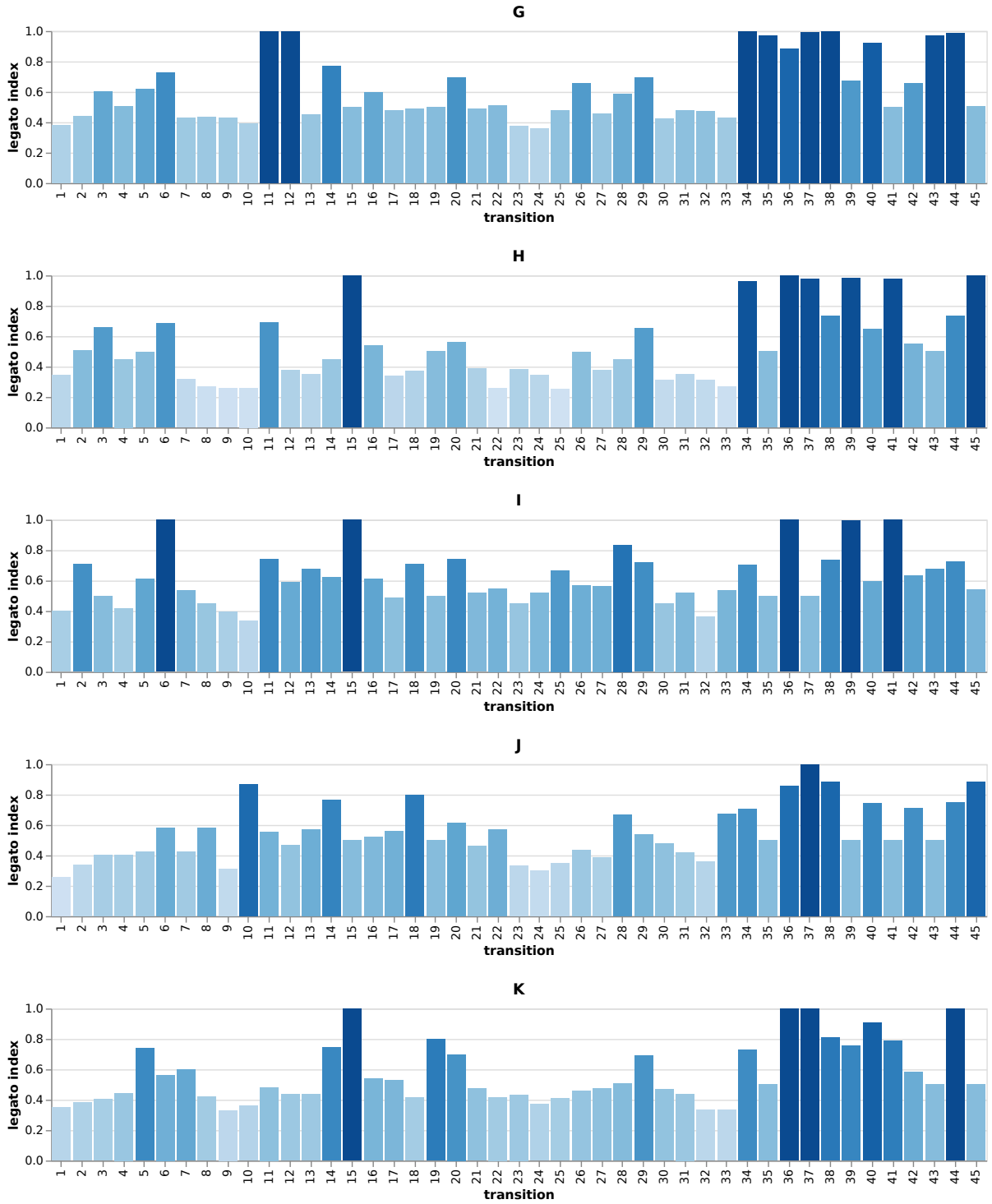


Figure 20 – Legato indexes for performances G to K.

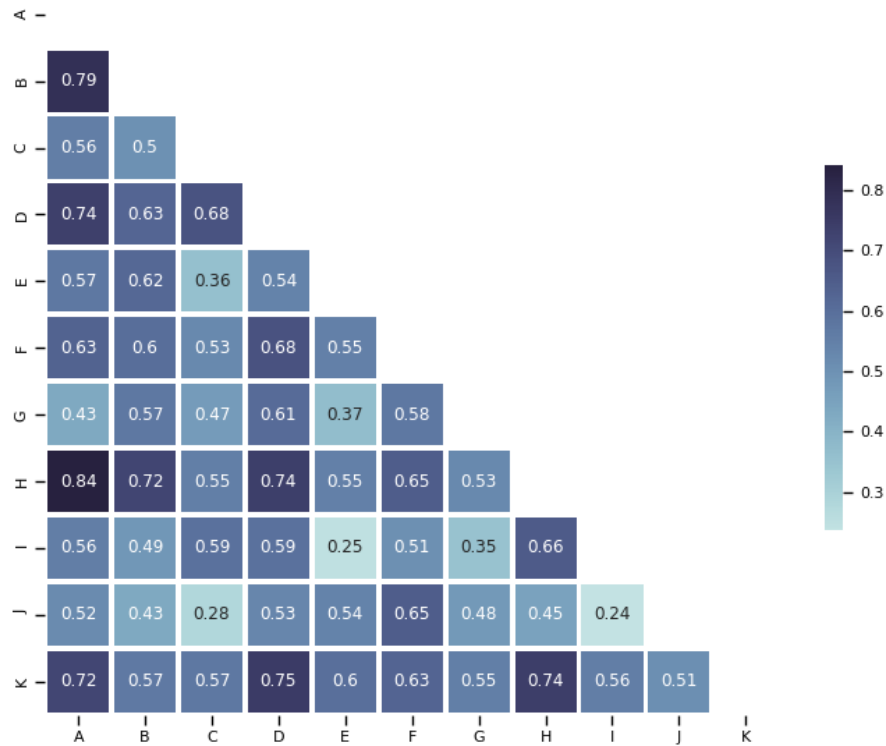


Figure 21 – Correlation matrix for the legato indexes for all the performances.

duces reasonable results for other monophonic instruments, this should be further investigated on larger datasets containing multiple instruments. To reach the best result possible for other instruments, it would be probably ideal to have available in the library other specific models trained for different instruments. Also, although Iracema includes the implementation of methods to calculate the local tempo and legato index, the library should be further extended to include additional expressive features (e.g. vibrato and higher-level timbral features). The current functionalities can also be improved, specially regarding the usability of the objects defined by the core classes of the system (*TimeSeries*, *Point*, *Segment*, etc.). Collecting opinions and suggestions from the final users of the library could certainly be highly helpful to define the most important points of improvement for the library, a step that was left out of the scope of the work performed in this thesis.

8

Conclusion

This thesis discussed techniques for extracting information from monophonic musical audio, aiming to support the empirical research on music performance. Our models and experiments adopted the clarinet as the baseline instrument. Among several topics explored in our research, we assessed the measurement error for a manual note onset annotation experiment using a small dataset containing clarinet performances recorded in our laboratory, in a room with low reverberation. The results showed that the annotation error was within ± 8 ms for 96.5% of the onsets, and ± 12 ms for 99.0% of the onsets. We also trained a specific note onset detection model for clarinet recordings based on a convolutional neural network, using a dataset that we created and annotated manually. This model performed significantly better to detect onsets in clarinet recordings than other models trained on mixed recordings did. The results obtained with this model on the dataset *clari-onsets-3* might be evidence that it tends to perform better in recordings with little reverberation. Our motivation for developing this method was to obtain an assertive model for the empirical study of clarinet performances, which in many cases consist of recordings made in smaller rooms (typically a laboratory or music studio). We obtained an F-score of approximately 99% for three recordings made under such low reverberation conditions (for a 50 ms tolerance window). Even in a more diverse clarinet dataset containing a large number of recordings made in concert rooms or halls with more reverberation, the model reached an F-score of approximately 95%. We believe that this model will enable faster data collection for the analysis of music performances. The most noticeable shortcomings of the model are: false negatives tend to occur in notes with a soft attack and in consecutive notes with the same pitch; false positives apparently tend to occur as duplicate onset detections, in situations where there is, in fact, just one onset.

The clarinet-specific note onset detection model, along with many other useful methods for analyzing recordings of monophonic performances, were implemented in the library *Iracema*. Many of these consisted of classic features, widely employed in the MIR field, such as RMS, spectral flux, spectral entropy, noisiness, and others. Nonetheless, there are also two higher-level features available in the library, which we decided to categorize as "expressiveness features": local tempo and legato index. In future versions of the library, we plan to include additional higher-level features, e.g., portamento, dynamics, and vibrato. We also wish to extend it futurally to other monophonic instruments, such as flute, trumpet, trombone, oboe, etc. In fact, we did some preliminary tests using the onset detection model on a few trumpet and trombone recordings, and so far, the results generated by the model for those instruments seem to be pretty good. However, we still need to evaluate this on a larger dataset.

Our case study showed the local tempi and legato indexes for 11 different performances of the same excerpt. Two performances were made by the same clarinetist, while among the other nine, each was played by a different clarinetist. The results showed a high correlation in both local tempo and legato index for two performances by the same professional clarinetist. We had already anticipated that this would probably occur because, over time, musicians develop a consistent individual style. Furthermore, professional musicians, in particular, will typically have not only a clear notion of the results they want to achieve in the performance, but also the necessary technical skill to produce highly consistent interpretations, which reflect their individual notion of the performance. Sloboda (2000) mentioned that "because effective expressive performance often requires very fine and subtle variations in performance parameters [...], expressive intentions frequently cannot be effectively communicated without a high level of technical mastery on the part of the performer."

Another interesting observation from the case study results was that performances executed by professional clarinetists tended to be more correlated to each other than to performances executed by students. Also, the correlation values observed between students' performances tended to be lower (although there were some obvious exceptions). A probable explanation for this result is that professional musicians have a more precise notion of the "archetypical" interpretation based on the piece's structural characteristics. According to (REPP, 1992), "there are two basic aspects of music performance: a normative aspect (i.e., commonality) that represents what is expected from a competent performer and is largely shared by different artists, and an individual aspect (i.e., diversity) that differentiates performers.". Thus, the high correlation values among professional clarinetists might be related to this normative aspect of the performance.

Bibliography

- BARTHET, M. et al. Acoustical Correlates of Timbre and Expressiveness in Clarinet Performance. *Music Perception*, v. 28, n. 2, p. 135–153, 2010. Cited in page 19.
- BARTHET, M. et al. Analysis-By-Synthesis of Timbre, Timing, Dynamics. *Music Perception*, v. 28, n. 3, p. 265–278, 2011. Cited in page 19.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software Architecture in Practice*. Third edit. [S.l.]: Addison-Wesley Publishing Company, 2013. Cited in page 50.
- BELLO, J. P. et al. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, v. 13, n. 5, p. 1035–1046, 2005. ISSN 10636676. Cited 2 times in pages 38 and 56.
- BENGTSSON, I. Empirische Rhythmusforschung in Uppsala. *Hamburger Jahrbuch für Musikwissenschaft*, v. 1, p. 195–220, 1974. Cited in page 17.
- BENGTSSON, I.; GABRIELSSON, A. Rhythm Research in Uppsala. *Music, Room, Acoustics*, v. 17, p. 19–56, 1977. Cited in page 17.
- BENGTSSON, I.; GABRIELSSON, A. Methods for analyzing performance of musical rhythm. *Scandinavian Journal of Psychology*, v. 21, n. 1, p. 257–268, 1980. ISSN 14679450. Cited in page 19.
- BENGTSSON, I.; GABRIELSSON, A.; THORSRN, S. M. Empirisk rytmforskning. *Swedish Journal of Musicology*, v. 51, p. 49–118, 1969. Cited 2 times in pages 17 and 30.
- BERNAYS, M.; TRAUBE, C. Investigating pianists' individuality in the performance of five timbral nuances through patterns of articulation, touch, dynamics, and pedaling. *Frontiers in Psychology*, v. 5, n. MAR, p. 1–19, 2014. ISSN 16641078. Cited in page 4.
- BINET, A.; COURTIER, J. Recherches graphiques sur la musique. *L'année psychologique*, v. 2, n. 1, p. 201–222, 1895. ISSN 0003-5033. Cited 4 times in pages viii, 15, 16, and 65.
- BÖCK, S. et al. madmom: a new Python Audio and Music Signal Processing Library. In: *Proceedings of the 24th ACM International Conference on Multimedia*. Amsterdam, The Netherlands: [s.n.], 2016. p. 1174–1178. Cited 2 times in pages 39 and 45.
- BÖCK, S.; WIDMER, G. Maximum Filter Vibrato Suppression for Onset Detection. In: *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13), Maynooth, Ireland, September 2-5, 2013*. [S.l.: s.n.], 2013. p. 1–7. Cited in page 42.

- CAMPOLINA, T. A. M.; MOTA, D. A.; LOUREIRO, M. A. Expan: a tool for musical expressiveness analysis. In: *Proceedings of the 2nd International Conference of Students of Systematic Musicology*. Ghent, Belgium: IPEM, 2009. p. 24–27. Cited 3 times in pages 3, 49, and 54.
- CARTWRIGHT, M. et al. Seeing Sound: Investigating the Effects of Visualizations and Complexity on Crowdsourced Audio Annotations. *Proc. ACM Hum.-Comput. Interact.* 1, CSCW, Article, v. 29, n. 2, p. 29, 2017. Cited in page 31.
- CLARKE, E. Empirical methods in the study of performance. In: *Empirical musicology: Aims, methods, prospects*. [S.l.: s.n.], 2004. p. 77–102. Cited 3 times in pages 2, 15, and 17.
- COLLINS, N. Using a pitch detector for onset detection. *Proceedings of the International Symposium on Music Information Retrieval*, p. 100–106, 2005. Cited in page 38.
- COOK, N. *Beyond the Score: Music as Performance*. [S.l.]: Oxford University Press, 2013. ISBN 2013027060. Cited in page 15.
- CUADRA, P. D. L. Efficient pitch detection techniques for interactive music. In: *ICMC*. [S.l.: s.n.], 2001. p. 403–406. Cited in page 53.
- DAUDET, L.; RICHARD, G.; LEVEAU, P. Methodology and Tools for the Evaluation of Automatic Onset Detection Algorithms in Music. *Proc Int Symp on Music Information Retrieval*, p. 72–75, 2004. Cited in page 30.
- De Poli, G.; MION, L. From audio to content. In: *Unpublished book*. [S.l.: s.n.], 2006. chap. 5. Cited in page 60.
- DIELEMAN, S.; SCHRAUWEN, B. End-to-end learning for music audio. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, p. 6964–6968, 2014. ISSN 15206149. Cited in page 22.
- DIXON, S. Onset Detection Revisited. In: *9th International Conference on Digital Audio Effects*. Montreal, Canada: [s.n.], 2006. p. 133–137. Cited in page 58.
- DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. p. 1–28, 2016. ISSN 16113349. Cited in page 26.
- DUXBURY, C.; SANDLER, M.; DAVIES, M. A hybrid approach to musical note onset detection. *Computer*, p. 33–38, 2002. Cited in page 38.
- EBHARDT, K. Zwei Beiträge zur Psychologie des Rhythmus und des Tempos. *Zeitschrift für Psychologie und Physiologie des Sinnesorgane*, v. 18, p. 99–154, 1898. Cited in page 15.
- EGOZY, E. B. *Deriving Musical Control Features from a Real-Time Timbre Analysis of the Clarinet*. 67 p. Phd Thesis (PhD Thesis) — Massachusetts Institute of Technology, 1995. Cited in page 9.
- EYBEN, F. et al. Universal Onset Detection with Bidirectional Long-Short Term Memory Neural Networks. *Proceedings 11th International Society for Music Information Retrieval Conference, ISMIR 2010*, n. June 2017, p. 589–594, 2010. Cited in page 41.

GABRIELSSON, A. The Performance of Music. In: DEUTSCH, D. (Ed.). *The Psychology of Music*. 2nd. ed. [S.l.]: Academic Press, 1999. chap. 14, p. 501–602. Cited 2 times in pages 14 and 17.

GABRIELSSON, A. Music Performance Research at the Millennium. *Psychology of Music*, v. 31, n. 3, p. 221–272, jul 2003. ISSN 0305-7356. Cited 2 times in pages 2 and 14.

GABRIELSSON, A.; BENGTSSON, I.; GABRIELSSON, B. Performance of musical rhythm in 3/4 and 6/8 meter. *Scandinavian Journal of Psychology*, v. 24, n. 1, p. 193–213, 1983. ISSN 14679450. Cited 2 times in pages 19 and 30.

GOEBL, W.; DIXON, S.; De Poli, G. Sense in expressive music performance: Data acquisition, computational studies, and models. *Sound to Sense – Sense to Sound: A State of the Art in Sound and Music Computing*, p. 195–242, 2005. Cited in page 2.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.: s.n.], 2016. 785 p. Cited 3 times in pages 21, 23, and 29.

HINTON, G. E. et al. *Improving neural networks by preventing co-adaptation of feature detectors*. [S.l.], 2012. 1–18 p. Cited in page 27.

JONES, E. et al. *SciPy: Open source scientific tools for Python*. 2001. Available at: <<http://www.scipy.org/>>. Cited in page 49.

KIM, J. W. et al. CREPE: A Convolutional Representation for Pitch Estimation. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2018*. [S.l.: s.n.], 2018. ISBN 9781538646588. ISSN 0749-8063. Cited 2 times in pages 49 and 54.

KOREN, R.; GINGRAS, B. Perceiving individuality in harpsichord performance. *Frontiers in Psychology*, v. 5, n. FEB, p. 1–13, 2014. ISSN 16641078. Cited in page 4.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. p. 1–9, 2012. Cited in page 26.

LAWSON, C. *The Early Clarinet: A Practical Guide*. Cambridge: [s.n.], 2000. Cited in page 7.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015. ISSN 14764687. Cited in page 22.

LECUN, Y. et al. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998. ISSN 00189219. Cited 3 times in pages viii, 24, and 26.

LERCH, A. *An introduction to audio content analysis: Applications in signal processing and music informatics*. [S.l.: s.n.], 2012. 1–248 p. ISBN 9781118393550. Cited 2 times in pages 13 and 57.

LERCH, A. et al. Music Performance Analysis: A Survey. 2019. Cited in page 2.

LOUREIRO, M. A. et al. Extração de conteúdo musical em sinais de áudio para análise de expressividade. In: *Anais do XVII Encontro da Sociedade Brasileira de Acústica SOBRAC*. Belo Horizonte: [s.n.], 2008. p. 146–152. Cited in page 3.

- LOUREIRO, M. A. et al. A retrospective of the research on musical expression conducted at CEGeME. In: *Proceedings of the 17th Brazilian Symposium on Computer Music*. São João Del Rey: [s.n.], 2019. p. 165–172. Cited 2 times in pages 2 and 49.
- MAESTRE, E.; GÓMEZ, E. Automatic characterization of dynamics and articulation of expressive monophonic recordings. *Audio Engineering Society - 118th Convention Spring Preprints 2005*, v. 1, p. 26–33, 2005. Cited in page 63.
- MAGALHAES, T.; BARROS, F.; LOUREIRO, M. Iracema: a Python library for audio content analysis. p. 127–138, 2020. Cited in page 49.
- MATHWORKS. *Spectral Entropy Documentation*. 2021. Available at: <<https://www.mathworks.com/help/signal/ref/pentropy.html>>. Cited in page 58.
- MAZZOLA, G. *The Topos of Music*. [S.l.]: Birkhäuser, 2012. ISSN 0065-2660. ISBN 3764357312. Cited in page 13.
- MCFEE, B. *Resampy*. 2016. Available at: <<https://resampy.readthedocs.io>>. Cited in page 49.
- MCFEE, B. et al. librosa: Audio and Music Signal Analysis in Python. *Proc. of The 14th Python in Science Conference*, n. Scipy, p. 1–7, 2015. Cited in page 49.
- MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw Hill Higher Education, 1997. ISSN 03600300. ISBN 0070428077. Cited in page 21.
- MOTA, D. A. *Análise dos padrões de sincronização em duos de clarineta a partir de parâmetros acústicos e cinemáticos*. Phd Thesis (Master Dissertation) — Universidade Federal de Minas Gerais, 2012. Cited in page 19.
- MOTA, D. A. *Multi-modal coupling in musical performance*. Phd Thesis (PhD Thesis) — Universidade Federal de Minas Gerais, 2017. Cited in page 20.
- PALMER, C. Music performance. *Annual review of psychology*, v. 48, p. 115–38, 1997. ISSN 0066-4308. Cited in page 2.
- PÀMIÉS-VILÀ, M.; HOFMANN, A.; CHATZIOANNOU, V. Analysis of tonguing and blowing actions during clarinet performance. *Frontiers in Psychology*, v. 9, n. APR, p. 1–12, 2018. ISSN 16641078. Cited in page 10.
- PARK, T. H. *Towards automatic musical instrument timbre recognition*. Phd Thesis (PhD Thesis) — Princeton University, 2004. Cited in page 56.
- PARK, T. H. *Introduction to digital signal processing: Computer musically speaking*. [S.l.]: World Scientific Publishing Co. Pte. Ltd., 2010. 429 p. ISBN 9789812790279. Cited in page 57.
- PEETERS, G. et al. The timbre toolbox: extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, v. 130, n. 5, p. 2902–2916, 2011. ISSN 0001-4966. Cited 2 times in pages 57 and 59.
- POVEL, D.-J. Temporal Structure of Performed Music. *Acta Psychologica*, v. 41, p. 309–320, 1977. Cited 2 times in pages 17 and 30.

REPP, B. H. . The Art of Inaccuracy : Why Pianists' Errors Are Difficult to Hear. *Music Perception: An Interdisciplinary Journal*, v. 14, n. 2, p. 161–183, 1996. Cited in page 13.

REPP, B. H. Patterns of expressive timing in performances of a Beethoven minuet by nineteen famous pianists. *The Journal of the Acoustical Society of America*, v. 88, n. 2, p. 622–641, 1990. ISSN 00014966. Cited 2 times in pages 19 and 30.

REPP, B. H. Diversity and commonality in music performance: an analysis of timing microstructure in Schumann's "Träumerei". *The Journal of the Acoustical Society of America*, v. 92, p. 227–260, 1992. ISSN 00014966. Cited 4 times in pages 4, 19, 30, and 73.

REPP, B. H. A microcosm of musical expression. I. Quantitative analysis of pianists' timing in the initial measures of Chopin's Etude in E major. *The Journal of the Acoustical Society of America*, v. 104, n. 2, p. 1085–1100, 1998. Cited in page 19.

REPP, B. H. Individual differences in the expressive shaping of a musical phrase: the opening of Chopin's Etude in E major. In: *Music, Mind and Science*. Seoul, Korea: Seoul National University Press, 1999. p. 239–270. Cited 3 times in pages 1, 12, and 19.

SAMPSON, A. *Audioread*. 2020. Available at: <<https://github.com/beetbox/audioread>>. Cited in page 49.

SCHLÜTER, J.; BÖCK, S. Improved musical onset detection with Convolutional Neural Networks. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Prague, Czech Republic: IEEE, 2014. p. 6979–6983. ISBN 978-1-4799-2893-4. ISSN 15206149. Cited 7 times in pages ix, 38, 39, 40, 43, 45, and 46.

SEARS, C. H. . A Contribution to the Psychology of Rhythm. *The American Journal of Psychology*, v. 13, n. 1, p. 28–61, 1902. Cited in page 15.

SEASHORE, C. E. The Tonoscope. *American Annals of the Deaf*, v. 61, n. 5, p. 405–408, 1916. Cited in page 17.

SEASHORE, C. E. *Psychology of Music*. [S.l.: s.n.], 1938. Cited 3 times in pages viii, 17, and 18.

SHACKLETON, N. The Development of The Clarinet. In: LAWSON, C. (Ed.). *The Cambridge Companion to the Clarinet*. Cambridge: The Press Syndicate of The University of Cambridge, 1995. chap. 02. Cited in page 8.

SLOBODA, J. A. Individual differences in music performance. *Trends in Cognitive Sciences*, v. 4, n. 10, p. 397–403, 2000. ISSN 13646613. Cited 3 times in pages 11, 12, and 73.

SMITH, J. *The Digital Audio Resampling Home Page*. Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, 2020. Available at: <<https://ccrma.stanford.edu/~jos/resample/>>. Cited in page 49.

TEIXEIRA, E. C. F.; LOUREIRO, M. A.; YEHIA, H. C. Expressiveness in Music From a Multimodal Perspective. *Music Perception: An Interdisciplinary Journal*, v. 36, n. 2, p. 201–216, dec 2018. ISSN 0730-7829. Cited in page 20.

THORNBURG, H.; LEISTIKOW, R. J.; BERGER, J. Melody extraction and musical onset detection via probabilistic models of framewise STFT peak data. *IEEE Transactions on Audio, Speech and Language Processing*, v. 15, n. 4, p. 1257–1272, 2007. ISSN 15587916. Cited in page 38.

VOS, J.; RASCH, R. The perceptual onset of musical tones. *Perception and Psychophysics*, v. 29, n. 4, p. 323–335, 1981. ISSN 15325962. Cited in page 13.

WALT, S. van der; COLBERT, S. C.; VAROQUAUX, G. The NumPy array: a structure for efficient numerical computation. *Computing in Science and Engineering, IEEE*, v. 13, n. 2, p. 22–30, 2011. Cited in page 49.

WOLFE, J. *Clarinet acoustics: an introduction*. 2002. 1–15 p. Available at: <http://www.phys.unsw.edu.au/jw/clarinetacoustics.html>. Cited 3 times in pages viii, 9, and 10.

WOLFE, J. The Acoustics of Woodwind Musical Instruments. *Acoustics Today*, v. 14, n. 1, p. 50–56, 2018. Cited 2 times in pages 7 and 10.

ZHOU, R.; REISS, J. Music onset detection combining energy-based and pitch-based approaches. *Proc. MIREX Audio Onset Detection Contest*, 2007. Cited in page 38.

A

Audio Segment Annotator

Figure 22 shows a screenshot of the page where the administrator creates or updates experiments. An experiment consists in a set of uploaded audio files, and a set of annotation categories. A category represents a distinct kind of structure that the annotator user will be expected to identify during the experiment, and will be specified as *points* or *segments* within the waveform. In the annotation interface, a point will render a vertical line over the waveform, and the segment will render a rectangle spanning an adjustable region. The example experiment shown in figure 22 consists in annotating note onset points and vibrato segments, and will result in the interactive interface shown in figure 23. The administrator may choose to let the experiment available publicly by checking the box labeled '*Allow anonymous annotations*' (figure 22). Otherwise, only authenticated users added by him will be able to undertake the experiment. It is also possible to specify how many audio files are supposed to be annotated per annotation session, and whether the files should be presented in random order (to avoid order effects).

Home · Experiment Manager · Experiments · Example experiment

Change experiment HISTORY

Basic information

Name:

Description:

This is an example experiment for the annotation of vibrato segments and note onsets.

Creation date: Oct. 27, 2019, 2:58 p.m.

Access and visibility

Allow anonymous annotation

Status: Online ▾

Annotation session options

File annotations per session: 2

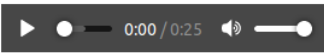
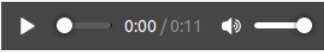
The number of files to be used in each annotation session. When empty, all the files in the experiment will be used.

Randomize file order

ANNOTATION CATEGORIES

CATEGORY NAME	ANNOTATION TYPE	COLOR	DELETE?
AnnotationCategory object (6)			
<input type="text" value="Note onsets"/>	Points ▾		<input type="checkbox"/>
AnnotationCategory object (7)			
<input type="text" value="Vibrato segment"/>	Segments ▾		<input type="checkbox"/>
+ Add another Annotation category			

AUDIO FILES

AUDIO	AUDIO FILE	UPLOAD DATE	DELETE?
/audio/uploads/experiment_5/01_-_Guitar_-_Stil_Got_The_Blues.wav			
	Currently: uploads/experiment_5/01_-_Guitar_-_Stil_Got_The_Blues.wav Change: Browse... No file selected.	Oct. 27, 2019, 2:58 p.m.	<input type="checkbox"/>
/audio/uploads/experiment_5/00_-_Flute_-_Iracema.wav			
	Currently: uploads/experiment_5/00_-_Flute_-_Iracema.wav Change: Browse... No file selected.	Oct. 27, 2019, 2:58 p.m.	<input type="checkbox"/>
+ Add another Audio file			

Delete
Save and add another
Save and continue editing
SAVE

Figure 22 – Editing an example experiment in the administration interface of Audio Segment Annotator.

Audio Segment Annotator

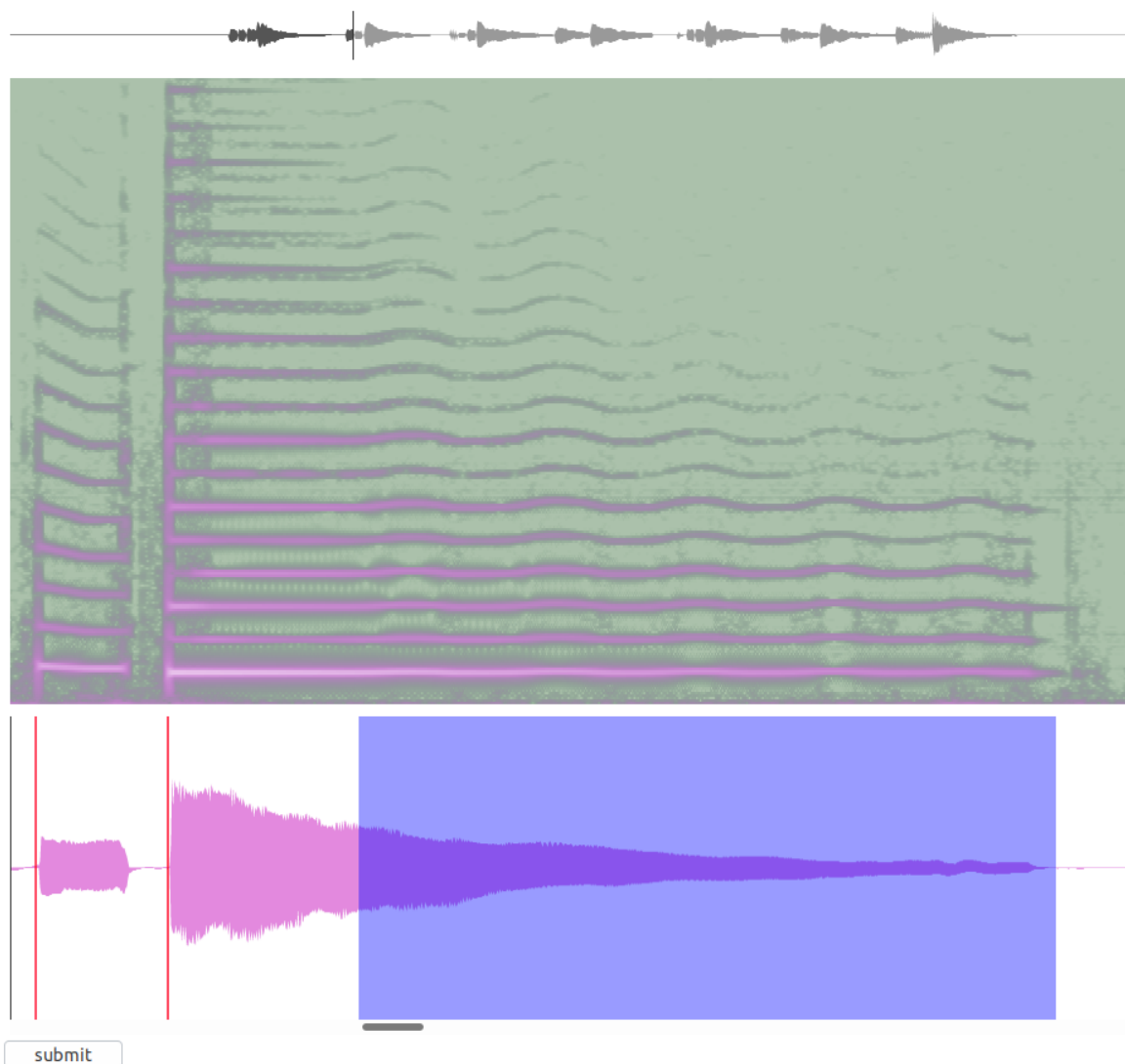


Figure 23 – The resulting interactive annotation interface for one audio file in an example experiment. It shows the overall waveform on top, followed by the zoomed-in spectrogram and audio waveform. In the zoomed-in waveform, there are two annotated note onset points (red vertical lines), and one vibrato segment (blue box). The black vertical line is the playhead.

B

Code Examples

The code examples shown in this section are compatible with the latest version of Iracema (0.2.1) available at the time of writing. This version requires Python 3.8 installed on the computer. We strongly recommend that you install iracema into a separate virtual environment when setting up your development environment, since it is a good practice to keep project-specific dependencies isolated from your base (system) Python installation. After activating your virtual environment, you can install iracema by running the following command on your command line:

```
$ pip3 install iracema
```

Listing B.1 – Installing Iracema.

If you are a Linux user, you might need to manually install an audio I/O library called PortAudio. If you are a MacOS X user, this library is probably already installed. In Debian / Ubuntu you can install it using the command *apt*:

```
$ sudo apt install libportaudio2
```

Listing B.2 – Installing PortAudio.

All the code examples in this section assume that Iracema has been imported using the convention shown in listing B.3.

```
1 import iracema as ir
```

Listing B.3 – Importing the library Iracema.

B.1 Loading and Processing Audio

To load an audio file, the user must provide a string containing the path to the location where it is stored. Iracema accepts file system paths to load files stored locally or HTTP URLs to download them from remote locations.

```
1 # loading audio file
2 audio = ir.Audio.load('mozart_q_m1.wav')
3
4 # playing audio
5 audio.play()
```

Listing B.4 – Loading and playing an audio object.

The class method `ir.Audio.load()` is used to load the content of an audio file into a newly instantiated audio object. The loaded `audio` object contains the attribute `audio.data`, which stores the loaded audio samples internally using a NumPy array. Other useful attributes are `audio.fs` (sampling frequency), `audio.start_time`, and `audio.duration`. This object also contains methods that enables the user to interact with this time series. For instance, the method `play()` plays the corresponding audio using the computer's sound device.

```
6 # resampling audio to 22050Hz
7 audio_resampled = audio.resample(22050)
8
9 # chaining methods to create a pipeline
10 audio_processed = (
11     audio.resample(16000).normalize().filter(1000, filter_type='highpass')
12 )
```

Listing B.5 – Methods of the audio object and method chaining.

Audio objects are equipped with several methods to process their data, like the method `resample()`, which resamples the audio to the specified sampling rate. It is possible to chain multiple processing methods to create a data processing pipeline. These methods do not modify the original audio object. Instead, they instantiate new objects. This chaining concept also applies to other time series objects in the library (although the available methods may differ from one class to another).

B.2 Extracting Features from Audio

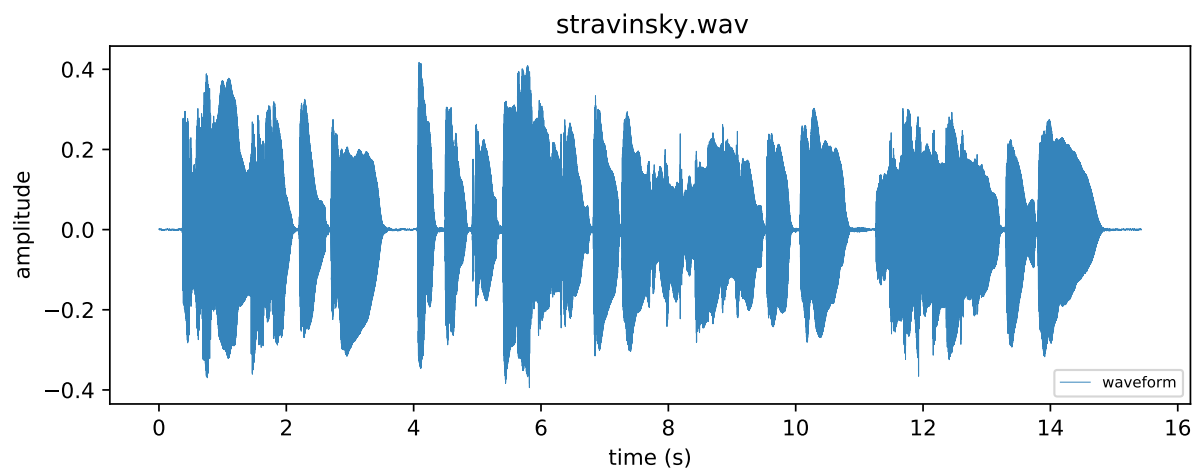
```
1 # loading audio file
2 clarinet = ir.Audio.load('stravinsky.wav')
3
4 # setting window and hop lengths
5 window, hop = 2048, 411
6
7 # calculating the RMS and STFT
8 clarinet_rms = ir.features.rms(clarinet, window, hop)
9 clarinet_stft = ir.spectral.STFT(clarinet, window, hop, fft_len=4096)
10
11 # plotting data
12 clarinet.plot();
13 clarinet_rms.plot();
14 ir.plot.spectrogram(clarinet_stft, fmax=10000);
```

Listing B.6 – Extracting the RMS and applying the STFT to the audio data.

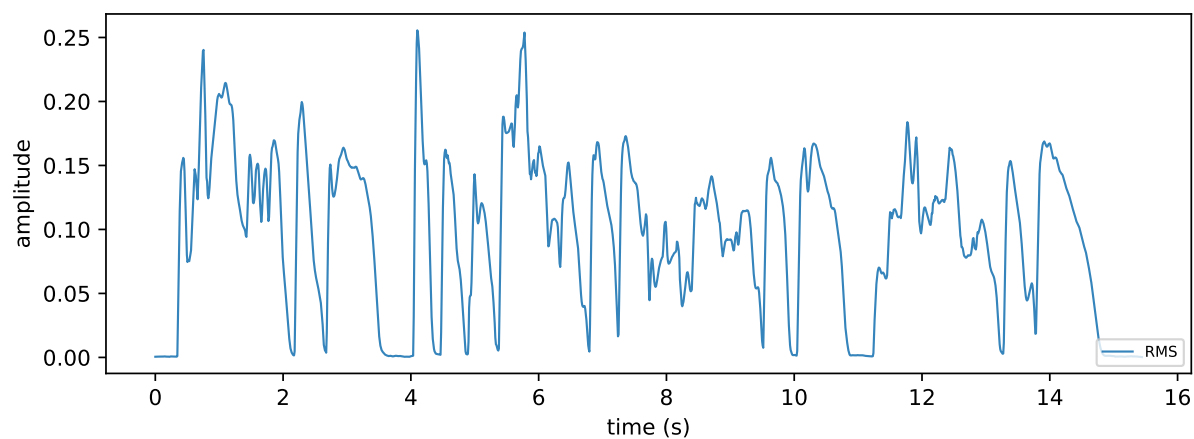
Many feature extraction methods receive as input the arguments `window` and `hop`, which define the length of the analysis window and the length of the hop between two successive windows. Two variables with these values are set in line 5 (listing B.6). The method `ir.features.rms()` in line 8 calculates the RMS of the loaded audio. In line 9, an object of the class `ir.spectral.STFT` is instantiated, generating an object that contains the complex form of the spectrum of the signal. In lines 12 and 13, we use the instance method `plot()` of the objects `clarinet` and `clarinet_rms` to display the waveform of the audio signal and its RMS. In line 14, we use another plot method, this time from the module `plot`, to display the signal's spectrogram (`ir.plot.spectrogram()`). The resulting plots are shown in figure 24.

```
15 # estimate the pitch and the frequencies of the harmonics
16 clarinet_pitch = ir.pitch.hps(clarinet_stft, 120, 2000)
17 clarinet_harmonics = ir.harmonics.extract(clarinet_stft, clarinet_pitch)
18
19 # plot the estimated pitch and harmonics
20 ir.plot.waveform_spectrogram_pitch(
21     clarinet, clarinet_stft, clarinet_pitch, fmax=5000
22 );
23 ir.plot.waveform_spectrogram_harmonics(
24     clarinet, clarinet_stft, clarinet_harmonics['frequency']
25 );
```

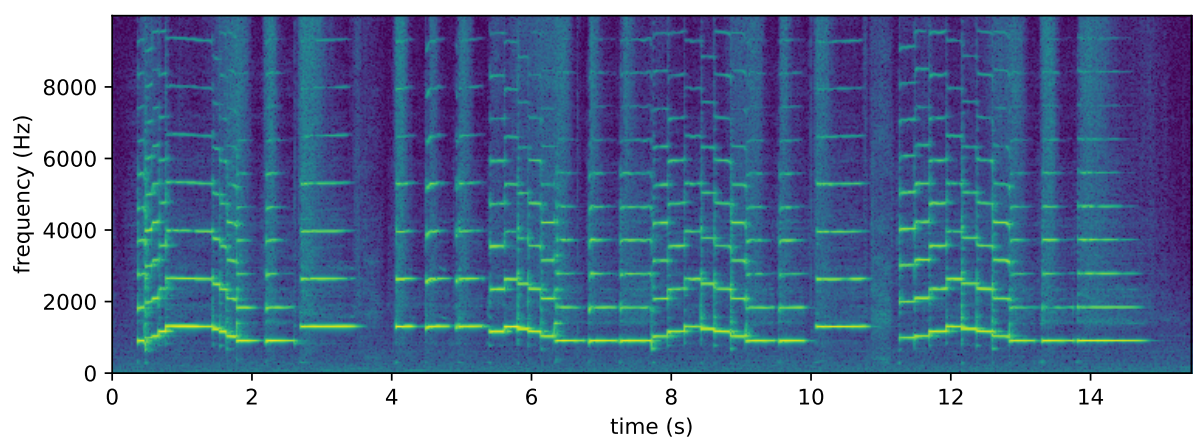
Listing B.7 – Estimating the pitch and the frequencies of the harmonics.



(a) Audio waveform.

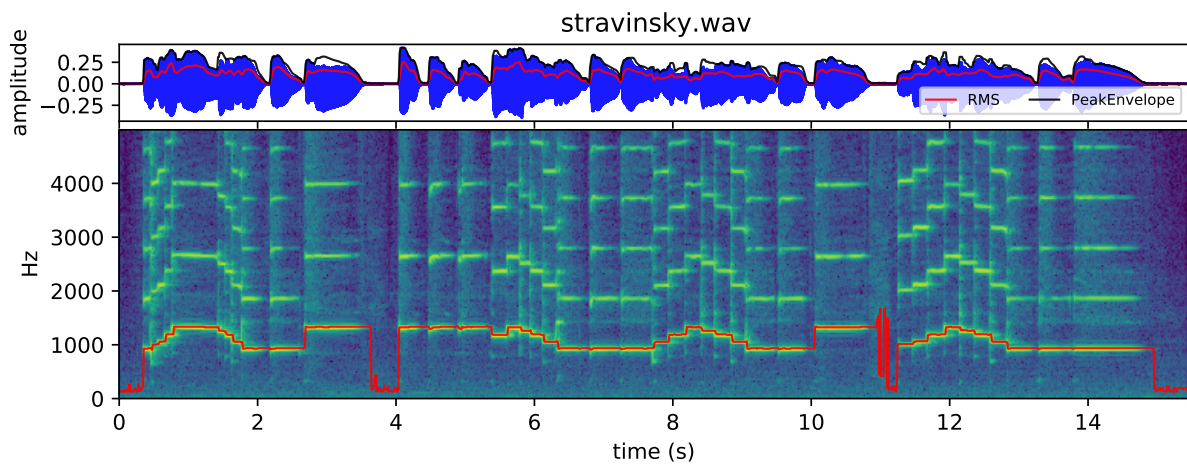


(b) RMS curve.

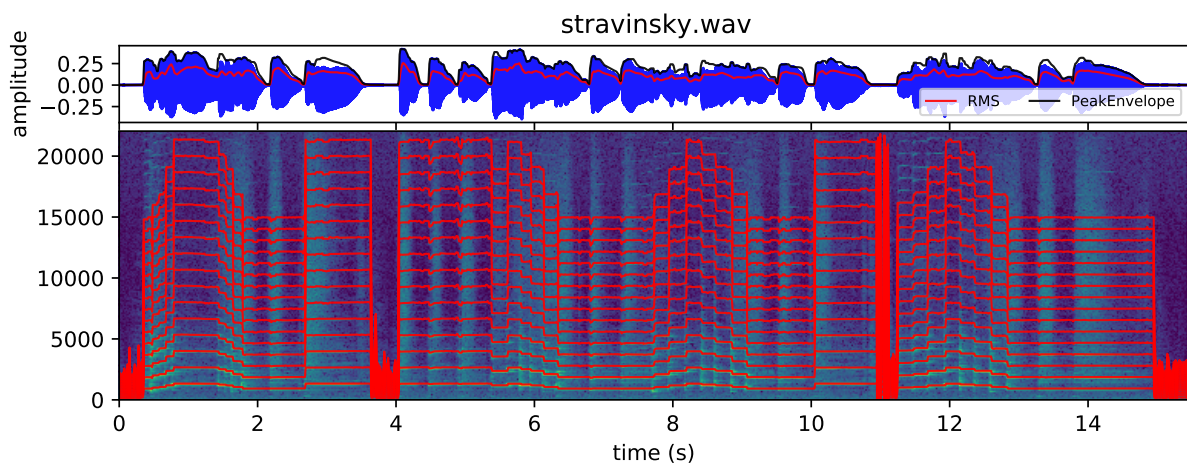


(c) Spectrogram visualization.

Figure 24 – Resulting plots showing the waveform, RMS and spectrogram, obtained running the code in listing B.6 for the excerpt `stravinsky.wav`.



(a) Pitch extracted using the harmonic product spectrum method.



(b) Estimated frequencies of the harmonics.

Figure 25 – Resulting plots showing the estimated pitch and harmonics, obtained running the code shown in listing B.7 for the excerpt `stravinsky.wav`.

In lines 16 and 17 (listing B.7) the pitch and frequencies of the harmonics are estimated, and their values plotted in lines 20-25. figure 25 shows the resulting plots.

```

26 # extracting features
27 sf = ir.features.spectral_flux(clarinet_stft)
28 sc = ir.features.spectral_centroid(clarinet_stft)
29 hfc = ir.features.hfc(clarinet_stft)
30 no = ir.features.noisiness(clarinet_stft, clarinet_harmonics['magnitude'])
31
32 # plotting features
33 ir.plot.waveform_trio_and_features(clarinet, features=(sf, sc, no, hfc));

```

Listing B.8 – Extracting and plotting four different features.

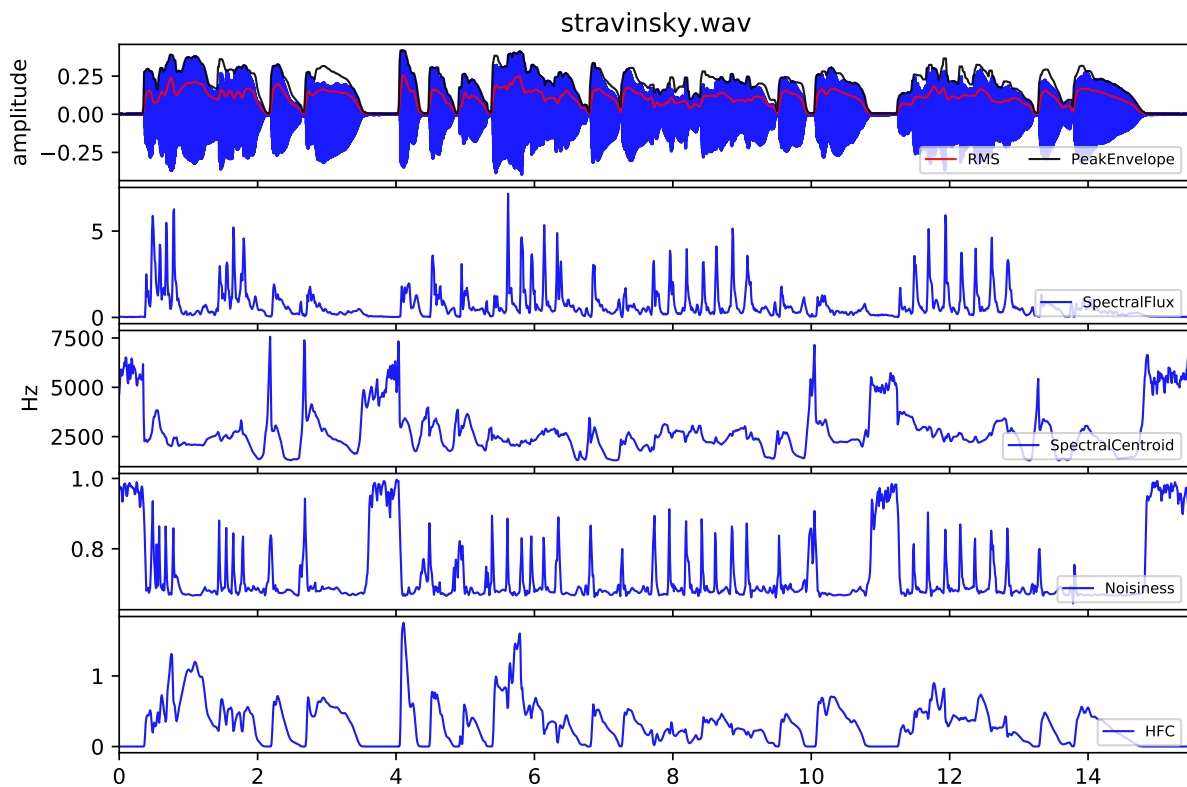


Figure 26 – Resulting plot for the features calculated in listing B.8.

In lines 27-30 (listing B.8), four different features are extracted for the audio excerpt: spectral flux, spectral centroid, HFC and noisiness. In line 33, these features are all plotted using charts with synchronized axes. The results are shown in figure 26.

B.3 Note Segmentation

In the example shown in listing B.9, the audio file is loaded, and its onsets are detected using the CNN onset detection model, with the method `ir.segmentation.onsets.cnn_model()` (line 5). This method returns a `PointList` object, which is attributed to the variable `onsets`. Then, in line 8, the instance method `audio.play_with_clicks()` is called, passing the list of onset points as an argument. As a result, this method will use the computer's sound device to play the loaded audio with short clicks synchronized to the note onsets detected by the CNN model. This method is useful to validate the results of the model.

```

1 # load audio file
2 audio = ir.Audio.load('mozart_q_m4.wav')
3
4 # detect note onsets
5 onsets = ir.segmentation.onsets.cnn_model(audio)

```

```
6  
7 # play audio with clicks synchronized to the onsets  
8 audio.play_with_clicks(onsets)
```

Listing B.9 – Detecting note onsets.